

ВЕЖБЕ ИЗ ОРГАНИЗАЦИЈЕ РАЧУНАРА

САДРЖАЈ

Садржај	1
Архитектура и организација процесора	2
Задатак 1.	2
Задатак 2.	6
Задатак 3.	10
Задатак 4.	14
Задатак 5.	19
Задатак 6.	21
Задатак 7.	22
Задатак 8.	24
Задатак 9.	26
Механизам прекида	28
Задатак 10.	28
Задатак 11.	29
Задатак 12.	29
Задатак 13.	31
Задатак 14.	32
Задатак 15.	33
Задатак 16.	35
Задатак 17.	36
Меморија	39
Задатак 18.	39
Задатак 19.	42
Задатак 20.	44
Магистрала	46
Задатак 21.	46
Задатак 22.	46
Задатак 23.	47
Задатак 24.	49
Задатак 25.	49
Улаз/излаз	52
Задатак 26.	52
Задатак 27.	52
Задатак 28.	53
Задатак 29.	54
Задатак 30.	55
Задатак 31.	57
Задатак 32.	58
Задатак 33.	59
Кеш меморија	61
Задатак 34.	61
Виртуелна меморија	63
Задатак 35.	63

АРХИТЕКТУРА И ОРГАНИЗАЦИЈА ПРОЦЕСОРА

Задатак 1.

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 без знака дужине 2 бајта. Подаци у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој а млађи бајт на вишој адреси.

У процесору постоји програмски бројач PC дужине 2 бајта, адресни регистар меморије MAR дужине 2 бајта, прихватни регистар податка меморије MBR дужине 1 бајт, прихватни регистар инструкције IR дужине 4 бајта, акумулатор A дужине 2 бајта, прихватни регистар податка B дужине 2 бајта, регистри опште намене R0 и R31 дужине 2 бајта, програмска статусна реч PSW дужине 1 бајт, указивач на врх стека SP дужине 2 бајта, регистар броја улаза у табелу са адресама прекидних рутина BR дужине 2 бита и указивач на табелу са адресама прекидних рутина IVTP дужине 2 бајта. Инструкције су дужине 1, 2, 3 или 4 бајта.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока. Инструкције скока су инструкција условног скока уколико је резултат нула (JZ), безусловног скока (JMP) и скока на потпрограма (JSR). Адреса скока је дата 2 и 3 бајтом инструкције, при чему је старији бајт адресе скока дат другим а млађи бајт трећим бајтом. Битовима 6 до 0 првог бајта инструкције специфицира се код операције за инструкције скока. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за остале инструкције. Бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције. Безадресне инструкције су инструкција повратка из потпрограма (RTS) и повратка из прекидне рутине (RTI). Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције. Дужина инструкција је 1 бајт.

Бит 6 првог бајта инструкције има вредност 1 за адресне инструкције. Адресне инструкције су инструкција преноса у акумулатор (LOAD), инструкција преноса из акумулатора (STORE), аритметичка инструкција сабирања (ADD), логичка инструкција логички производ (AND) и инструкција аритметичког померања удесно за једно место место код које резултат остаје само у акумулатору (ASR). Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције. Дужина инструкција је 2 или 4 бајта и зависи од специфицираног начина адресирања.

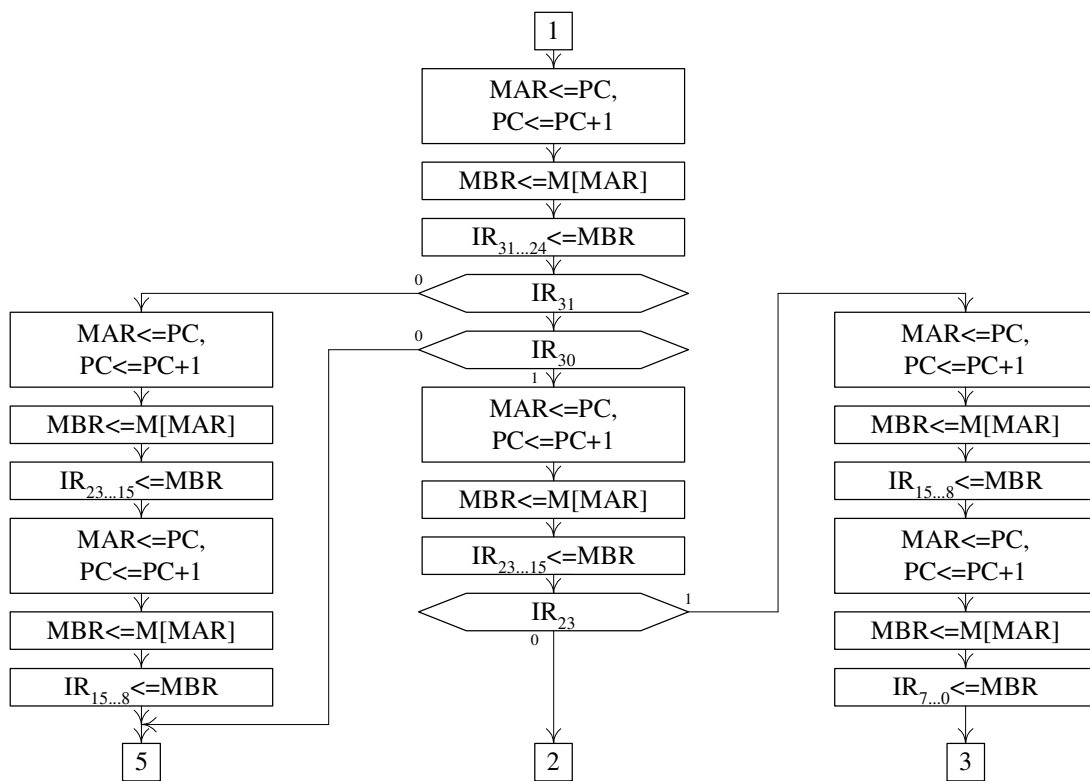
Начини адресирања су специфицирани битовима 7, 6 и 5 другог бајта инструкције и то на следећи начин: 000-регистарско директно адресирање, 001-регистарско индиректно адресирање, 010-регистарско индиректно адресирање са постдекрементирањем, 011-регистарско индиректно адресирање са преинкрементирањем, 100-меморијско директно адресирање, 101-меморијско индиректно адресирање, 110-регистарско индиректно са померајем и 111-непосредно адресирање. Адресирања код којих бит 7 има вредност 0 користе неки од регистара опште намене R0 до R31 специфициран битовима 4 до 0 другог бајта инструкције. Дужина инструкција је 2 бајта. Адресирања код којих бит 7 има вредност 1 имају и трећи и четврти бајт инструкције. Код меморијског директног и меморијског индиректног адресирања трећи и четврти бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе меморијске локације дат трећим а млађи бајт четвртим бајтом. Код меморијског индиректног адресирања адреса дужине 16 бита заузима две суседне меморијске локације, при чему се старији бајт налази на нижој а млађи бајт на вишој адреси. Битови 4 до 0 другог бајта инструкције се не користе. Код регистарског индиректног адресирања са померајем трећи и четврти бајт инструкције садрже 16 битни померај, при чему је старији бајт помераја дат трећим а млађи бајт четвртим бајтом. Један од регистара опште намене R0 до R31 који се користи специфициран је битовима 4 до 0 другог бајта инструкције. Код непосредног адресирања трећи и четврти бајт инструкције садрже 16 битни податак, при чему је старији бајт податка дат трећим а млађи бајт четвртим бајтом. Битови 4 до 0 другог бајта инструкције се не користе.

