

1 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Уколико бит 2 има вредност 0, ради се о регистарском директном адресирању (regdir) код кога се користи један од регистара опште намене $R[0]$ и $R[1]$, или базни регистар $BR_{15..0}$ или индексни регистар $XR_{15..0}$ што је специфициран вредностима 00, 01, 10 и 11 битова 1 и 0. Дужина инструкције је један бајт. Уколико бит 2 има вредност 1, ради се о непосредном (immed), базном адресирању са померајем (base), индексном адресирању са померајем (index) или о базно индексном адресирању са померајем (baseindex) специфицираном вредностима битова 1 и 0. Код непосредног адресирања други и трећи бајт инструкције садрже 16 битни податак, при чему је млађи бајт податка дат другим а старији бајт податка трећим бајтом инструкције. Дужина инструкције је три бајта. Код базног адресирања са померајем имплицитно се користи базни регистар $BR_{15..0}$, док други и трећи бајт инструкције садржи 16 битни померај који је дат као целобројна величина са знаком, при чему је млађи бајт помераја дат другим а старији бајт адресе трећим бајтом инструкције. Дужина инструкције је три бајта. Код индексног адресирања са померајем имплицитно се користи индексни регистар $XR_{15..0}$, док други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је два бајта. Код базно индексног адресирања са померајем имплицитно се користе регистри $BR_{15..0}$ и $XR_{15..0}$, а други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је 2 бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора А у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора А у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI registra програмске статусне речи $PSW_{15...0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15...0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15...0}$ и програмска статусна речи $PSW_{15...0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на

коју указује садржај регистра $IVTP_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$ највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15..0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15..0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

2 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Уколико бит 2 има вредност 0, ради се о регистарском директном адресирању (regdir) код кога се користи један од регистара опште намене $R[0]$ и $R[1]$, или базни регистар $BR_{15..0}$ или индексни регистар $XR_{15..0}$ што је специфициран вредностима 00, 01, 10 и 11 битова 1 и 0. Дужина инструкције је један бајт. Уколико бит 2 има вредност 1, ради се о непосредном (immed), базном адресирању са померајем (base), индексном адресирању са померајем (index) или о базно индексном адресирању са померајем (baseindex) специфицираном вредностима битова 1 и 0. Код непосредног адресирања други и трећи бајт инструкције садрже 16 битни податак, при чему је млађи бајт податка дат другим а старији бајт податка трећим бајтом инструкције. Дужина инструкције је три бајта. Код базног адресирања са померајем имплицитно се користи базни регистар $BR_{15..0}$, док други и трећи бајт инструкције садржи 16 битни померај који је дат као целобројна величина са знаком, при чему је млађи бајт помераја дат другим а старији бајт адресе трећим бајтом инструкције. Дужина инструкције је три бајта. Код индексног адресирања са померајем имплицитно се користи индексни регистар $XR_{15..0}$, док други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је два бајта. Код базно индексног адресирања са померајем имплицитно се користе регистри $BR_{15..0}$ и $XR_{15..0}$, а други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је 2 бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора А у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора А у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI registra програмске статусне речи $PSW_{15...0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15...0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15...0}$ и програмска статусна речи $PSW_{15...0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на

коју указује садржај регистра $IVTP_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$ највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15..0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15..0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

3 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине један бајт. Адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Бит 2 има вредност 0 за регистарско директно и регистарско индиректно адресирање, а вредност 1 за непосредно, РС релативно са померајем, меморијско директно и меморијско индиректно адресирање. Уколико бит 2 има вредност 0, тада се вредностима 0 и 1 бита 1 одређује регистарско директно (regdir) и регистарско индиректно адресирање (regind), респективно, а битом 0 специфицира регистар опште намене R[0] или R[1] који се користе за ова два адресирања. Дужина инструкције је 1 бајт. Уколико бит 2 има вредност 1, тада битови 1 и 0 одређује непосредно (immed), РС релативно са померајем (pcrel), меморијско директно (memdir) и меморијско индиректно адресирање (memind). Код непосредног адресирања други бајт инструкције садрже 8 битни податак. Дужина инструкције је 2 бајта. Код РС релативног са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је 2 бајта. Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе меморијске локације дат другим а млађи бајт трећим бајтом. Дужина инструкције је 3 бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на последњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за

једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса 8 битне величине у акумулатор AB (LD), инструкција преноса 16 битне величине у акумулатор AW (LDW), инструкција преноса 8 битне величине из акумулатора AB (ST), инструкција преноса 16 битне величине из акумулатора AW (STW), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $\text{IVTP}_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

4 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: меморијско директно адресирање (*memdir*), меморијско индиректно адресирање (*memind*), РС релативно са померајем (*pcrel*) и непосредно адресирање (*immed*). Код меморијског директног и меморијског индиректног адресирања трећи и четврти бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе дат трећим бајтом а млађи бајт адресе четвртим бајтом. Код меморијског индиректног адресирања адреса дужине 16 бита заузима две суседне меморијске локације, при чему је старији бајт адресе налази на нижој а млађи бајт адресе на вишој локацији. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта. Код РС релативно са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је три бајта. Код непосредног адресирања трећи и четврти бајт инструкције садрже 16 битни податак, при чему је старији бајт податка дат трећим бајтом а млађи бајт податка четвртим бајтом. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (*RTS*), инструкција повратка из прекидне рутине (*RTI*), инструкција аритметичког померања садржаја акумулатора удесно за једно место (*ASR*), инструкција логичког померања садржаја акумулатора удесно за једно место (*LSR*), инструкција ротирања садржаја акумулатора удесно за једно место (*ROR*), инструкција ротирања садржаја акумулатора и индикатора С

удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограма (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15..0}$ и програмска статусна речи $PSW_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $IVTP_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

5 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Уколико бит 2 има вредност 0, ради се о регистарском директном адресирању (regdir) код кога се користи један од регистара опште намене $R[0]$ и $R[1]$, или базни регистар $BR_{15...0}$ или индексни регистар $XR_{15...0}$ што је специфициран вредностима 00, 01, 10 и 11 битова 1 и 0. Дужина инструкције је један бајт. Уколико бит 2 има вредност 1, ради се о непосредном (immed), базном адресирању са померајем (base), индексном адресирању са померајем (index) или о базно индексном адресирању са померајем (baseindex) специфицираном вредностима битова 1 и 0. Код непосредног адресирања други и трећи бајт инструкције садрже 16 битни податак, при чему је старији бајт податка дат другим а млађи бајт податка трећим бајтом инструкције. Дужина инструкције је три бајта. Код базног адресирања са померајем имплицитно се користи базни регистар $BR_{15...0}$, док други и трећи бајт инструкције садржи 16 битни померај који је дат као целобројна величина са знаком, при чему је старији бајт помераја дат другим а млађи бајт адресе трећим бајтом инструкције. Дужина инструкције је три бајта. Код индексног адресирања са померајем имплицитно се користи индексни регистар $XR_{15...0}$, док други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је два бајта. Код базно индексног адресирања са померајем имплицитно се користе регистри $BR_{15...0}$ и $XR_{15...0}$, а други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је 2 бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја

аккумулятора удесно за једно место (ASR), инструкција логичког померања садржаја аккумулятора удесно за једно место (LSR), инструкција ротирања садржаја аккумулятора удесно за једно место (ROR), инструкција ротирања садржаја аккумулятора и индикатора C удесно за једно место (RORC), инструкција аритметичког померања садржаја аккумулятора улево за једно место (ASL), инструкција логичког померања садржаја аккумулятора улево за једно место (LSR), инструкција ротирања садржаја аккумулятора улево за једно место (ROR), инструкција ротирања садржаја аккумулятора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај аккумулятора A у регистар IVTP (STIVTP) и инструкција преноса садржај аккумулятора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у аккумулятор (LD), инструкција преноса из аккумулятора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15..0}$ и програмска статусна речи $PSW_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $IVTP_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који

долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$ највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

6 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 16 битна целобројна величина са знаком дата другим и трећим бајтом инструкције, при чему је млађи бајт дат другим а старији трећим бајтом инструкције. Дужина инструкције је три бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: регистарско директно адресирање (regdir), регистарско индиректно адресирање (regind), меморијско директно адресирање (memdir), регистарско индиректно са померајем адресирање (regindpom). Код регистарског директног и регистарског индиректног адресирања се користе неки од регистара опште намене R[0] до R[63] специфициран битовима 5 до 0 другог бајта инструкције. Дужина инструкција је два бајта. Код меморијског директног адресирања трећи и четврти бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе дат трећим бајтом а старији бајт адресе четвртим бајтом. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта. Код регистарског индиректног са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Један од регистара опште намене R[0] до R[63] који се користи специфициран је битовима 5 до 0 другог бајта инструкције. Дужина инструкција је три бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на последњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора

улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограма (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15...0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15...0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15...0}$ и програмска статусна речи $PSW_{15...0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15...0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$ највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна

величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

7 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 16 битна целобројна величина са знаком дата другим и трећим бајтом инструкције, при чему је млађи бајт дат другим а старији трећим бајтом инструкције. Дужина инструкције је три бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: регистарско директно адресирање (regdir), регистарско индиректно адресирање (regind), меморијско директно адресирање (memdir), регистарско индиректно са померајем адресирање (regindpom). Код регистарског директног и регистарског индиректног адресирања се користе неки од регистара опште намене R[0] до R[63] специфициран битовима 5 до 0 другог бајта инструкције. Дужина инструкција је два бајта. Код меморијског директног адресирања трећи и четврти бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе дат трећим бајтом а старији бајт адресе четвртим бајтом. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта. Код регистарског индиректног са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Један од регистара опште намене R[0] до R[63] који се користи специфициран је битовима 5 до 0 другог бајта инструкције. Дужина инструкција је три бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на последњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора

улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограма (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15..0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна

величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

8 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине 2 бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој а млађи бајт на вишој адреси. У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције условног скока Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0001 за све инструкције безусловног скока Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 1111 за безадресне инструкције. Битовима 3 до 0 првог бајта инструкције специфицира код операције за безадресне инструкције. Дужина инструкција је један бајт.

Битови 7, 6, 5 и 4 првог бајта инструкције у опсегу вредности 0010 до 1110 специфицирају код операције за адресне инструкције. Дужина инструкција је 1, 2, или 3 бајта и зависи од специфицираног начина адресирања.