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

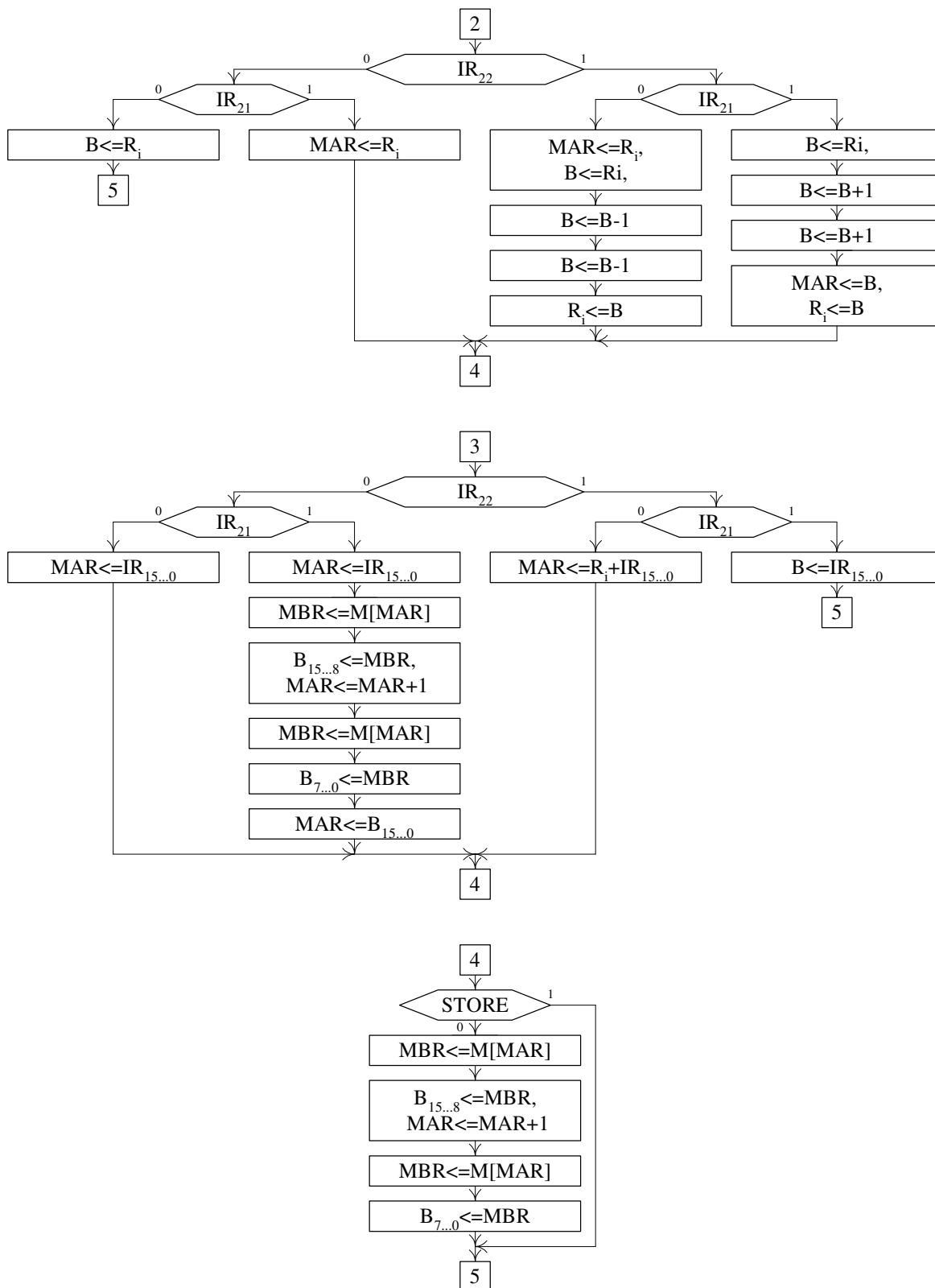
Захтеви за прекид долазе од 4 улазно/излазна уређаја по линијама означеним од 0 до 3. По линији 0 стиже захтев за прекид најнижег, а по линији 3 највишег приоритета. Број линије највишег приоритета по којој је стигао захтев за прекид налази се у бинарном облику у регистру BR дужине 2 разреда. Адресе прекидних рутина 4 улазно/излазна уређаја који по линијама означеним од 0 до 3 шаљу захтеве за прекид налазе се у улазима 0 до 3 табеле са адресама прекидних рутина. Адресе дужине 16 бита заузимају по две суседне меморијске локације, при чему се старији бајт налази на нижој а млађи бајт на вишој адреси. Садржај регистра BR представља број улаза у табелу са адресама прекидних рутина. Почетна адреса табеле са адресама прекидних рутина се налази у регистру IVTP дужине 2 бајта. У оквиру хардверског дела опслуживања захтева за прекид на стек са стављају само регистри PC и PSW.

Нацртати дијаграм тока фаза извршавања инструкције и то: фазе читања инструкције, фазе формирања адресе и читања операнда, фаза извршавања операција LOAD, STORE, ADD, AND, ASR, JZ, JMP, JSR, RTS и RTI и фазе опслуживања захтева за прекид.

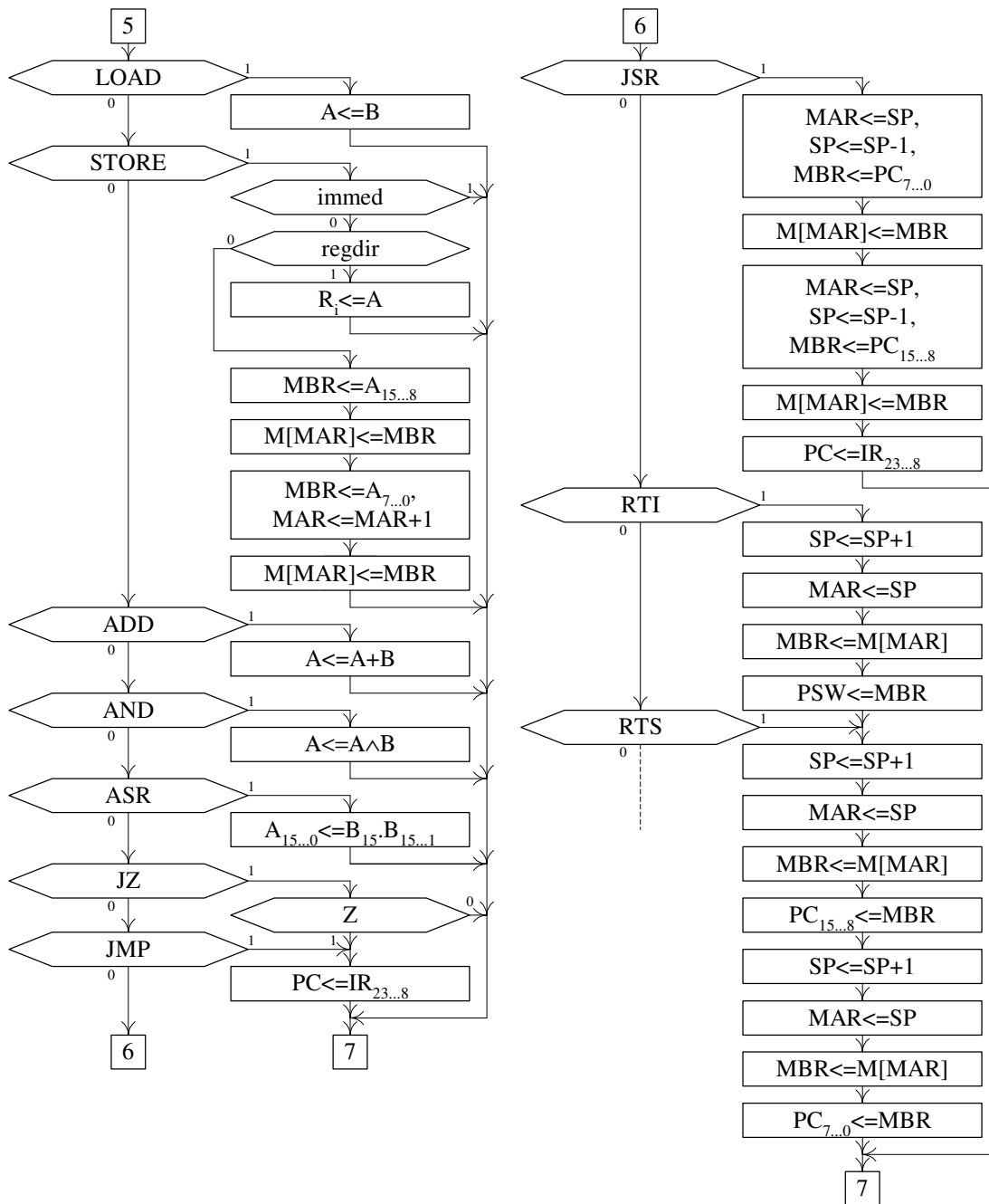
Решење:



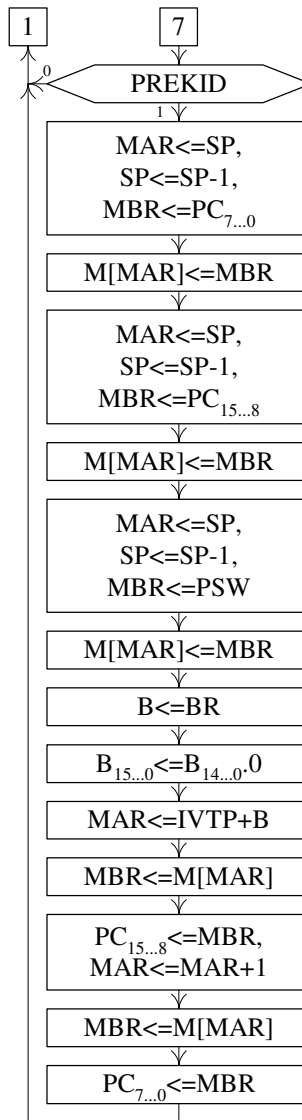
Дијаграм тока – фаза читање инструкције



Дијаграм тока – фаза формирање адресе и читање операнда



Дијаграм тока – фаза извршавање операција



Дијаграм тока – фаза опслуживање прекида

Задатак 2.

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 без знака дужине 2 бајта. Подаци у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој а старији бајт на вишој адреси.

У процесору постоји програмски бројач PC дужине 2 бајта, адресни регистар меморије MAR дужине 2 бајта, прихватни регистар податка меморије MBR дужине 1 бајт, прихватни регистар инструкције IR дужине 3 бајта, акумулатор A дужине 2 бајта, прихватни регистар податка B дужине 2 бајта, регистри опште намене R0 и R3 дужине 2 бајта, програмска статусна реч PSW дужине 1 бајт, указивач на врх стека SP дужине 2 бајта, регистар броја улаза у табелу са адресама прекидних рутина BR дужине 2 бита и указивач на табелу са адресама прекидних рутина IVTP дужине 2 бајта. Инструкције су дужине 1 или 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције скока, док се битовима 3 до 0 првог бајта инструкција специфицира код операције за инструкције скока. Инструкције скока су инструкција условног скока уколико је резултат негативан (JN), безусловног скока (JMP) и скока на потпрограма (JSR). Адреса скока је дата 2 и 3 бајтом инструкције, при чему је млађи бајт адресе скока дат другим а старији бајт трећим бајтом. Дужина инструкција је 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 1111 за безадресне инструкције, док се битовима 3 до 0 првог бајта инструкција специфицира код операције за безадресне инструкције. Безадресне инструкције су инструкција повратка из потпрограма (RTS) и повратка из прекидне рутине (RTI). Дужина инструкција је 1 бајт.

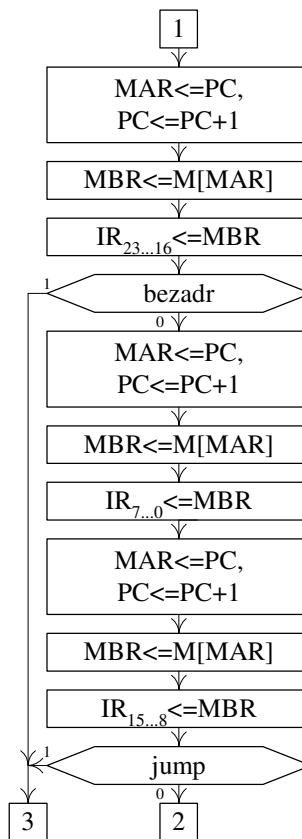
Битови 7, 6, 5 и 4 првог бајта инструкције у опсегу вредности 0001 до 1110 специфицирају код операције за адресне инструкције. Адресне инструкције су инструкције преноса у акумулатор (LOAD), инструкције преноса из акумулатора (STORE), аритметичка инструкција одузимања (SUB), логичка инструкција логичка сума (OR) и инструкција аритметичког померања улево за једно место место код које резултат остаје само у акумулатору (ASL). Начини адресирања су специфицирани битовима 3 и 2 првог бајта инструкције и то на следећи начин: 00-непосредно адресирање, 01-меморијско директно адресирање, 10-регистарско индиректно адресирање са померајем и 11-PC релативно адресирање. Код непосредног адресирања 16 битни операнд је дат другим и трећим бајтом инструкције, при чему је млађи бајт податка дат другим а старији бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се не користе. 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. Код регистарског индиректног адресирања са померајем 16 битни померај је дат другим и трећим бајтом инструкције, при чему је млађи бајт помераја дат другим а старији бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R0 до R3. Код PC релативног адресирања 16 битни померај је дат другим и трећим бајтом инструкције, при чему је млађи бајт помераја 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.

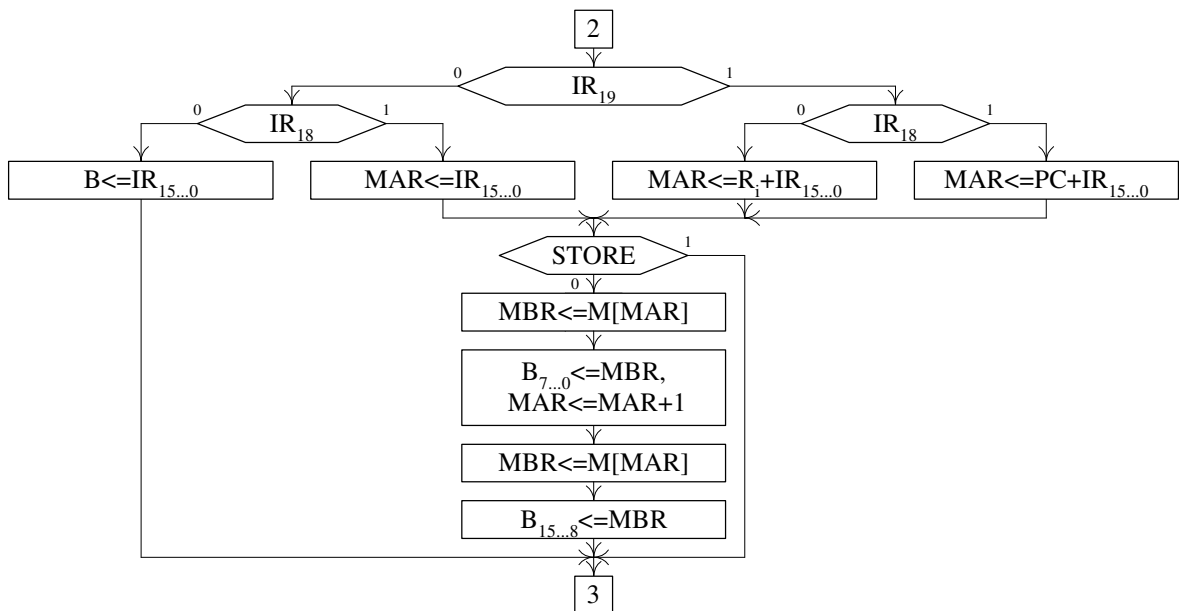
Захтеви за прекид долазе од 4 улазно/излазна уређаја по линијама означеним од 0 до 3. По линији 0 стиже захтев за прекид најнижег, а по линији 3 највишег приоритета. Број линије највишег приоритета по којој је стигао захтев за прекид налази се у бинарном облику у регистру BR дужине 2 разреда. Адресе прекидних рутина 4 улазно/излазна уређаја који по линијама означеним од 0 до 3 шаљу захтеве за прекид налазе се у улазима 0 до 3 табеле са адресама прекидних рутина. Адресе дужине 16 бита заузимају по две суседне меморијске локације, при чему се млађи бајт налази на нижој а старији бајт на вишој адреси. Садржај регистра BR представља број улаза у табелу са адресама прекидних рутина. Почетна адреса табеле са адресама прекидних рутина се налази у регистру IVTP дужине 2 бајта. У оквиру хардверског дела опслуживања захтева за прекид на стек са стављају само регистри PC и PSW.