Начини адресирања су специфицирани битовима 3 и 2 првог бајта инструкције. Процесор поседује следеће начине адресирања: регистарско индиректно са померајем адресирање (regindrom), регистарско директно адресирање (regdir), РС релативно са померајем (pcrel) и непосредно адресирање (immed). Код регистарског индиректног адресирања са померајем 16 битни померај је целобројна величина са знаком у другом комплементу дата другим и трећим бајтом инструкције при чему је старији бајт дат другим бајтом а млађи бајт помераја трећим бајтом. Битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је три бајта. Код регистарског директног адресирања битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 1 бајт. Код РС релативно са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је два бајта. Код непосредног адресирања 16 битни операнд је дат другим и трећим бајтом инструкције, при чему је старији бајт операнда дат другим а млађи бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 3 бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за

једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $\text{IVTP}_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по

две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

9 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Бит 2 има вредност 0 за регистарско директно и регистарско индиректно адресирање, а вредност 1 за непосредно, РС релативно са померајем, меморијско директно и меморијско индиректно адресирање. Уколико бит 2 има вредност 0, тада се вредностима 0 и 1 бита 1 одређује регистарско директно (regdir) и регистарско индиректно адресирање (regind), респективно, а битом 0 специфицира регистар опште намене R[0] или R[1] који се користе за ова два адресирања. Дужина инструкције је 1 бајт. Уколико бит 2 има вредност 1, тада битови 1 и 0 одређује непосредно (immed), РС релативно са померајем (pcrel), меморијско директно (memdir) и меморијско индиректно адресирање (memind). Код непосредног адресирања други и трећи бајт инструкције садрже 16 битни податак, при чему је старији бајт податка дат другим бајтом а млађи бајт податка трећим бајтом. Дужина инструкције је 3 бајта. Код РС релативног са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је 2 бајта. Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе меморијске локације дат другим а млађи бајт трећим бајтом. Дужина инструкције је 3 бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С

удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограма (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15..0}$ и програмска статусна речи $PSW_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $IVTP_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

10 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине 2 бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој а млађи бајт на вишој адреси.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције условног скока. Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0001 за све инструкције безусловног скока. Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 1111 за безадресне инструкције. Битовима 3 до 0 првог бајта инструкције специфицира код операције за безадресне инструкције. Дужина инструкција је један бајт.

Битови 7, 6, 5 и 4 првог бајта инструкције у опсегу вредности 0010 до 1110 специфицирају код операције за адресне инструкције. Дужина инструкција је 1, 2, или 3 бајта и зависи од специфицираног начина адресирања.

Начини адресирања су специфицирани битовима 3 и 2 првог бајта инструкције. Процесор поседује следеће начине адресирања: регистарско индиректно са померајем адресирање (regindrom), регистарско директно адресирање (regdir), РС релативно са померајем (pcrel) и непосредно адресирање (immed). Код регистарског индиректног адресирања са померајем 16 битни померај је целобројна величина са знаком у другом комплементу дата другим и трећим бајтом инструкције при чему је старији бајт дат другим бајтом а млађи бајт помераја трећим бајтом. Битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је три бајта. Код регистарског директног адресирања битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 1 бајт. Код РС релативно са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је два бајта. Код непосредног адресирања 16 битни операнд је дат другим и трећим бајтом инструкције, при чему је старији бајт операнда дат другим а млађи бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 3 бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за

једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $\text{IVTP}_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по

две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

11 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине 1 бајт. Адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој а млађи бајт на вишој адреси.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције условног скока. Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0001 за све инструкције безусловног скока. Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 1111 за безадресне инструкције. Битовима 3 до 0 првог бајта инструкције специфицира код операције за безадресне инструкције. Дужина инструкција је један бајт.

Битови 7, 6, 5 и 4 првог бајта инструкције у опсегу вредности 0010 до 1110 специфицирају код операције за адресне инструкције. Дужина инструкција је 1, 2, или 3 бајта и зависи од специфицираног начина адресирања.

Начини адресирања су специфицирани битовима 3 и 2 првог бајта инструкције. Процесор поседује следеће начине адресирања: непосредно адресирање (immed), меморијско директно адресирање (memdir), регистарско индиректно са померајем адресирање (regindrom) и регистарско директно адресирање (regdir). Код непосредног адресирања 8 битни операнд је дат другим бајтом инструкције. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 2 бајта. Код меморијског директног адресирања 16 битна адреса меморијске локације је дата другим и трећим бајтом инструкције, при чему је старији бајт адресе дат другим а млађи бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 3 бајта. Код регистарског индиректног адресирања са померајем 8 битни померај је целобројна величина са знаком у другом комплементу дата другим бајтом инструкције. Битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 2 бајта. Код регистарског директног адресирања битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 1 бајт.

Стек расте према вишим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С улево за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR),

инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса 8 битне величине у акумулатор AB (LD), инструкција преноса 16 битне величине у акумулатор AW (LDW), инструкција преноса 8 битне величине из акумулатора AB (ST), инструкција преноса 16 битне величине из акумулатора AW (STW), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST и STW није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15...0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15...0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15...0}$ и програмска статусна речи $\text{PSW}_{15...0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15...0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна

величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

12 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине један бајт. Адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: меморијско директно адресирање (*memdir*), меморијско индиректно адресирање (*memind*), РС релативно са померајем (*pcrel*) и непосредно адресирање (*immed*). Код меморијског директног и меморијског индиректног адресирања трећи и четврти бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе дат трећим бајтом а млађи бајт адресе четвртим бајтом. Код меморијског индиректног адресирања адреса дужине 16 бита заузима две суседне меморијске локације, при чему је старији бајт адресе налази на нижој а млађи бајт адресе на вишој локацији. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта. Код РС релативно са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је три бајта. Код непосредног адресирања трећи бајт инструкције садржи 8 битни податак. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је три бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (*RTS*), инструкција повратка из прекидне рутине (*RTI*), инструкција аритметичког померања садржаја акумулатора удесно за једно место (*ASR*), инструкција логичког померања садржаја акумулатора удесно за једно место (*LSR*), инструкција ротирања садржаја акумулатора удесно за једно место (*ROR*), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (*RORC*), инструкција аритметичког померања садржаја акумулатора улево за једно место (*ASL*), инструкција логичког померања садржаја акумулатора улево за

једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса 8 битне величине у акумулатор AB (LD), инструкција преноса 16 битне величине у акумулатор AW (LDW), инструкција преноса 8 битне величине из акумулатора AB (ST), инструкција преноса 16 битне величине из акумулатора AW (STW), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST и STW није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $\text{IVTP}_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

13 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине 1 бајта. Адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој а старији бајт на вишој адреси.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције условног скока Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0001 за све инструкције безусловног скока Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 1111 за безадресне инструкције. Битовима 3 до 0 првог бајта инструкције специфицира код операције за безадресне инструкције. Дужина инструкција је један бајт.

Битови 7, 6, 5 и 4 првог бајта инструкције у опсегу вредности 0010 до 1110 специфицирају код операције за адресне инструкције. Дужина инструкција је 1, 2, или 3 бајта и зависи од специфицираног начина адресирања.

Начини адресирања су специфицирани битовима 3 и 2 првог бајта инструкције. Процесор поседује следеће начине адресирања: регистарско индиректно са померајем адресирање (regindrom), регистарско директно адресирање (regdir), РС релативно са померајем (pcrel) и непосредно адресирање (immed). Код регистарског индиректног адресирања са померајем 16 битни померај је целобројна величина са знаком у другом комплементу дата другим и трећим бајтом инструкције при чему је старији бајт дат другим бајтом а млађи бајт помераја трећим бајтом. Битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је три бајта. Код регистарског директног адресирања битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 1 бајт. Код РС релативно са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је два бајта. Код непосредног адресирања 8 битни операнд је дат другим бајтом инструкције. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 2 бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на задњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR),

инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса 8 битне величине у акумулатор AB (LD), инструкција преноса 16 битне величине у акумулатор AW (LDW), инструкција преноса 8 битне величине из акумулатора AB (ST), инструкција преноса 16 битне величине из акумулатора AW (STW), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST и STW није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15...0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15...0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15...0}$ и програмска статусна речи $\text{PSW}_{15...0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15...0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна

величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

14 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине 1 бајт. Адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој а старији бајт на вишој адреси.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције условног скока Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0001 за све инструкције безусловног скока Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 1111 за безадресне инструкције. Битовима 3 до 0 првог бајта инструкције специфицира код операције за безадресне инструкције. Дужина инструкција је један бајт.

Битови 7, 6, 5 и 4 првог бајта инструкције у опсегу вредности 0010 до 1110 специфицирају код операције за адресне инструкције. Дужина инструкција је 1, 2, или 3 бајта и зависи од специфицираног начина адресирања.

Начини адресирања су специфицирани битовима 3 и 2 првог бајта инструкције. Процесор поседује следеће начине адресирања: непосредно адресирање (immed), меморијско директно адресирање (memdir), регистарско индиректно са померајем адресирање (regindrom) и регистарско директно адресирање (regdir). Код непосредног адресирања 8 битни операнд је дат другим бајтом инструкције. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 2 бајта. Код меморијског директног адресирања 16 битна адреса меморијске локације је дата другим и трећим бајтом инструкције, при чему је млађи бајт адресе дат другим а старији бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 3 бајта. Код регистарског индиректног адресирања са померајем 8 битни померај је целобројна величина са знаком у другом комплементу дата другим бајтом инструкције. Битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 2 бајта. Код регистарског директног адресирања битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 1 бајт.

Стек расте према нижим меморијским локацијама, а регистар SP указује на задњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С улево за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR),

инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса 8 битне величине у акумулатор AB (LD), инструкција преноса 16 битне величине у акумулатор AW (LDW), инструкција преноса 8 битне величине из акумулатора AB (ST), инструкција преноса 16 битне величине из акумулатора AW (STW), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST и STW није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15...0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15...0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15...0}$ и програмска статусна речи $\text{PSW}_{15...0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15...0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна

величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

15 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Уколико бит 2 има вредност 0, ради се о регистарском директном адресирању (`regdir`) код кога се користи један од регистара опште намене $R[0]$, $R[1]$, $R[2]$ и $R[3]$ специфициран вредностима 00, 01, 10 и 11 битова 1 и 0. Дужина инструкције је један бајт. Уколико бит 2 има вредност 1, ради се о непосредном (`immed`), РС релативном са померајем (`pcrel`), меморијском директном (`memdir`) или меморијском индиректном (`memind`) адресирању специфицираном вредностима битова 1 и 0. Код непосредног адресирања други и трећи бајт инструкције садрже 16 битни податак, при чему је старији бајт податка дат другим а млађи бајт податка трећим бајтом инструкције. Дужина инструкције је три бајта. Код РС релативног са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је два бајта. Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе дат другим а млађи бајт адресе трећим бајтом инструкције. Дужина инструкције је три бајта.

Стек расте према нижим меморијским локацијама, а регистар `SP` указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (`RTS`), инструкција повратка из прекидне рутине (`RTI`), инструкција аритметичког померања садржаја акумулатора удесно за једно место (`ASR`), инструкција логичког померања садржаја акумулатора удесно за једно место (`LSR`), инструкција ротирања садржаја акумулатора удесно за једно место (`ROR`), инструкција ротирања садржаја акумулатора и индикатора `C` удесно за једно место (`RORC`), инструкција аритметичког померања садржаја акумулатора улево за једно место (`ASL`), инструкција логичког померања садржаја акумулатора улево за

једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $\text{IVTP}_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по

две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

16 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине 1 бајт. Адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој а млађи бајт на вишој адреси.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције условног скока Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0001 за све инструкције безусловног скока Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 1111 за безадресне инструкције. Битовима 3 до 0 првог бајта инструкције специфицира код операције за безадресне инструкције. Дужина инструкција је један бајт.

Битови 7, 6, 5 и 4 првог бајта инструкције у опсегу вредности 0010 до 1110 специфицирају код операције за адресне инструкције. Дужина инструкција је 1, 2, или 3 бајта и зависи од специфицираног начина адресирања.

Начини адресирања су специфицирани битовима 3 и 2 првог бајта инструкције. Процесор поседује следеће начине адресирања: меморијско директно адресирање (memdir), меморијско индиректно адресирање (memind), регистарско директно адресирање (regdir) и непосредно адресирање (immed). Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе дат другим бајтом а млађи бајт адресе трећим бајтом. Код меморијског индиректног адресирања адреса дужине 16 бита заузима две суседне меморијске локације, при чему је старији бајт адресе налази на нижој а млађи бајт адресе на вишој локацији. Битови 1 до 0 другог бајта инструкције се не користе. Дужина инструкција је три бајта. Код регистарског директног адресирања битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 1 бајт. Код непосредног адресирања 8 битни операнд је дат другим бајтом инструкције. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 2 бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С улево за једно место (RORC),

инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса 8 битне величине у акумулатор AB (LD), инструкција преноса 16 битне величине у акумулатор AW (LDW), инструкција преноса 8 битне величине из акумулатора AB (ST), инструкција преноса 16 битне величине из акумулатора AW (STW), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST и STW није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15..0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по

две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

17 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине један бајт. Адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Бит 2 има вредност 0 за регистарско директно и регистарско индиректно адресирање, а вредност 1 за непосредно, РС релативно са померајем, меморијско директно и меморијско индиректно адресирање. Уколико бит 2 има вредност 0, тада се вредностима 0 и 1 бита 1 одређује регистарско директно (regdir) и регистарско индиректно адресирање (regind), респективно, а битом 0 специфицира регистар опште намене R[0] или R[1] који се користе за ова два адресирања. Дужина инструкције је 1 бајт. Уколико бит 2 има вредност 1, тада битови 1 и 0 одређује непосредно (immed), РС релативно са померајем (pcrel), меморијско директно (memdir) и меморијско индиректно адресирање (memind), респективно Код непосредног адресирања други бајт инструкције садржи 8 битни податак. Дужина инструкције је 2 бајта. Код РС релативног са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је 2 бајта. Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе меморијске локације дат другим а млађи бајт трећим бајтом. Дужина инструкције је 3 бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на последњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за

једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса 8 битне величине у акумулатор AB (LD), инструкција преноса 16 битне величине у акумулатор AW (LDW), инструкција преноса 8 битне величине из акумулатора AB (ST), инструкција преноса 16 битне величине из акумулатора AW (STW), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST и STW није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15..0}$ и програмска статусна речи $PSW_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $IVTP_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

18 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 16 битна целобројна величина са знаком дата другим и трећим бајтом инструкције, при чему је млађи бајт дат другим а старији трећим бајтом инструкције. Дужина инструкције је три бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: регистарско директно адресирање (regdir), регистарско индиректно адресирање (regind), меморијско директно адресирање (memdir), регистарско индиректно са померајем адресирање (regindpom). Код регистарског директног и регистарског индиректног адресирања се користе неки од регистара опште намене R[0] до R[63] специфициран битовима 5 до 0 другог бајта инструкције. Дужина инструкција је два бајта. Код меморијског директног адресирања трећи и четврти бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе дат трећим бајтом а старији бајт адресе четвртим бајтом. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта. Код регистарског индиректног са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Један од регистара опште намене R[0] до R[63] који се користи специфициран је битовима 5 до 0 другог бајта инструкције. Дужина инструкција је три бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на последњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С

удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограма (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15..0}$ и програмска статусна речи $PSW_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $IVTP_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

19 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Бит 2 има вредност 0 за регистарско директно и регистарско индиректно адресирање, а вредност 1 за непосредно, РС релативно са померајем, меморијско директно и меморијско индиректно адресирање. Уколико бит 2 има вредност 0, тада се вредностима 0 и 1 бита 1 одређује регистарско директно (regdir) и регистарско индиректно адресирање (regind), респективно, а битом 0 специфицира регистар опште намене R[0] или R[1] који се користе за ова два адресирања. Дужина инструкције је 1 бајт. Уколико бит 2 има вредност 1, тада битови 1 и 0 одређује непосредно (immed), РС релативно са померајем (pcrel), меморијско директно (memdir) и меморијско индиректно адресирање (memind), респективно Код непосредног адресирања други и трећи бајт инструкције садрже 16 битни податак, при чему је млађи бајт податка дат другим бајтом а старији бајт податка трећим бајтом. Дужина инструкције је 3 бајта. Код РС релативног са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је 2 бајта. Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе меморијске локације дат другим а старији бајт трећим бајтом. Дужина инструкције је 3 бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С

удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограма (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15..0}$ и програмска статусна речи $PSW_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $IVTP_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

20 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Бит 2 има вредност 0 за регистарско директно и регистарско индиректно адресирање, а вредност 1 за непосредно, РС релативно са померајем, меморијско директно и меморијско индиректно адресирање. Уколико бит 2 има вредност 0, тада се вредностима 0 и 1 бита 1 одређује регистарско директно (regdir) и регистарско индиректно адресирање (regind), респективно, а битом 0 специфицира регистар опште намене R[0] или R[1] који се користе за ова два адресирања. Дужина инструкције је 1 бајт. Уколико бит 2 има вредност 1, тада битови 1 и 0 одређује непосредно (immed), РС релативно са померајем (pcrel), меморијско директно (memdir) и меморијско индиректно адресирање (memind). Код непосредног адресирања други и трећи бајт инструкције садрже 16 битни податак, при чему је млађи бајт податка дат другим бајтом а старији бајт податка трећим бајтом. Дужина инструкције је 3 бајта. Код РС релативног са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је 2 бајта. Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе меморијске локације дат другим а старији бајт трећим бајтом. Дужина инструкције је 3 бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С

удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15..0}$ и програмска статусна речи $PSW_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $IVTP_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

21 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине један бајт. Адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: меморијско директно адресирање (memdir), меморијско индиректно адресирање (memind), РС релативно са померајем (pcrel) и непосредно адресирање (immed). Код меморијског директног и меморијског индиректног адресирања трећи и четврти бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе дат трећим бајтом а млађи бајт адресе четвртим бајтом. Код меморијског индиректног адресирања адреса дужине 16 бита заузима две суседне меморијске локације, при чему је старији бајт адресе налази на нижој а млађи бајт адресе на вишој локацији. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта. Код РС релативно са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је три бајта. Код непосредног адресирања трећи бајт инструкције садржи 8 битни податак. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је три бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за

једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса 8 битне величине у акумулатор AB (LD), инструкција преноса 16 битне величине у акумулатор AW (LDW), инструкција преноса 8 битне величине из акумулатора AB (ST), инструкција преноса 16 битне величине из акумулатора AW (STW), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST и STW није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $\text{IVTP}_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

22 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине један бајт. Адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Бит 2 има вредност 0 за регистарско директно и регистарско индиректно адресирање, а вредност 1 за непосредно, РС релативно са померајем, меморијско директно и меморијско индиректно адресирање. Уколико бит 2 има вредност 0, тада се вредностима 0 и 1 бита 1 одређује регистарско директно (regdir) и регистарско индиректно адресирање (regind), респективно, а битом 0 специфицира регистар опште намене R[0] или R[1] који се користе за ова два адресирања. Дужина инструкције је 1 бајт. Уколико бит 2 има вредност 1, тада битови 1 и 0 одређује непосредно (immed), РС релативно са померајем (pcrel), меморијско директно (memdir) и меморијско индиректно адресирање (memind). Код непосредног адресирања други бајт инструкције садрже 8 битни податак. Дужина инструкције је 2 бајта. Код РС релативног са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је 2 бајта. Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе меморијске локације дат другим а млађи бајт трећим бајтом. Дужина инструкције је 3 бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на последњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за

једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	Z = 1
BNEQ	скок на неједнако	Z = 0
BNEG	скок на N = 1	N = 1
BNNG	скок на N = 0	N = 0
BOVF	скок на V = 1	V = 1
BNVF	скок на V = 0	V = 0
BCR	скок на C = 1	C = 1
BNCR	скок на C = 0	C = 0
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	C = 0
BLSSU	скок на мање него (без знака)	C = 1
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса 8 битне величине у акумулатор AB (LD), инструкција преноса 16 битне величине у акумулатор AW (LDW), инструкција преноса 8 битне величине из акумулатора AB (ST), инструкција преноса 16 битне величине из акумулатора AW (STW), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST и STW није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $\text{IVTP}_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

23 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 16 битна целобројна величина са знаком дата другим и трећим бајтом инструкције, при чему је млађи бајт дат другим а старији трећим бајтом инструкције. Дужина инструкције је три бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: регистарско директно адресирање (regdir), регистарско индиректно са померајем адресирање (regindrom), РС релативно са померајем (pcrel) и непосредно адресирање (immed). Код регистарског директног адресирања се користе неки од регистара опште намене R[0] до R[63] специфициран битовима 5 до 0 другог бајта инструкције. Дужина инструкција је два бајта. Код регистарског индиректног са померајем адресирања трећи и четврти бајт инструкције садрже 16 битни померај који је дат као целобројна величина са знаком, при чему је млађи бајт адресе дат трећим бајтом а старији бајт адресе четвртим бајтом. Један од регистара опште намене R[0] до R[63] који се користи специфициран је битовима 5 до 0 другог бајта инструкције. Дужина инструкција је четири бајта. Код РС релативно са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је три бајта. Код непосредног адресирања трећи и четврти бајт инструкције садрже 16 битни податак, при чему је млађи бајт податка дат трећим бајтом а старији бајт податка четвртим бајтом. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на последњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја

акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора А у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора А у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSW регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSW регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $\text{IVTP}_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0

уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$ највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

24 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Бит 2 има вредност 0 за регистарско директно и регистарско индиректно адресирање, а вредност 1 за непосредно, РС релативно са померајем, меморијско директно и меморијско индиректно адресирање. Уколико бит 2 има вредност 0, тада се вредностима 0 и 1 бита 1 одређује регистарско директно (regdir) и регистарско индиректно адресирање (regind), респективно, а битом 0 специфицира регистар опште намене R[0] или R[1] који се користе за ова два адресирања. Дужина инструкције је 1 бајт. Уколико бит 2 има вредност 1, тада битови 1 и 0 одређује непосредно (immed), РС релативно са померајем (pcrel), меморијско директно (memdir) и меморијско индиректно адресирање (memind). Код непосредног адресирања други и трећи бајт инструкције садрже 16 битни податак, при чему је млађи бајт податка дат другим бајтом а старији бајт податка трећим бајтом. Дужина инструкције је 3 бајта. Код РС релативног са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је 2 бајта. Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе меморијске локације дат другим а старији бајт трећим бајтом. Дужина инструкције је 3 бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С

удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограма (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15..0}$ и програмска статусна речи $PSW_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $IVTP_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

25 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине 1 бајт. Адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој а старији бајт на вишој адреси.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције условног скока Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0001 за све инструкције безусловног скока Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 1111 за безадресне инструкције. Битовима 3 до 0 првог бајта инструкције специфицира код операције за безадресне инструкције. Дужина инструкција је један бајт.