Нацртати дијаграм тока фаза извршавања инструкције и то: фазе читања инструкције, фазе формирања адресе и читања операнда, фаза извршавања операција LOAD, STORE, SUB, OR, ASL, JN, JMP, JSR, RTS и RTI и фазе опслуживања захтева за прекид.

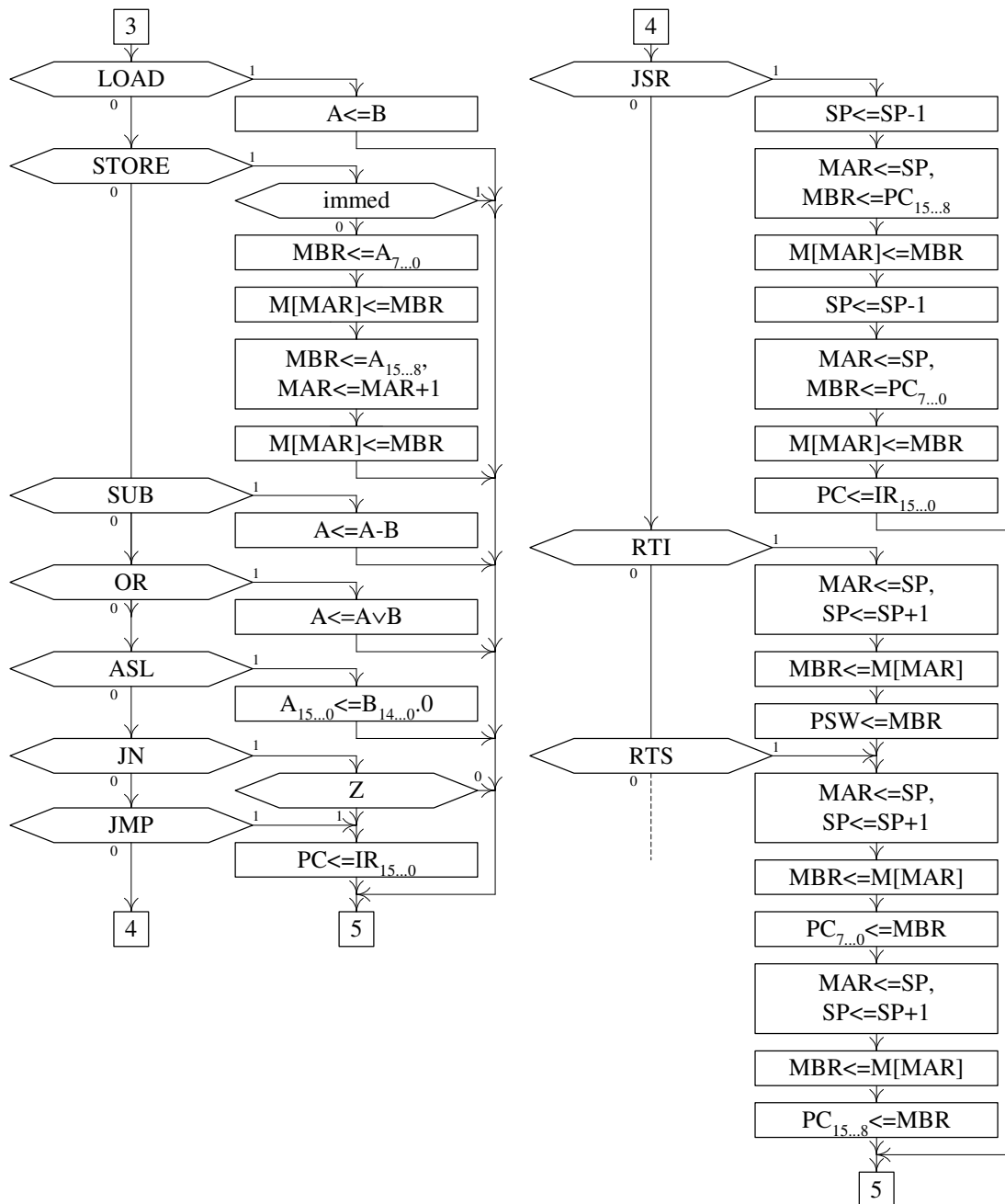
Решење:



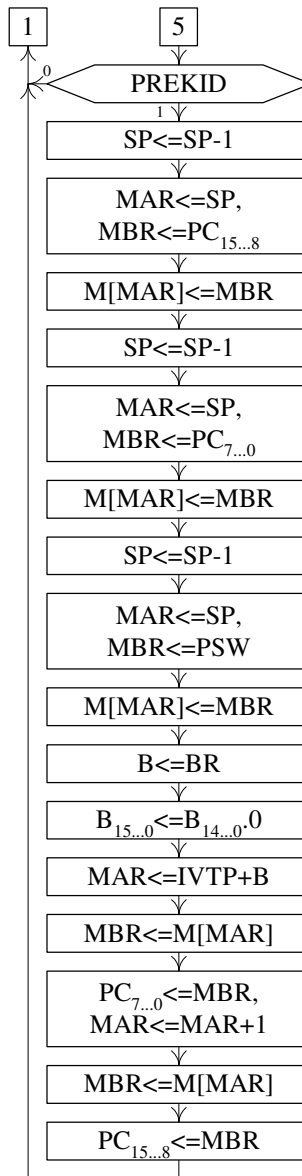
Дијаграм тока – фаза читање инструкције



Дијаграм тока – фаза формирање адресе и читање операнда



Дијаграм тока – фаза извршавање операција



Дијаграм тока – фаза опслуживање прекида

Задатак 3.

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 без знака дужине 1 бајт.

У процесору постоји програмски бројач PC дужине 2 бајта, адресни регистар меморије MAR дужине 2 бајта, прихватни регистар податка меморије MBR дужине 1 бајт, прихватни регистар инструкције IR дужине 3 бајта, акумулатор А дужине 1 бајт, прихватни регистар податка В дужине 1 бајт, регистри опште намене R0 и R1 дужине 2 бајта, програмска статусна реч PSW дужине 1 бајт, указивач на врх стека SP дужине 2 бајта, регистар броја улаза у табелу са адресама прекидних рутина BR дужине 2 бита и указивач на табелу са адресама прекидних рутина IVTP дужине 2 бајта. Инструкције су дужине један, два или три бајта.