Битови 7, 6, 5 и 4 првог бајта инструкције у опсегу вредности 0010 до 1110 специфицирају код операције за адресне инструкције. Дужина инструкција је 1, 2, или 3 бајта и зависи од специфицираног начина адресирања.

Начини адресирања су специфицирани битовима 3 и 2 првог бајта инструкције. Процесор поседује следеће начине адресирања: непосредно адресирање (immed), меморијско директно адресирање (memdir), регистарско индиректно са померајем адресирање (regindrom) и регистарско директно адресирање (regdir). Код непосредног адресирања 8 битни операнд је дат другим бајтом инструкције. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 2 бајта. Код меморијског директног адресирања 16 битна адреса меморијске локације је дата другим и трећим бајтом инструкције, при чему је млађи бајт адресе дат другим а старији бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 3 бајта. Код регистарског индиректног адресирања са померајем 8 битни померај је целобројна величина са знаком у другом комплементу дата другим бајтом инструкције. Битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 2 бајта. Код регистарског директног адресирања битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 1 бајт.

Стек расте према нижим меморијским локацијама, а регистар SP указује на задњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR),

инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса 8 битне величине у акумулатор AB (LD), инструкција преноса 16 битне величине у акумулатор AW (LDW), инструкција преноса 8 битне величине из акумулатора AB (ST), инструкција преноса 16 битне величине из акумулатора AW (STW), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST и STW није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15...0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15...0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15...0}$ и програмска статусна речи $\text{PSW}_{15...0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15...0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна

величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

26 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине 2 бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој а млађи бајт на вишој адреси. У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције условног скока. Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0001 за све инструкције безусловног скока. Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 1111 за безадресне инструкције. Битовима 3 до 0 првог бајта инструкције специфицира код операције за безадресне инструкције. Дужина инструкција је један бајт.

Битови 7, 6, 5 и 4 првог бајта инструкције у опсегу вредности 0010 до 1110 специфицирају код операције за адресне инструкције. Дужина инструкција је 1, 2, или 3 бајта и зависи од специфицираног начина адресирања.

Начини адресирања су специфицирани битовима 3 и 2 првог бајта инструкције. Процесор поседује следеће начине адресирања: меморијско директно адресирање (memdir), меморијско индиректно адресирање (memind), регистарско директно адресирање (regdir) и непосредно адресирање (immed). Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе дат другим бајтом а млађи бајт адресе трећим бајтом. Код меморијског индиректног адресирања адреса дужине 16 бита заузима две суседне меморијске локације, при чему је старији бајт адресе налази на нижој а млађи бајт адресе на вишој локацији. Битови 1 до 0 другог бајта инструкције се не користе. Дужина инструкција је три бајта. Код регистарског директног адресирања битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 1 бајт. Код непосредног адресирања 16 битни операнд је дат другим и трећим бајтом инструкције, при чему је старији бајт операнда дат другим а млађи бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 3 бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR),

инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15..0}$ и програмска статусна речи $PSW_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15..0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$ највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у

померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

27 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине један бајт. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Бит 2 има вредност 0 за регистарско директно и регистарско индиректно адресирање, а вредност 1 за непосредно, РС релативно са померајем, меморијско директно и меморијско индиректно адресирање. Уколико бит 2 има вредност 0, тада се вредностима 0 и 1 бита 1 одређује регистарско директно (*regdir*) и регистарско индиректно адресирање (*regind*), респективно, а битом 0 специфицира регистар опште намене $R[0]$ или $R[1]$ који се користе за ова два адресирања. Дужина инструкције је 1 бајт. Уколико бит 2 има вредност 1, тада битови 1 и 0 одређује непосредно (*immed*), РС релативно са померајем (*pcrel*), меморијско директно (*memdir*) и меморијско индиректно адресирање (*memind*). Код непосредног адресирања други бајт инструкције садрже 8 битни податак. Дужина инструкције је 2 бајта. Код РС релативног са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је 2 бајта. Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе меморијске локације дат другим а старији бајт трећим бајтом. Дужина инструкције је 3 бајта.

Стек расте према вишим меморијским локацијама, а регистар *SP* указује на последњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (*RTS*), инструкција повратка из прекидне рутине (*RTI*), инструкција аритметичког померања садржаја акумулатора удесно за једно место (*ASR*), инструкција логичког померања садржаја акумулатора удесно за једно место (*LSR*), инструкција ротирања садржаја акумулатора удесно за једно место (*ROR*), инструкција ротирања садржаја акумулатора и индикатора *C* удесно за једно место (*RORC*), инструкција аритметичког померања садржаја акумулатора

улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса 8 битне величине у акумулатор AB (LD), инструкција преноса 16 битне величине у акумулатор AW (LDW), инструкција преноса 8 битне величине из акумулатора AB (ST), инструкција преноса 16 битне величине из акумулатора AW (STW), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST и STW није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $\text{IVTP}_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво

приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$ највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15..0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15..0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

28 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине 1 бајт. Адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој а млађи бајт на вишој адреси.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције условног скока Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0001 за све инструкције безусловног скока Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 1111 за безадресне инструкције. Битовима 3 до 0 првог бајта инструкције специфицира код операције за безадресне инструкције. Дужина инструкција је један бајт.

Битови 7, 6, 5 и 4 првог бајта инструкције у опсегу вредности 0010 до 1110 специфицирају код операције за адресне инструкције. Дужина инструкција је 1, 2, или 3 бајта и зависи од специфицираног начина адресирања.

Начини адресирања су специфицирани битовима 3 и 2 првог бајта инструкције. Процесор поседује следеће начине адресирања: регистарско индиректно са померајем адресирање (regindrom), регистарско директно адресирање (regdir), РС релативно са померајем (pcrel) и непосредно адресирање (immed). Код регистарског индиректног адресирања са померајем 16 битни померај је целобројна величина са знаком у другом комплементу дата другим и трећим бајтом инструкције при чему је старији бајт дат другим бајтом а млађи бајт помераја трећим бајтом. Битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је три бајта. Код регистарског директног адресирања битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 1 бајт. Код РС релативно са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је два бајта. Код непосредног адресирања 8 битни операнд је дат другим бајтом инструкције. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 2 бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR),

инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса 8 битне величине у акумулатор AB (LD), инструкција преноса 16 битне величине у акумулатор AW (LDW), инструкција преноса 8 битне величине из акумулатора AB (ST), инструкција преноса 16 битне величине из акумулатора AW (STW), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST и STW није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15...0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15...0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15...0}$ и програмска статусна речи $\text{PSW}_{15...0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15...0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна

величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

29 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 16 битна целобројна величина са знаком дата другим и трећим бајтом инструкције, при чему је млађи бајт дат другим а старији трећим бајтом инструкције. Дужина инструкције је три бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: регистарско директно адресирање (regdir), регистарско индиректно са померајем адресирање (regindpom), РС релативно са померајем (pcrel) и непосредно адресирање (immed). Код регистарског директног адресирања се користе неки од регистара опште намене R[0] до R[63] специфициран битовима 5 до 0 другог бајта инструкције. Дужина инструкција је два бајта. Код регистарског индиректног са померајем адресирања трећи и четврти бајт инструкције садрже 16 битни померај који је дат као целобројна величина са знаком, при чему је млађи бајт адресе дат трећим бајтом а старији бајт адресе четвртим бајтом. Један од регистара опште намене R[0] до R[63] који се користи специфициран је битовима 5 до 0 другог бајта инструкције. Дужина инструкција је четири бајта. Код РС релативно са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је три бајта. Код непосредног адресирања трећи бајт инструкције садржи 8 битни податак. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је три бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на последњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С

удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограма (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $\text{IVTP}_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

30 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Бит 2 има вредност 0 за регистарско директно и регистарско индиректно адресирање, а вредност 1 за непосредно, РС релативно са померајем, меморијско директно и меморијско индиректно адресирање. Уколико бит 2 има вредност 0, тада се вредностима 0 и 1 бита 1 одређује регистарско директно (regdir) и регистарско индиректно адресирање (regind), респективно, а битом 0 специфицира регистар опште намене R[0] или R[1] који се користе за ова два адресирања. Дужина инструкције је 1 бајт. Уколико бит 2 има вредност 1, тада битови 1 и 0 одређује непосредно (immed), РС релативно са померајем (pcrel), меморијско директно (memdir) и меморијско индиректно адресирање (memind), респективно Код непосредног адресирања други и трећи бајт инструкције садрже 16 битни податак, при чему је млађи бајт податка дат другим бајтом а старији бајт податка трећим бајтом. Дужина инструкције је 3 бајта. Код РС релативног са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је 2 бајта. Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе меморијске локације дат другим а старији бајт трећим бајтом. Дужина инструкције је 3 бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С

удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15..0}$ и програмска статусна речи $PSW_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $IVTP_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

31 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 16 битна целобројна величина са знаком дата другим и трећим бајтом инструкције, при чему је млађи бајт дат другим а старији трећим бајтом инструкције. Дужина инструкције је три бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: меморијско директно адресирање (*memdir*), меморијско индиректно адресирање (*memind*), РС релативно са померајем (*pcrel*) и непосредно адресирање (*immed*). Код меморијског директног и меморијског индиректног адресирања трећи и четврти бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе дат трећим бајтом а старији бајт адресе четвртим бајтом. Код меморијског индиректног адресирања адреса дужине 16 бита заузима две суседне меморијске локације, при чему је млађи бајт адресе налази на нижој а старији бајт адресе на вишој локацији. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта. Код РС релативно са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је три бајта. Код непосредног адресирања трећи бајт инструкције садржи 8 битни податак. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је три бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на последњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (*RTS*), инструкција повратка из прекидне рутине (*RTI*), инструкција аритметичког померања садржаја акумулатора удесно за једно место (*ASR*), инструкција логичког померања садржаја акумулатора удесно за једно место (*LSR*), инструкција ротирања садржаја акумулатора удесно за једно место (*ROR*), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (*RORC*), инструкција аритметичког померања садржаја акумулатора

улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограма (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $\text{IVTP}_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна

величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

32 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Уколико бит 2 има вредност 0, ради се о регистарском директном адресирању (`regdir`) код кога се користи један од регистара опште намене $R[0]$ и $R[1]$, или базни регистар $BR_{15..0}$ или индексни регистар $XR_{15..0}$ што је специфициран вредностима 00, 01, 10 и 11 битова 1 и 0. Дужина инструкције је један бајт. Уколико бит 2 има вредност 1, ради се о меморијском директном (`memdir`), базном адресирању са померајем (`base`), индексном адресирању са померајем (`index`) или о базно индексном адресирању са померајем (`baseindex`) специфицираном вредностима битова 1 и 0. Код меморијског адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе дат другим а старији бајт адресе трећим бајтом инструкције. Дужина инструкције је три бајта. Код базног адресирања са померајем имплицитно се користи базни регистар $BR_{15..0}$, док други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је два бајта. Код индексног адресирања са померајем имплицитно се користи индексни регистар $XR_{15..0}$, док други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је два бајта. Код базно индексног адресирања са померајем имплицитно се користе регистри $BR_{15..0}$ и $XR_{15..0}$, а други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је 2 бајта.

Стек расте према нижим меморијским локацијама, а регистар `SP` указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (`RTS`), инструкција повратка из прекидне рутине (`RTI`), инструкција аритметичког померања садржаја акумулатора удесно за једно место (`ASR`), инструкција логичког померања садржаја

акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSW регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSW регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15..0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0

уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$ највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

33 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Бит 2 има вредност 0 за регистарско директно и регистарско индиректно адресирање, а вредност 1 за непосредно, РС релативно са померајем, меморијско директно и меморијско индиректно адресирање. Уколико бит 2 има вредност 0, тада се вредностима 0 и 1 бита 1 одређује регистарско директно (regdir) и регистарско индиректно адресирање (regind), респективно, а битом 0 специфицира регистар опште намене R[0] или R[1] који се користе за ова два адресирања. Дужина инструкције је 1 бајт. Уколико бит 2 има вредност 1, тада битови 1 и 0 одређује непосредно (immed), РС релативно са померајем (pcrel), меморијско директно (memdir) и меморијско индиректно адресирање (memind). Код непосредног адресирања други и трећи бајт инструкције садрже 16 битни податак, при чему је старији бајт податка дат другим бајтом а млађи бајт податка трећим бајтом. Дужина инструкције је 3 бајта. Код РС релативног са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је 2 бајта. Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе меморијске локације дат другим а млађи бајт трећим бајтом. Дужина инструкције је 3 бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С

удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограма (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15..0}$ и програмска статусна речи $PSW_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $IVTP_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

34 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: регистарско директно адресирање (regdir), регистарско индиректно са померајем адресирање (regindpom), РС релативно са померајем (pcrel) и непосредно адресирање (immed). Код регистарског директног адресирања се користе неки од регистара опште намене R[0] до R[63] специфициран битовима 5 до 0 другог бајта инструкције. Дужина инструкција је два бајта. Код регистарског индиректног са померајем адресирања трећи и четврти бајт инструкције садрже 16 битни померај који је дат као целобројна величина са знаком, при чему је старији бајт адресе дат трећим бајтом а млађи бајт адресе четвртим бајтом. Један од регистара опште намене R[0] до R[63] који се користи специфициран је битовима 5 до 0 другог бајта инструкције. Дужина инструкција је четири бајта. Код РС релативно са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је три бајта. Код непосредног адресирања трећи бајт инструкције садржи 8 битни податак. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је три бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С

удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограма (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15..0}$ и програмска статусна речи $PSW_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $IVTP_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

35 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Уколико бит 2 има вредност 0, ради се о регистарском директном адресирању (regdir) код кога се користи један од регистара опште намене $R[0]$ и $R[1]$, или базни регистар $BR_{15...0}$ или индексни регистар $XR_{15...0}$ што је специфициран вредностима 00, 01, 10 и 11 битова 1 и 0. Дужина инструкције је један бајт. Уколико бит 2 има вредност 1, ради се о непосредном (immed), базном адресирању са померајем (base), индексном адресирању са померајем (index) или о базно индексном адресирању са померајем (baseindex) специфицираном вредностима битова 1 и 0. Код непосредног адресирања други и трећи бајт инструкције садрже 16 битни податак, при чему је млађи бајт податка дат другим а старији бајт податка трећим бајтом инструкције. Дужина инструкције је три бајта. Код базног адресирања са померајем имплицитно се користи базни регистар $BR_{15...0}$, док други и трећи бајт инструкције садржи 16 битни померај који је дат као целобројна величина са знаком, при чему је млађи бајт помераја дат другим а старији бајт адресе трећим бајтом инструкције. Дужина инструкције је три бајта. Код индексног адресирања са померајем имплицитно се користи индексни регистар $XR_{15...0}$, док други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је два бајта. Код базно индексног адресирања са померајем имплицитно се користе регистри $BR_{15...0}$ и $XR_{15...0}$, а други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је 2 бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја

аккумулятора удесно за једно место (ASR), инструкција логичког померања садржаја аккумулятора удесно за једно место (LSR), инструкција ротирања садржаја аккумулятора удесно за једно место (ROR), инструкција ротирања садржаја аккумулятора и индикатора C удесно за једно место (RORC), инструкција аритметичког померања садржаја аккумулятора улево за једно место (ASL), инструкција логичког померања садржаја аккумулятора улево за једно место (LSR), инструкција ротирања садржаја аккумулятора улево за једно место (ROR), инструкција ротирања садржаја аккумулятора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај аккумулятора A у регистар IVTP (STIVTP) и инструкција преноса садржај аккумулятора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у аккумулятор (LD), инструкција преноса из аккумулятора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15..0}$ и програмска статусна речи $PSW_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $IVTP_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који

долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$ највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

36 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине један бајта. Адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: регистарско директно адресирање (regdir), регистарско индиректно адресирање (regind), меморијско директно адресирање (memdir), регистарско индиректно са померајем адресирање (regindprom). Код регистарског директног и регистарског индиректног адресирања се користе неки од регистара опште намене R[0] до R[63] специфициран битовима 5 до 0 другог бајта инструкције. Дужина инструкција је два бајта. Код меморијског директног адресирања трећи и четврти бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе дат трећим бајтом а млађи бајт адресе четвртим бајтом. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта. Код регистарског индиректног са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Један од регистара опште намене R[0] до R[63] који се користи специфициран је битовима 5 до 0 другог бајта инструкције. Дужина инструкција је три бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за

једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	Z = 1
BNEQ	скок на неједнако	Z = 0
BNEG	скок на N = 1	N = 1
BNNG	скок на N = 0	N = 0
BOVF	скок на V = 1	V = 1
BNVF	скок на V = 0	V = 0
BCR	скок на C = 1	C = 1
BNCR	скок на C = 0	C = 0
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	C = 0
BLSSU	скок на мање него (без знака)	C = 1
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса 8 битне величине у акумулатор AB (LD), инструкција преноса 16 битне величине у акумулатор AW (LDW), инструкција преноса 8 битне величине из акумулатора AB (ST), инструкција преноса 16 битне величине из акумулатора AW (STW), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST и STW није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $\text{IVTP}_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

37 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине један бајт. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Бит 2 има вредност 0 за регистарско директно и регистарско индиректно адресирање, а вредност 1 за непосредно, РС релативно са померајем, меморијско директно и меморијско индиректно адресирање. Уколико бит 2 има вредност 0, тада се вредностима 0 и 1 бита 1 одређује регистарско директно (regdir) и регистарско индиректно адресирање (regind), респективно, а битом 0 специфицира регистар опште намене R[0] или R[1] који се користе за ова два адресирања. Дужина инструкције је 1 бајт. Уколико бит 2 има вредност 1, тада битови 1 и 0 одређује непосредно (immed), РС релативно са померајем (pcrel), меморијско директно (memdir) и меморијско индиректно адресирање (memind), респективно Код непосредног адресирања други бајт инструкције садржи 8 битни податак. Дужина инструкције је 2 бајта. Код РС релативног са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је 2 бајта. Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе меморијске локације дат другим а старији бајт трећим бајтом. Дужина инструкције је 3 бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на последњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора

улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса 8 битне величине у акумулатор AB (LD), инструкција преноса 16 битне величине у акумулатор AW (LDW), инструкција преноса 8 битне величине из акумулатора AB (ST), инструкција преноса 16 битне величине из акумулатора AW (STW), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST и STW није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15...0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15...0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15...0}$ и програмска статусна речи $PSW_{15...0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $IVTP_{15...0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво

приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$ највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15..0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15..0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

38 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: регистарско директно адресирање (regdir), регистарско индиректно са померајем адресирање (regindpom), РС релативно са померајем (pcrel) и непосредно адресирање (immed). Код регистарског директног адресирања се користе неки од регистара опште намене R[0] до R[63] специфициран битовима 5 до 0 другог бајта инструкције. Дужина инструкција је два бајта. Код регистарског индиректног са померајем адресирања трећи и четврти бајт инструкције садрже 16 битни померај који је дат као целобројна величина са знаком, при чему је старији бајт адресе дат трећим бајтом а млађи бајт адресе четвртим бајтом. Један од регистара опште намене R[0] до R[63] који се користи специфициран је битовима 5 до 0 другог бајта инструкције. Дужина инструкција је четири бајта. Код РС релативно са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је три бајта. Код непосредног адресирања трећи и четврти бајт инструкције садрже 16 битни податак, при чему је старији бајт податка дат трећим бајтом а млађи бајт податка четвртим бајтом. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора

удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C
удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора
улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за
једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR),
инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC),
инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на
0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и
инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока
на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из
акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција
одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка
инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У
инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање
инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$.
Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и
маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не
реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи
 $PSW_{15...0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се
реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред
PSWI регистра $PSW_{15...0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15...0}$ и програмска
статусна речи $PSW_{15...0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине.
Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса
прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV
табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на
коју указује садржај регистра $IVTP_{15...0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са
8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који
долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$
уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво

приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$ највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15..0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15..0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

39 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине један бајт. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Бит 2 има вредност 0 за регистарско директно и регистарско индиректно адресирање, а вредност 1 за непосредно, РС релативно са померајем, меморијско директно и меморијско индиректно адресирање. Уколико бит 2 има вредност 0, тада се вредностима 0 и 1 бита 1 одређује регистарско директно (regdir) и регистарско индиректно адресирање (regind), респективно, а битом 0 специфицира регистар опште намене R[0] или R[1] који се користе за ова два адресирања. Дужина инструкције је 1 бајт. Уколико бит 2 има вредност 1, тада битови 1 и 0 одређује непосредно (immed), РС релативно са померајем (pcrel), меморијско директно (memdir) и меморијско индиректно адресирање (memind). Код непосредног адресирања други бајт инструкције садрже 8 битни податак. Дужина инструкције је 2 бајта. Код РС релативног са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је 2 бајта. Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе меморијске локације дат другим а старији бајт трећим бајтом. Дужина инструкције је 3 бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на последњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора

улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса 8 битне величине у акумулатор AB (LD), инструкција преноса 16 битне величине у акумулатор AW (LDW), инструкција преноса 8 битне величине из акумулатора AB (ST), инструкција преноса 16 битне величине из акумулатора AW (STW), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST и STW није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $\text{IVTP}_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво

приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$ највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15..0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15..0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

40 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине 2 бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој а старији бајт на вишој адреси. У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције условног скока Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0001 за све инструкције безусловног скока Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 1111 за безадресне инструкције. Битовима 3 до 0 првог бајта инструкције специфицира код операције за безадресне инструкције. Дужина инструкција је један бајт.