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. Безадресне инструкције су инструкција повратка из потпрограма (RTS) и повратка из прекидне рутине (RTI). Дужина инструкција је 1 бајт. Инструкције скока су инструкција условног скока уколико је резултат нула (JZ), безусловног скока (JMP) и скока на потпрограма (JSR). Адреса скока је дата 2 и 3 бајтом инструкције, при чему је старији бајт адресе скока дат другим а млађи бајт трећим бајтом. Дужина инструкција је 3 бајта. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције и инструкције скока.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Адресне инструкције су инструкције преноса у акумулатор (LOAD) и из акумулатора (STORE), аритметичка инструкција сабирања (ADD), логичка инструкција

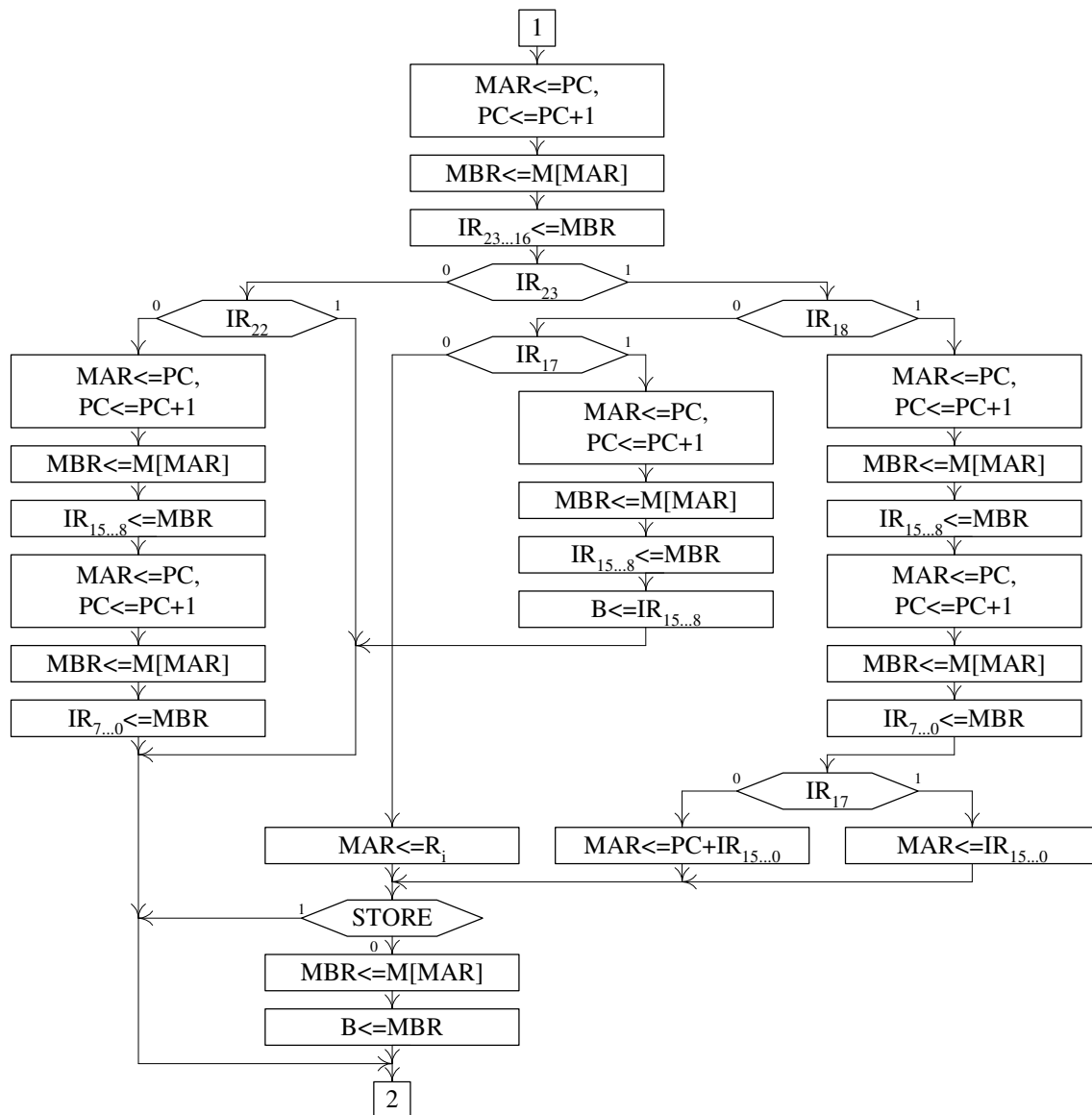
ексклузивно ИЛИ (XOR) и инструкција логичког померања удесно за једно место место код које резултат остаје само у акумулатору (LSR). Битовима 6 до 3 првог бајта инструкција специфицира се код операције, битовима 2 и 1 начин адресирања и битом 0 регистар опште намене уколико се користи у задатом начину адресирања за адресне инструкције. Битовима 2 и 1 начин адресирања се задаје на следећи начин: 00-регистарско индиректно адресирање, 01-непосредно адресирање, 10-PC релативно адресирање са померајем и 11-меморијско директно адресирање. Код регистарског индиректног адресирања регистар опште намене R0 или R1 се специфицира битом 0 првог бајта инструкције. Дужина инструкције је 1 бајт. Код непосредног адресирања други бајт инструкције садрже 8 битни податак, док се бит 0 првог бајта инструкције се не користи. Дужина инструкције је 2 бајта. Код PC релативног адресирања са померајем други и трећи бајт инструкције садрже 16 битни померај, при чему је старији бајт помераја дат другим а млађи бајт трећим бајтом, док се бит 0 првог бајта инструкције се не користи. Дужина инструкције је 3 бајта. Код меморијског директног 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, док се бит 0 првог бајта инструкције се не користи. Дужина инструкције је 3 бајта.