Битови 7, 6, 5 и 4 првог бајта инструкције у опсегу вредности 0010 до 1110 специфицирају код операције за адресне инструкције. Дужина инструкција је 1, 2, или 3 бајта и зависи од специфицираног начина адресирања.

Начини адресирања су специфицирани битовима 3 и 2 првог бајта инструкције. Процесор поседује следеће начине адресирања: регистарско индиректно са померајем адресирање (regindrom), регистарско директно адресирање (regdir), РС релативно са померајем (pcrel) и непосредно адресирање (immed). Код регистарског индиректног адресирања са померајем 16 битни померај је целобројна величина са знаком у другом комплементу дата другим и трећим бајтом инструкције при чему је старији бајт дат другим бајтом а млађи бајт помераја трећим бајтом. Битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је три бајта. Код регистарског директног адресирања битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 1 бајт. Код РС релативно са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је два бајта. Код непосредног адресирања 16 битни операнд је дат другим и трећим бајтом инструкције, при чему је млађи бајт операнда дат другим а старији бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 3 бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на задњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за

једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	Z = 1
BNEQ	скок на неједнако	Z = 0
BNEG	скок на N = 1	N = 1
BNNG	скок на N = 0	N = 0
BOVF	скок на V = 1	V = 1
BNVF	скок на V = 0	V = 0
BCR	скок на C = 1	C = 1
BNCR	скок на C = 0	C = 0
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	C = 0
BLSSU	скок на мање него (без знака)	C = 1
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $\text{IVTP}_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по

две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

41 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: регистарско директно адресирање (regdir), регистарско индиректно адресирање (regind), меморијско директно адресирање (memdir), регистарско индиректно са померајем адресирање (regindpom). Код регистарског директног и регистарског индиректног адресирања се користе неки од регистара опште намене R[0] до R[63] специфициран битовима 5 до 0 другог бајта инструкције. Дужина инструкција је два бајта. Код меморијског директног адресирања трећи и четврти бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе дат трећим бајтом а млађи бајт адресе четвртим бајтом. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта. Код регистарског индиректног са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Један од регистара опште намене R[0] до R[63] који се користи специфициран је битовима 5 до 0 другог бајта инструкције. Дужина инструкција је три бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора

улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограма (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15..0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна

величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

42 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Уколико бит 2 има вредност 0, ради се о регистарском директном адресирању (`regdir`) код кога се користи један од регистара опште намене $R[0]$ и $R[1]$, или базни регистар $BR_{15..0}$ или индексни регистар $XR_{15..0}$ што је специфициран вредностима 00, 01, 10 и 11 битова 1 и 0. Дужина инструкције је један бајт. Уколико бит 2 има вредност 1, ради се о меморијском директном (`memdir`), базном адресирању са померајем (`base`), индексном адресирању са померајем (`index`) или о базно индексном адресирању са померајем (`baseindex`) специфицираном вредностима битова 1 и 0. Код меморијског адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе дат другим а млађи бајт адресе трећим бајтом инструкције. Дужина инструкције је три бајта. Код базног адресирања са померајем имплицитно се користи базни регистар $BR_{15..0}$, док други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је два бајта. Код индексног адресирања са померајем имплицитно се користи индексни регистар $XR_{15..0}$, док други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је два бајта. Код базно индексног адресирања са померајем имплицитно се користе регистри $BR_{15..0}$ и $XR_{15..0}$, а други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је 2 бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја

акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора А у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора А у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSW регистра програмске статусне речи $PSW_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSW регистра $PSW_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15..0}$ и програмска статусна речи $PSW_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15..0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$

уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$ највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

43 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине 2 бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој а млађи бајт на вишој адреси. У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције условног скока. Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0001 за све инструкције безусловног скока. Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 1111 за безадресне инструкције. Битовима 3 до 0 првог бајта инструкције специфицира код операције за безадресне инструкције. Дужина инструкција је један бајт.

Битови 7, 6, 5 и 4 првог бајта инструкције у опсегу вредности 0010 до 1110 специфицирају код операције за адресне инструкције. Дужина инструкција је 1, 2, или 3 бајта и зависи од специфицираног начина адресирања.

Начини адресирања су специфицирани битовима 3 и 2 првог бајта инструкције. Процесор поседује следеће начине адресирања: непосредно адресирање (immed), меморијско директно адресирање (memdir), регистарско индиректно са померајем адресирање (regindrom) и регистарско директно адресирање (regdir). Код непосредног адресирања 16 битни операнд је дат другим и трећим бајтом инструкције, при чему је старији бајт операнда дат другим а млађи бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 3 бајта. Код меморијског директног адресирања 16 битна адреса меморијске локације је дата другим и трећим бајтом инструкције, при чему је старији бајт адресе дат другим а млађи бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 3 бајта. Код регистарског индиректног адресирања са померајем 8 битни померај је целобројна величина са знаком у другом комплементу дата другим бајтом инструкције. Битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 2 бајта. Код регистарског директног адресирања битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 1 бајт.

Стек расте према вишим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за

једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	Z = 1
BNEQ	скок на неједнако	Z = 0
BNEG	скок на N = 1	N = 1
BNNG	скок на N = 0	N = 0
BOVF	скок на V = 1	V = 1
BNVF	скок на V = 0	V = 0
BCR	скок на C = 1	C = 1
BNCR	скок на C = 0	C = 0
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	C = 0
BLSSU	скок на мање него (без знака)	C = 1
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $\text{IVTP}_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по

две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

44 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 16 битна целобројна величина са знаком дата другим и трећим бајтом инструкције, при чему је млађи бајт дат другим а старији трећим бајтом инструкције. Дужина инструкције је три бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: регистарско директно адресирање (regdir), регистарско индиректно адресирање (regind), меморијско директно адресирање (memdir), регистарско индиректно са померајем адресирање (regindpom). Код регистарског директног и регистарског индиректног адресирања се користе неки од регистара опште намене R[0] до R[63] специфициран битовима 5 до 0 другог бајта инструкције. Дужина инструкција је два бајта. Код меморијског директног адресирања трећи и четврти бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе дат трећим бајтом а старији бајт адресе четвртим бајтом. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта. Код регистарског индиректног са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Један од регистара опште намене R[0] до R[63] који се користи специфициран је битовима 5 до 0 другог бајта инструкције. Дужина инструкција је три бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на последњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С

удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15..0}$ и програмска статусна речи $PSW_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $IVTP_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

45 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Уколико бит 2 има вредност 0, ради се о регистарском директном адресирању (regdir) код кога се користи један од регистара опште намене R[0], R[1], R[2] и R[3] специфициран вредностима 00, 01, 10 и 11 битова 1 и 0. Дужина инструкције је један бајт. Уколико бит 2 има вредност 1, ради се о непосредном (immed), РС релативном са померајем (pcrel), меморијском директном (memdir) или меморијском индиректном (memind) адресирању специфицираном вредностима битова 1 и 0. Код непосредног адресирања други и трећи бајт инструкције садрже 16 битни податак, при чему је старији бајт податка дат другим а млађи бајт податка трећим бајтом инструкције. Дужина инструкције је три бајта. Код РС релативног са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је два бајта. Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе дат другим а млађи бајт адресе трећим бајтом инструкције. Дужина инструкције је три бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за

једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	Z = 1
BNEQ	скок на неједнако	Z = 0
BNEG	скок на N = 1	N = 1
BNNG	скок на N = 0	N = 0
BOVF	скок на V = 1	V = 1
BNVF	скок на V = 0	V = 0
BCR	скок на C = 1	C = 1
BNCR	скок на C = 0	C = 0
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	C = 0
BLSSU	скок на мање него (без знака)	C = 1
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $\text{IVTP}_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по

две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

46 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: регистарско директно адресирање (regdir), регистарско индиректно адресирање (regind), меморијско директно адресирање (memdir), регистарско индиректно са померајем адресирање (regindpom). Код регистарског директног и регистарског индиректног адресирања се користе неки од регистара опште намене R[0] до R[63] специфициран битовима 5 до 0 другог бајта инструкције. Дужина инструкција је два бајта. Код меморијског директног адресирања трећи и четврти бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе дат трећим бајтом а млађи бајт адресе четвртим бајтом. Битови 5 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта. Код регистарског индиректног са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Један од регистара опште намене R[0] до R[63] који се користи специфициран је битовима 5 до 0 другог бајта инструкције. Дужина инструкција је три бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора

улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограма (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15..0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна

величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

47 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине 1 бајт. Адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој а млађи бајт на вишој адреси.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције условног скока Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0001 за све инструкције безусловног скока Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 1111 за безадресне инструкције. Битовима 3 до 0 првог бајта инструкције специфицира код операције за безадресне инструкције. Дужина инструкција је један бајт.

Битови 7, 6, 5 и 4 првог бајта инструкције у опсегу вредности 0010 до 1110 специфицирају код операције за адресне инструкције. Дужина инструкција је 1, 2, или 3 бајта и зависи од специфицираног начина адресирања.

Начини адресирања су специфицирани битовима 3 и 2 првог бајта инструкције. Процесор поседује следеће начине адресирања: непосредно адресирање (immed), меморијско директно адресирање (memdir), регистарско индиректно са померајем адресирање (regindrom) и регистарско директно адресирање (regdir). Код непосредног адресирања 8 битни операнд је дат другим бајтом инструкције. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 2 бајта. Код меморијског директног адресирања 16 битна адреса меморијске локације је дата другим и трећим бајтом инструкције, при чему је старији бајт адресе дат другим а млађи бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 3 бајта. Код регистарског индиректног адресирања са померајем 8 битни померај је целобројна величина са знаком у другом комплементу дата другим бајтом инструкције. Битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 2 бајта. Код регистарског директног адресирања битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 1 бајт.

Стек расте према вишим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR),

инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса 8 битне величине у акумулатор AB (LD), инструкција преноса 16 битне величине у акумулатор AW (LDW), инструкција преноса 8 битне величине из акумулатора AB (ST), инструкција преноса 16 битне величине из акумулатора AW (STW), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST и STW није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15..0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна

величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

48 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: меморијско директно адресирање (*memdir*), меморијско индиректно адресирање (*memind*), РС релативно са померајем (*pcrel*) и непосредно адресирање (*immed*). Код меморијског директног и меморијског индиректног адресирања трећи и четврти бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе дат трећим бајтом а млађи бајт адресе четвртим бајтом. Код меморијског индиректног адресирања адреса дужине 16 бита заузима две суседне меморијске локације, при чему је старији бајт адресе налази на нижој а млађи бајт адресе на вишој локацији. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта. Код РС релативно са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је три бајта. Код непосредног адресирања трећи и четврти бајт инструкције садрже 16 битни податак, при чему је старији бајт податка дат трећим бајтом а млађи бајт податка четвртим бајтом. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С

удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15..0}$ и програмска статусна речи $PSW_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $IVTP_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

49 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине 2 бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој а млађи бајт на вишој адреси. У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције условног скока. Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0001 за све инструкције безусловног скока. Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 1111 за безадресне инструкције. Битовима 3 до 0 првог бајта инструкције специфицира код операције за безадресне инструкције. Дужина инструкција је један бајт.

Битови 7, 6, 5 и 4 првог бајта инструкције у опсегу вредности 0010 до 1110 специфицирају код операције за адресне инструкције. Дужина инструкција је 1, 2, или 3 бајта и зависи од специфицираног начина адресирања.

Начини адресирања су специфицирани битовима 3 и 2 првог бајта инструкције. Процесор поседује следеће начине адресирања: меморијско директно адресирање (memdir), меморијско индиректно адресирање (memind), регистарско директно адресирање (regdir) и непосредно адресирање (immed). Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе дат другим бајтом а млађи бајт адресе трећим бајтом. Код меморијског индиректног адресирања адреса дужине 16 бита заузима две суседне меморијске локације, при чему је старији бајт адресе налази на нижој а млађи бајт адресе на вишој локацији. Битови 1 до 0 другог бајта инструкције се не користе. Дужина инструкција је три бајта. Код регистарског директног адресирања битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 1 бајт. Код непосредног адресирања 16 битни операнд је дат другим и трећим бајтом инструкције, при чему је старији бајт операнда дат другим а млађи бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 3 бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR),

инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15..0}$ и програмска статусна речи $PSW_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15..0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$ највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у

померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

50 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине један бајт. Адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Бит 2 има вредност 0 за регистарско директно и регистарско индиректно адресирање, а вредност 1 за непосредно, РС релативно са померајем, меморијско директно и меморијско индиректно адресирање. Уколико бит 2 има вредност 0, тада се вредностима 0 и 1 бита 1 одређује регистарско директно (regdir) и регистарско индиректно адресирање (regind), респективно, а битом 0 специфицира регистар опште намене R[0] или R[1] који се користе за ова два адресирања. Дужина инструкције је 1 бајт. Уколико бит 2 има вредност 1, тада битови 1 и 0 одређује непосредно (immed), РС релативно са померајем (pcrel), меморијско директно (memdir) и меморијско индиректно адресирање (memind), респективно Код непосредног адресирања други бајт инструкције садржи 8 битни податак. Дужина инструкције је 2 бајта. Код РС релативног са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је 2 бајта. Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе меморијске локације дат другим а млађи бајт трећим бајтом. Дужина инструкције је 3 бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на последњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за

једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса 8 битне величине у акумулатор AB (LD), инструкција преноса 16 битне величине у акумулатор AW (LDW), инструкција преноса 8 битне величине из акумулатора AB (ST), инструкција преноса 16 битне величине из акумулатора AW (STW), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST и STW није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $\text{IVTP}_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

51 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 16 битна целобројна величина са знаком дата другим и трећим бајтом инструкције, при чему је млађи бајт дат другим а старији трећим бајтом инструкције. Дужина инструкције је три бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкције инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: меморијско директно адресирање (memdir), меморијско индиректно адресирање (memind), РС релативно са померајем (pcrel) и непосредно адресирање (immed). Код меморијског директног и меморијског индиректног адресирања трећи и четврти бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе дат трећим бајтом а старији бајт адресе четвртим бајтом. Код меморијског индиректног адресирања адреса дужине 16 бита заузима две суседне меморијске локације, при чему је млађи бајт адресе налази на нижој а старији бајт адресе на вишој локацији. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта. Код РС релативно са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је три бајта. Код непосредног адресирања трећи и четврти бајт инструкције садрже 16 битни податак, при чему је млађи бајт податка дат трећим бајтом а старији бајт податка четвртим бајтом. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на последњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора

удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C
удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора
улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за
једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR),
инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC),
инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на
0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и
инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока
на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из
акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција
одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка
инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У
инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање
инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$.
Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и
маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не
реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи
 $PSW_{15...0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се
реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред
PSWI регистра $PSW_{15...0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15...0}$ и програмска
статусна речи $PSW_{15...0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине.
Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса
прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV
табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на
коју указује садржај регистра $IVTP_{15...0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са
8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који
долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$
уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво

приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$ највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15..0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15..0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

52 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине један бајт. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за безадресне инструкције и инструкције скока, док бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и вредност 1 за инструкције скока. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт. Бит 5 првог бајта инструкције има вредност 0 за инструкције условног скока и 1 за инструкције безусловног скока. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција условног скока. Инструкција условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата 2. бајтом инструкције. Дужина инструкција је 2 бајта. Битовима 4 до 0 првог бајта инструкција специфицира се код операције за инструкција безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Битовима 6 до 3 првог бајта инструкција специфицира се код операције. Дужина инструкција је 1, 2 или 3 бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 2, 1 и 0 првог бајта инструкције специфицира начин адресирања и регистар опште намене уколико се користи у задатом начину адресирања. Бит 2 има вредност 0 за регистарско директно и регистарско индиректно адресирање, а вредност 1 за непосредно, РС релативно са померајем, меморијско директно и меморијско индиректно адресирање. Уколико бит 2 има вредност 0, тада се вредностима 0 и 1 бита 1 одређује регистарско директно (*regdir*) и регистарско индиректно адресирање (*regind*), респективно, а битом 0 специфицира регистар опште намене $R[0]$ или $R[1]$ који се користе за ова два адресирања. Дужина инструкције је 1 бајт. Уколико бит 2 има вредност 1, тада битови 1 и 0 одређује непосредно (*immed*), РС релативно са померајем (*pcrel*), меморијско директно (*memdir*) и меморијско индиректно адресирање (*memind*), респективно Код непосредног адресирања други бајт инструкције садржи 8 битни податак. Дужина инструкције је 2 бајта. Код РС релативног са померајем адресирања други бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкције је 2 бајта. Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе меморијске локације дат другим а старији бајт трећим бајтом. Дужина инструкције је 3 бајта.

Стек расте према вишим меморијским локацијама, а регистар *SP* указује на последњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (*RTS*), инструкција повратка из прекидне рутине (*RTI*), инструкција аритметичког померања садржаја акумулатора удесно за једно место (*ASR*), инструкција логичког померања садржаја акумулатора удесно за једно место (*LSR*), инструкција ротирања садржаја акумулатора удесно за једно место (*ROR*), инструкција ротирања садржаја акумулатора и индикатора *C*

удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса 8 битне величине у акумулатор AB (LD), инструкција преноса 16 битне величине у акумулатор AW (LDW), инструкција преноса 8 битне величине из акумулатора AB (ST), инструкција преноса 16 битне величине из акумулатора AW (STW), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST и STW није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSW регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSW регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15..0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0

уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$ највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15..0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15..0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

53 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 16 битна целобројна величина са знаком дата другим и трећим бајтом инструкције, при чему је млађи бајт дат другим а старији трећим бајтом инструкције. Дужина инструкције је три бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: меморијско директно адресирање (*memdir*), меморијско индиректно адресирање (*memind*), РС релативно са померајем (*pcrel*) и непосредно адресирање (*immed*). Код меморијског директног и меморијског индиректног адресирања трећи и четврти бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе дат трећим бајтом а старији бајт адресе четвртим бајтом. Код меморијског индиректног адресирања адреса дужине 16 бита заузима две суседне меморијске локације, при чему је млађи бајт адресе налази на нижој а старији бајт адресе на вишој локацији. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта. Код РС релативно са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је три бајта. Код непосредног адресирања трећи бајт инструкције садржи 8 битни податак. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је три бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на последњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (*RTS*), инструкција повратка из прекидне рутине (*RTI*), инструкција аритметичког померања садржаја акумулатора удесно за једно место (*ASR*), инструкција логичког померања садржаја акумулатора удесно за једно место (*LSR*), инструкција ротирања садржаја акумулатора удесно за једно место (*ROR*), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (*RORC*), инструкција аритметичког померања садржаја акумулатора

улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограма (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15..0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна

величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

54 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине 2 бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој а старији бајт на вишој адреси. У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције условног скока. Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0001 за све инструкције безусловног скока. Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 1111 за безадресне инструкције. Битовима 3 до 0 првог бајта инструкције специфицира код операције за безадресне инструкције. Дужина инструкција је један бајт.

Битови 7, 6, 5 и 4 првог бајта инструкције у опсегу вредности 0010 до 1110 специфицирају код операције за адресне инструкције. Дужина инструкција је 1, 2, или 3 бајта и зависи од специфицираног начина адресирања.

Начини адресирања су специфицирани битовима 3 и 2 првог бајта инструкције. Процесор поседује следеће начине адресирања: непосредно адресирање (immed), меморијско директно адресирање (memdir), регистарско индиректно са померајем адресирање (regindrom) и регистарско директно адресирање (regdir). Код непосредног адресирања 16 битни операнд је дат другим и трећим бајтом инструкције, при чему је млађи бајт операнда дат другим а старији бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 3 бајта. Код меморијског директног адресирања 16 битна адреса меморијске локације је дата другим и трећим бајтом инструкције, при чему је млађи бајт адресе дат другим а старији бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 3 бајта. Код регистарског индиректног адресирања са померајем 8 битни померај је целобројна величина са знаком у другом комплементу дата другим бајтом инструкције. Битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 2 бајта. Код регистарског директног адресирања битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 1 бајт.