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

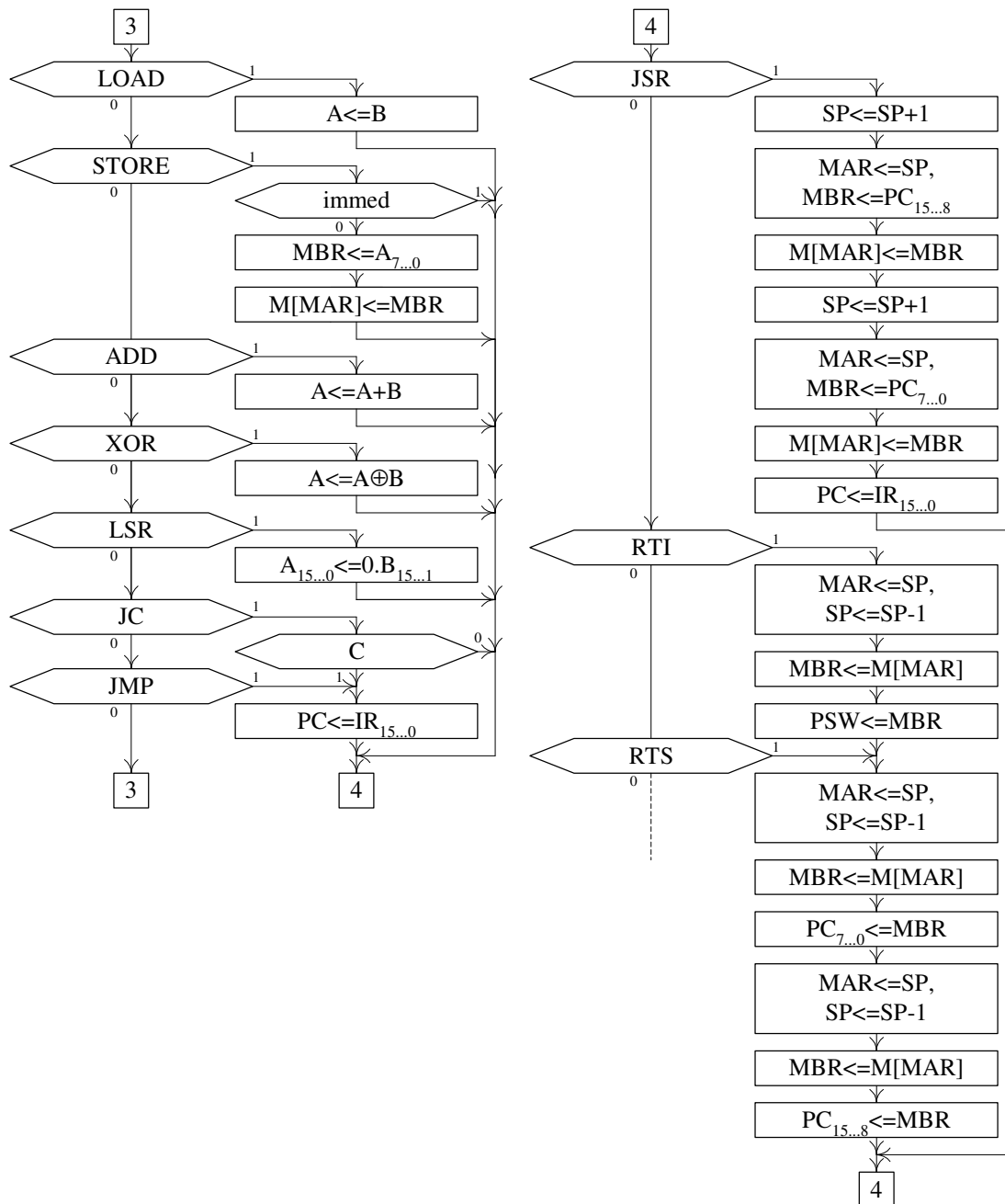
Захтеви за прекид долазе од 4 улазно/излазна уређаја по линијама означеним од 0 до 3. По линији 0 стиже захтев за прекид најнижег, а по линији 3 највишег приоритета. Број линије највишег приоритета по којој је стигао захтев за прекид налази се у бинарном облику у регистру BR дужине 2 разреда. Адресе прекидних рутина 4 улазно/излазна уређаја који по линијама означеним од 0 до 3 шаљу захтеве за прекид налазе се у улазима 0 до 3 табеле са адресама прекидних рутина. Адресе дужине 16 бита заузимају по две суседне меморијске локације, при чему се старији бајт налази на нижој а млађи бајт на вишој адреси. Садржај регистра BR представља број улаза у табелу са адресама прекидних рутина. Почетна адреса табеле са адресама прекидних рутина се налази у регистру IVTP дужине 2 бајта. У оквиру хардверског дела опслуживања захтева за прекид на стек са стављају само регистри PC и PSW.

Нацртати дијаграм тока фаза извршавања инструкције и то: фазе читања инструкције, фазе формирања адресе и читања операнда, фаза извршавања операција LOAD, STORE, ADD, AND, ASR, JZ, JMP, JSR, RTS и RTI и фазе опслуживања захтева за прекид.

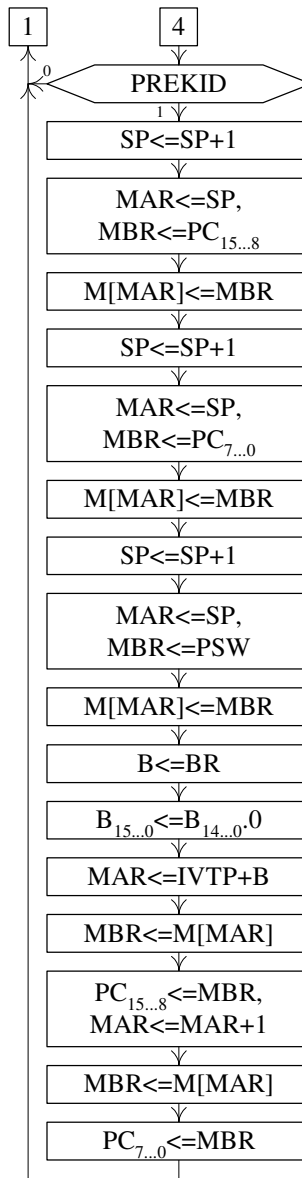
Решење:



Дијаграм тока – фазе читање инструкције и формирање адресе и читање операнда



Дијаграм тока – фаза извршавање операција



Дијаграм тока – фаза опслуживање прекида

Задатак 4.

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 без знака дужине 1 бајт.

У процесору постоји програмски бројач PC дужине 2 бајта, адресни регистар меморије MAR дужине 2 бајта, прихватни регистар податка меморије MBR дужине 1 бајт, прихватни регистар инструкције IR дужине 3 бајта, акумулатор A дужине 1 бајт, прихватни регистар податка B дужине 1 бајт, регистар податка DR дужине 1 бајт, адресни регистар AR дужине 2 бајта, базни регистар BR дужине 2 бајта, индексни регистар XR дужине 2 бајта, програмска статусна реч PSW дужине 1 бајт, указивач на врх стека SP дужине 2 бајта, регистар броја улаза у табелу са адресама прекидних рутина BR дужине 2 бита и указивач на табелу са адресама прекидних рутина IVTP дужине 2 бајта. Инструкције су дужине 1 или 3 бајта.

Битови 7 до 3 првог бајта инструкције су 00000 за све инструкције скока, док се битовима 2 до 0 првог бајта инструкција специфицира код операције за инструкције скока. Инструкције скока су инструкција условног скока уколико резултат није нула (JNZ), безусловног скока (JMP) и скока на потпрограма (JSR). Адреса скока је дата 2 и 3 бајтом инструкције, при чему је млађи бајт адресе скока дат другим а старији бајт трећим бајтом. Дужина инструкција је 3 бајта.

Битови 7 до 3 првог бајта инструкције су 11111 за безадресне инструкције, док се битовима 2 до 0 првог бајта инструкција специфицира код операције за безадресне инструкције. Безадресне инструкције су инструкција повратка из потпрограма (RTS) и повратка из прекидне рутине (RTI). Дужина инструкција је 1 бајт.

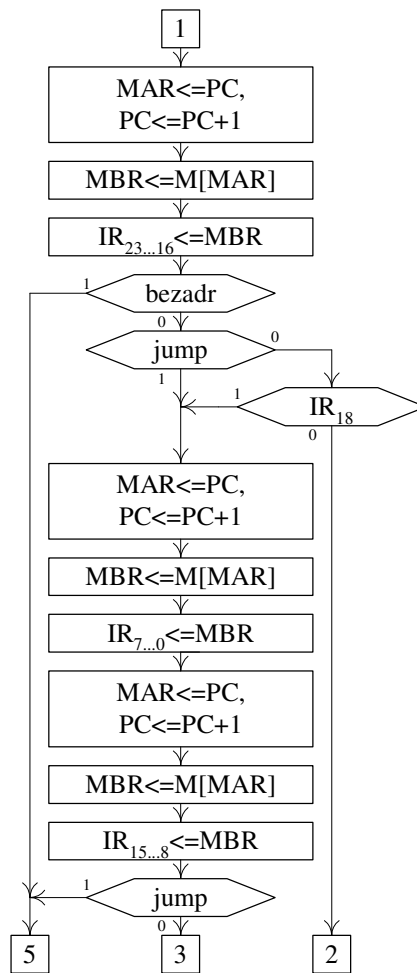
Битови 7 до 3 првог бајта инструкције у опсегу вредности 00001 до 11110 специфицирају код операције за адресне инструкције. Адресне инструкције су инструкција преноса у акумулатор (LOAD), инструкција преноса из акумулатора (STORE), аритметичка инструкција одузимања (SUB), логичка инструкција комплементирања (NOT) и инструкција логичког померања улево за једно место место код које резултат остаје само у акумулатору (LSL). Начини адресирања су специфицирани битовима 2 до 0 првог бајта инструкције и то на следећи начин: 000-регистарско директно адресирање, 001-регистарско индиректно адресирање, 010-регистарско индиректно адресирање са постинкрементирањем, 011-регистарско индиректно адресирање са предекрементирањем, 100-меморијско директно адресирање, 101-базно адресирање са померајем, 110-индексно адресирање са померајем и 111-базно индексно адресирање са померајем. Адресирања код којих бит 2 има вредност 0 имплицитно користе регистар DR(000-регистарско директно адресирање) или AR(001-регистарско индиректно адресирање, 010-регистарско индиректно адресирање са постинкрементирањем, 011-регистарско индиректно адресирање са предекрементирањем). Дужина инструкција је 1 бајт. Адресирања код којих бит 2 има вредност 1 имају и други и трећи бајт инструкције. Код меморијског директног адресирања (100) други и трећи бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе меморијске локације дат другим а старији бајт трећим бајтом. Код базног адресирања са померајем (101) имплицитно се користи регистар BR, а други и трећи бајт инструкције садрже 16 битни померај, при чему је млађи бајт помераја дат другим а старији бајт трећим бајтом. Код индексног адресирања са померајем (110) имплицитно се користи регистар XR, а други и трећи бајт инструкције садрже 16 битни померај, при чему је млађи бајт помераја дат другим а старији бајт трећим бајтом. Код базно индексног адресирања са померајем (111) имплицитно се користи регистри BR и XR, а други и трећи бајт инструкције садрже 16 битни померај, при чему је млађи бајт помераја дат другим а старији бајт трећим бајтом.

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

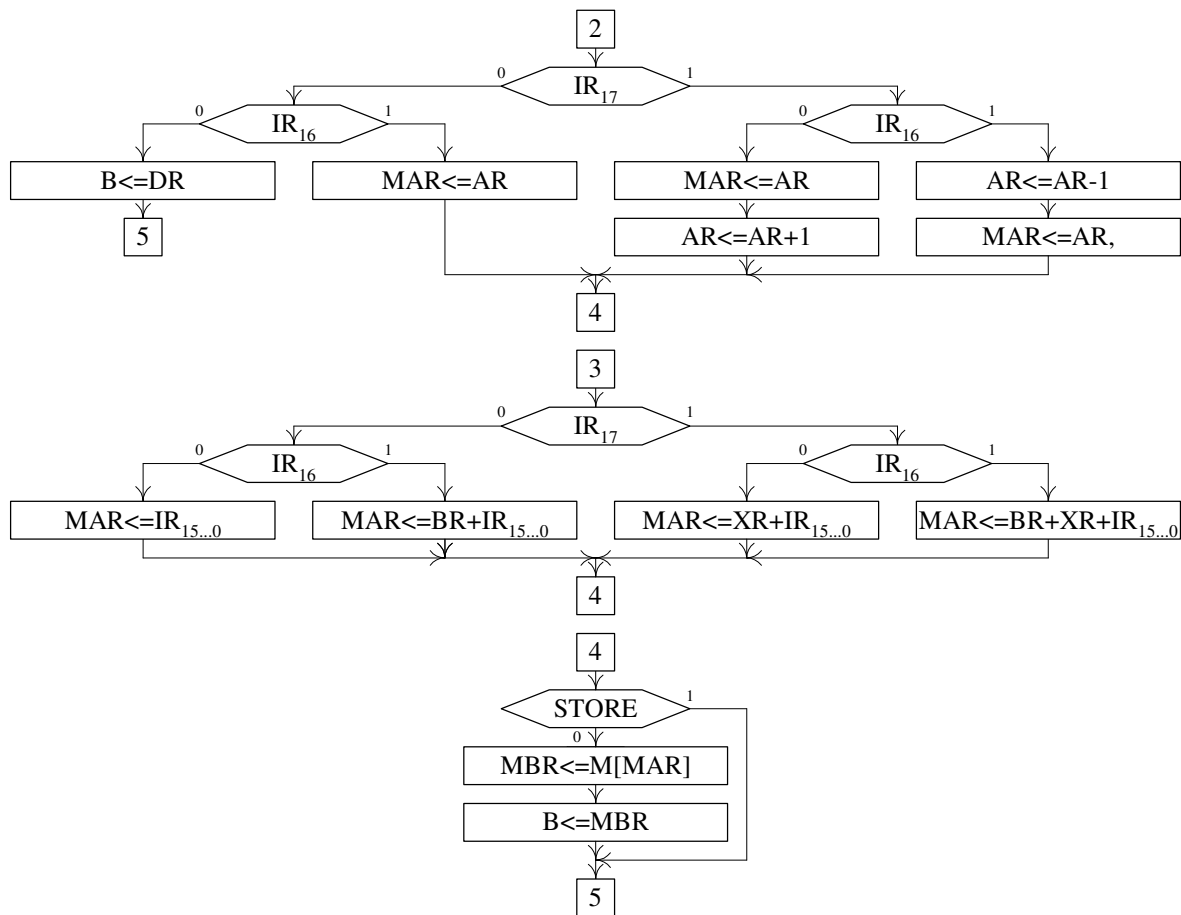
Захтеви за прекид долазе од 4 улазно/излазна уређаја по линијама означеним од 0 до 3. По линији 0 стиже захтев за прекид најнижег, а по линији 3 највишег приоритета. Број линије највишег приоритета по којој је стигао захтев за прекид налази се у бинарном облику у регистру BR дужине 2 разреда. Адресе прекидних рутина 4 улазно/излазна уређаја који по линијама означеним од 0 до 3 шаљу захтеве за прекид налазе се у улазима 0 до 3 табеле са адресама прекидних рутина. Адресе дужине 16 бита заузимају по две суседне меморијске локације, при чему се млађи бајт налази на нижој а старији бајт на вишој адреси. Садржај регистра BR представља број улаза у табелу са адресама прекидних рутина. Почетна адреса табеле са адресама прекидних рутина се налази у регистру IVTP дужине 2 бајта. У оквиру хардверског дела опслуживања захтева за прекид на стек са стављају само регистри PC и PSW.

Нацртати дијаграм тока фаза извршавања инструкције и то: фазе читање инструкције, фазе формирања адресе и читања операнда, фазе извршавања операција LOAD, STORE, SUB, NOT, LSL, JNZ, JMP, JSR, RTS и RTI и фазе опслуживање захтева за прекид.