Стек расте према нижим меморијским локацијама, а регистар SP указује на задњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за

једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	Z = 1
BNEQ	скок на неједнако	Z = 0
BNEG	скок на N = 1	N = 1
BNNG	скок на N = 0	N = 0
BOVF	скок на V = 1	V = 1
BNVF	скок на V = 0	V = 0
BCR	скок на C = 1	C = 1
BNCR	скок на C = 0	C = 0
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	C = 0
BLSSU	скок на мање него (без знака)	C = 1
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $\text{IVTP}_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по

две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

55 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој локацији а млађи бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је старији бајт адресе скока дат другим бајтом инструкције а млађи бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: регистарско директно адресирање (*regdir*), регистарско индиректно са померајем адресирање (*regindpom*), РС релативно са померајем (*pcrel*) и непосредно адресирање (*immed*). Код регистарског директног адресирања се користе неки од регистара опште намене $R[0]$ до $R[63]$ специфициран битовима 5 до 0 другог бајта инструкције. Дужина инструкција је два бајта. Код регистарског индиректног са померајем адресирања трећи и четврти бајт инструкције садрже 16 битни померај који је дат као целобројна величина са знаком, при чему је старији бајт адресе дат трећим бајтом а млађи бајт адресе четвртим бајтом. Један од регистара опште намене $R[0]$ до $R[63]$ који се користи специфициран је битовима 5 до 0 другог бајта инструкције. Дужина инструкција је четири бајта. Код РС релативно са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је три бајта. Код непосредног адресирања трећи бајт инструкције садржи 8 битни податак. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је три бајта.

Стек расте према нижим меморијским локацијама, а регистар *SP* указује на прву слободну меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (*RTS*), инструкција повратка из прекидне рутине (*RTI*), инструкција аритметичког померања садржаја акумулатора удесно за једно место (*ASR*), инструкција логичког померања садржаја акумулатора удесно за једно место (*LSR*), инструкција ротирања садржаја акумулатора удесно за једно место (*ROR*), инструкција ротирања садржаја акумулатора и индикатора *C*

удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора AW у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора AW у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограма (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцијама ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15..0}$ и програмска статусна речи $PSW_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра $IVTP_{15..0}$ (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$

највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

56 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине два бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој локацији а старији бајт на вишој локацији.

У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока, при чему бит 6 првог бајта инструкције има вредност 0 за инструкција условног скока и 1 за инструкције безусловног скока. Битовима 5 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 16 битна целобројна величина са знаком дата другим и трећим бајтом инструкције, при чему је млађи бајт дат другим а старији трећим бајтом инструкције. Дужина инструкције је три бајта. Битовима 5 до 0 првог бајта инструкције се специфицира код операције за инструкције безусловног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата другим и трећим бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је три бајта.

Бит 7 првог бајта инструкције има вредност 1 за безадресне и адресне инструкције, при чему бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције и 1 за адресне инструкције. Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина безадресних инструкција је један бајт. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је два, три или четири бајта и зависи од специфицираног начина адресирања.

За адресне инструкције се битовима 7 и 6 другог бајта инструкције специфицира начина адресирања. Процесор поседује следеће начине адресирања: регистарско директно адресирање (regdir), регистарско индиректно са померајем адресирање (regindpom), РС релативно са померајем (pcrel) и непосредно адресирање (immed). Код регистарског директног адресирања се користе неки од регистара опште намене R[0] до R[63] специфициран битовима 5 до 0 другог бајта инструкције. Дужина инструкција је два бајта. Код регистарског индиректног са померајем адресирања трећи и четврти бајт инструкције садрже 16 битни померај који је дат као целобројна величина са знаком, при чему је млађи бајт адресе дат трећим бајтом а старији бајт адресе четвртим бајтом. Један од регистара опште намене R[0] до R[63] који се користи специфициран је битовима 5 до 0 другог бајта инструкције. Дужина инструкција је четири бајта. Код РС релативно са померајем адресирања трећи бајт инструкције садржи 8 битни померај који је дат као целобројна величина са знаком. Дужина инструкција је три бајта. Код непосредног адресирања трећи и четврти бајт инструкције садрже 16 битни податак, при чему је млађи бајт податка дат трећим бајтом а старији бајт податка четвртим бајтом. Битови 4 до 0 другог бајта инструкције се не користе. Дужина инструкција је четири бајта.

Стек расте према вишим меморијским локацијама, а регистар SP указује на последњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја

акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора А у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора А у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSW регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSW регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15..0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0

уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$ највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у померај, па затим померај сабира са садржајем регистра $IVTP_{15..0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15..0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на zarko@etf.bg.ac.rs.

57 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине 2 бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој а старији бајт на вишој адреси. У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције условног скока. Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0001 за све инструкције безусловног скока. Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 1111 за безадресне инструкције. Битовима 3 до 0 првог бајта инструкције специфицира код операције за безадресне инструкције. Дужина инструкција је један бајт.

Битови 7, 6, 5 и 4 првог бајта инструкције у опсегу вредности 0010 до 1110 специфицирају код операције за адресне инструкције. Дужина инструкција је 1, 2, или 3 бајта и зависи од специфицираног начина адресирања.

Начини адресирања су специфицирани битовима 3 и 2 првог бајта инструкције. Процесор поседује следеће начине адресирања: меморијско директно адресирање (memdir), меморијско индиректно адресирање (memind), регистарско директно адресирање (regdir) и непосредно адресирање (immed). Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе дат другим бајтом а старији бајт адресе трећим бајтом. Код меморијског индиректног адресирања адреса дужине 16 бита заузима две суседне меморијске локације, при чему је млађи бајт адресе налази на нижој а старији бајт адресе на вишој локацији. Битови 1 до 0 другог бајта инструкције се не користе. Дужина инструкција је три бајта. Код регистарског директног адресирања битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 1 бајт. Код непосредног адресирања 16 битни операнд је дат другим и трећим бајтом инструкције, при чему је млађи бајт операнда дат другим а старији бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 3 бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на задњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR),

инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама intr_7 до intr_0 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $\text{PSW}_{15..0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $\text{PSW}_{15..0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $\text{PC}_{15..0}$ и програмска статусна речи $\text{PSW}_{15..0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15..0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_0 , респективно. Прекиди који долазе по линијама intr_7 до intr_0 уређени су по приоритетима при чему линија intr_7 има највиши а линија intr_0 најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије intr_7 до intr_0 највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у

померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

1. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.

58 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.

Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

Процесор је са једноадресним форматом инструкција. Подаци су целобројне величине са знаком и без знака дужине 2 бајта. Подаци и адресе у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој а старији бајт на вишој адреси. У процесору постоје безадресне инструкције, инструкције условног скока, инструкције безусловног скока и адресне инструкције.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције условног скока. Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције условног скока се реализују као релативни скок у односу на текућу вредност програмског бројача РС, а померај је 8 битна целобројна величина са знаком дата другим бајтом инструкције. Дужина инструкције је два бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0001 за све инструкције безусловног скока. Битовима 3 до 0 првог бајта инструкције специфицира код операције за инструкције условног скока. Инструкције безусловног скока се реализују као апсолутни скокови, а адреса скока је дата 2. и 3. бајтом инструкције, при чему је млађи бајт адресе скока дат другим бајтом инструкције а старији бајт адресе скока трећим бајтом инструкције. Дужина инструкција је 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 1111 за безадресне инструкције. Битовима 3 до 0 првог бајта инструкције специфицира код операције за безадресне инструкције. Дужина инструкција је један бајт.

Битови 7, 6, 5 и 4 првог бајта инструкције у опсегу вредности 0010 до 1110 специфицирају код операције за адресне инструкције. Дужина инструкција је 1, 2, или 3 бајта и зависи од специфицираног начина адресирања.

Начини адресирања су специфицирани битовима 3 и 2 првог бајта инструкције. Процесор поседује следеће начине адресирања: меморијско директно адресирање (memdir), меморијско индиректно адресирање (memind), регистарско директно адресирање (regdir) и непосредно адресирање (immed). Код меморијског директног и меморијског индиректног адресирања други и трећи бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе дат другим бајтом а старији бајт адресе трећим бајтом. Код меморијског индиректног адресирања адреса дужине 16 бита заузима две суседне меморијске локације, при чему је млађи бајт адресе налази на нижој а старији бајт адресе на вишој локацији. Битови 1 до 0 другог бајта инструкције се не користе. Дужина инструкција је три бајта. Код регистарског директног адресирања битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R[0] до R[3]. Дужина инструкција је 1 бајт. Код непосредног адресирања 16 битни операнд је дат другим и трећим бајтом инструкције, при чему је млађи бајт операнда дат другим а старији бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 3 бајта.

Стек расте према нижим меморијским локацијама, а регистар SP указује на задњу заузету меморијску локацију.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора С удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSR), инструкција ротирања садржаја акумулатора улево за једно место (ROR),

инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (RORC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржај акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржај акумулатора A у регистар SP (STSP).

Инструкције условног скока су:

Инструкција	Значење	Услов
BEQL	скок на једнако	$Z = 1$
BNEQ	скок на неједнако	$Z = 0$
BNEG	скок на $N = 1$	$N = 1$
BNNG	скок на $N = 0$	$N = 0$
BOVF	скок на $V = 1$	$V = 1$
BNVF	скок на $V = 0$	$V = 0$
BCR	скок на $C = 1$	$C = 1$
BNCR	скок на $C = 0$	$C = 0$
BGRT	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
BGRE	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$
BLEQ	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$
BGREU	скок на веће него или једнако (без знака)	$C = 0$
BLSSU	скок на мање него (без знака)	$C = 1$
BLEQU	скок на мање него или једнако (без знака)	$C \vee Z = 1$

Инструкције безусловног скока су инструкција безусловног скока (JMP) и инструкција скока на потпрограм (JSR).

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT). У инструкцији ST није дозвољено непосредно адресирање, па уколико се јаве ово адресирање инструкција треба да буду без дејства.

Захтеве за прекид може да генерише осам контролера периферија по линијама $intr_7$ до $intr_0$. Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи $PSW_{15...0}$ налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра $PSW_{15...0}$ уписују вредности 1 или 0, респективно.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач $PC_{15...0}$ и програмска статусна речи $PSW_{15...0}$. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP_{15...0} (*Interrupt Vector Table Pointer*), креирана IV табела са 8 улаза, тако да се у улазима 7 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$, респективно. Прекиди који долазе по линијама $intr_7$ до $intr_0$ уређени су по приоритетима при чему линија $intr_7$ има највиши а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу генерише процесор на основу позиције линије $intr_7$ до $intr_0$ највишег нивоа приоритета на којој постоји захтев за прекид. Како је меморијска реч 8 битна величина а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза множењем са два претвара у

померај, па затим померај сабира са садржајем регистра $IVTP_{15...0}$ и на крају добијена вредност користити као адресу са које се чита адреса прекидне рутине и уписује у регистар $PC_{15...0}$.

Реализовати процесор према задатој спецификацији његове архитектуре. Операциона јединица треба да буде реализована са три магистрале а управљачка јединица микропрограмирањем.

НАПОМЕНА:

2. За све нејасноће око спецификације архитектуре процесора и начина реализације процесора јавити се на **zarko@etf.bg.ac.rs**.