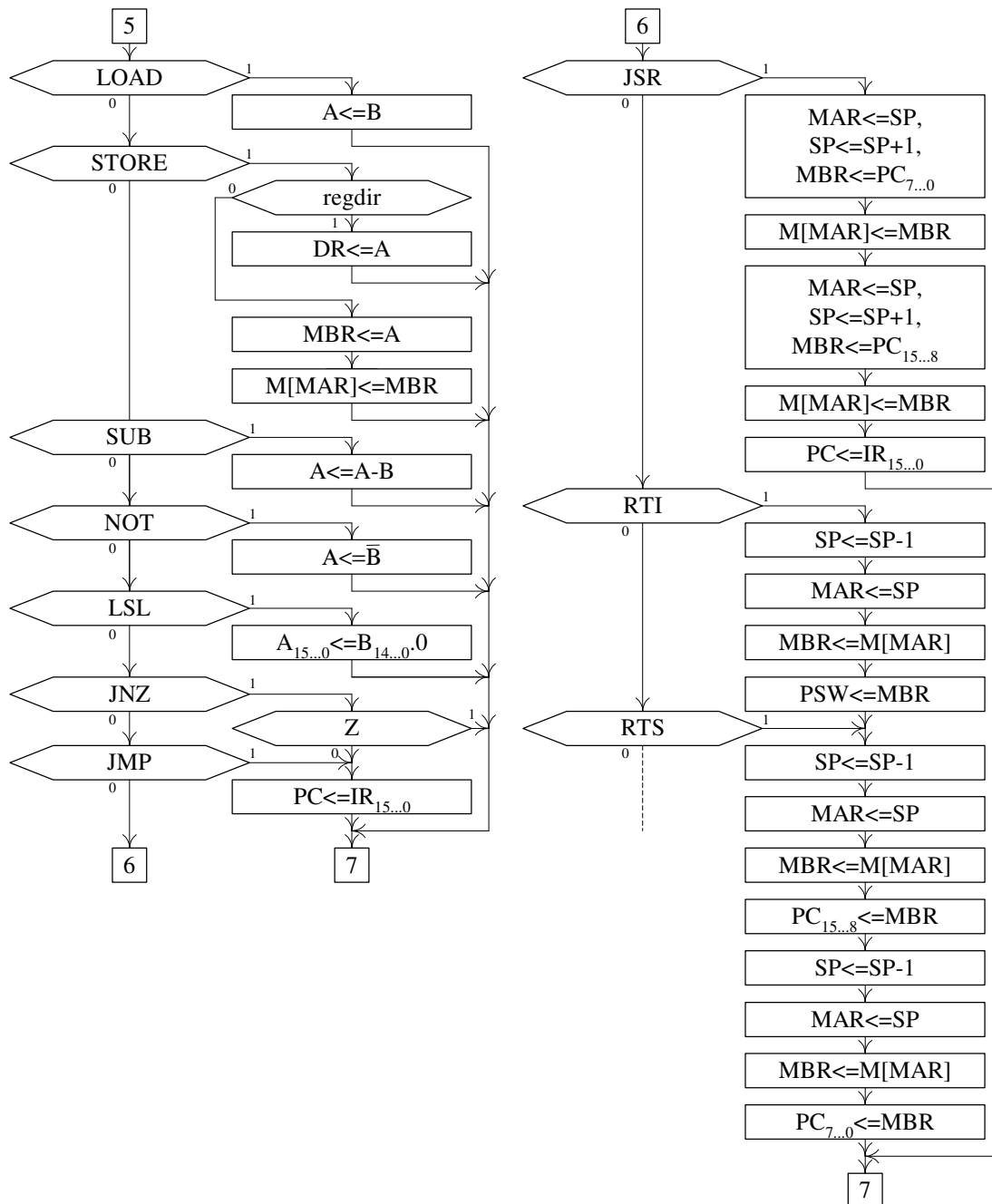
Решење:



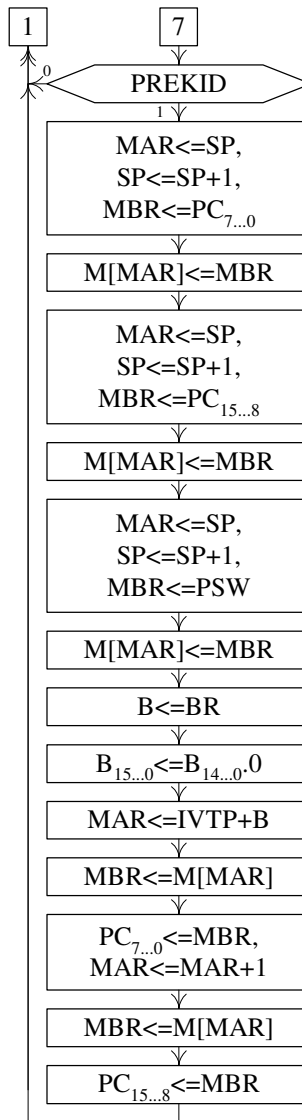
Дијаграм тока – фаза читање инструкције



Дијаграм тока – фаза формирање адресе и читање операнда



Дијаграм тока – фаза извршавање операција



Дијаграм тока – фаза опслуживање прекида

Задатак 5.

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 без знака дужине 2 бајта. Подаци у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој а старији бајт на вишој адреси.

У процесору постоји програмски бројач PC дужине 2 бајта, адресни регистар меморије MAR дужине 2 бајта, прихватни регистар податка меморије MBR дужине 1 бајт, прихватни регистар инструкције IR дужине 3 бајта, акумулатор A дужине 2 бајта, прихватни регистар податка B дужине 2 бајта, регистри опште намене R0 до R3 дужине 2 бајта, програмска статусна реч PSW дужине 1 бајт. Инструкције су дужине 1 или 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције скока, док се битовима 3 до 0 првог бајта инструкција специфицира код операције за инструкције скока и то на следећи начин: 0001-инструкција условног скока уколико постоји пренос (JC), 0010-инструкција безусловног скока (JMP) и 0100-инструкција скока на потпрограма (JSR). Адреса скока је дата 2 и 3 бајтом инструкције, при чему је млађи бајт адресе скока дат другим а старији бајт трећим бајтом. Дужина инструкција је 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 1111 за безадресне инструкције, док се битовима 3 до 0 првог бајта инструкција специфицира код операције за безадресне инструкције и то на следећи начин: 0010-инструкција повратка из потпрограма (RTS) и 0101-инструкција повратка из прекидне рутине (RTI). Дужина инструкција је 1 бајт.

Битови 7, 6, 5 и 4 првог бајта инструкције у опсегу вредности 0001 до 1110 специфицирају код операције за адресне инструкције и то на следећи начин: 0001-инструкција преноса у акумулатор (LOAD), 0011-инструкције преноса из акумулатора (STORE), 1001-аритметичка инструкција одузимања (SUB), 1100-логичка инструкција логичка сума (OR) и 0110-инструкција аритметичког померања удесно за једно место код које резултат остаје само у акумулатору (ASR). Начини адресирања су специфицирани битовима 3 и 2 првог бајта инструкције и то на следећи начин: 00-непосредно адресирање, 01-меморијско директно адресирање, 10-регистарско индиректно адресирање са померајем и 11-PC релативно адресирање. Код непосредног адресирања 16 битни операнд је дат другим и трећим бајтом инструкције, при чему је млађи бајт податка дат другим а старији бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се не користе. 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. Код регистарског индиректног адресирања са померајем 16 битни померај је дат другим и трећим бајтом инструкције, при чему је млађи бајт помераја дат другим а старији бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R0 до R3. Код PC релативног адресирања 16 битни померај је дат другим и трећим бајтом инструкције, при чему је млађи бајт помераја дат другим а старији бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се не користе. Дужина инструкција је 3 бајта.

На индикаторе у програмској статусној речи утичу следеће инструкције: LOAD на N и Z, SUB на N, Z, V и C, OR на N и Z и ASR на N, Z и C.

Садржај дела меморије је:

ЛОКАЦИЈА:	10h	11h	12h	13h	14h	15h	16h	17h	18h	19h
САДРЖАЈ:	01h	00h	41h	00h	E3h	13h	00h	21h	4Ch	B8h
ЛОКАЦИЈА:	1000h	1001h	1002h	1003h	1004h	1005h	1006h	1007h	1008h	1009h
САДРЖАЈ:	14h	10h	00h	90h	0Fh	00h	01h	0Ch	10h	34h
ЛОКАЦИЈА:	100Ah	100Bh	100Ch	100Dh	100Eh	100Fh	1010h	1011h	1012h	1013h
САДРЖАЈ:	10h	00h	68h	05h	00h	34h	10h	00h	10h	00h

Ако је PC=1000h, ACC=7Ch, R0=0Eh, N=0, Z=0, V=0 и C=0. Извршити 4 сукцесивних наредби и за сваку наредбу одредити следеће: у фази читања инструкције меморијске адресе са којих је прочитана инструкција, садржај регистра IR, инструкцију која је прочитана и нови садржај регистра PC, у фази одређивања адресе операнда и читања операнда меморијске адресе са којих је прочитана адреса операнда, меморијске адресе са којих је прочитан операнд, вредност операнда и нови садржај регистара опште намене који су у овој фази промењени, у фази извршавања меморијске адресе којима се у овој фази приступа, садржај акумулатора A и индикатора N, Z, V и C након извршавања инструкције и нови садржај регистара и меморијских локација који су у овој фази промењени. Резултате представити у форми једне или више табела које садрже наведене колоне.

Решење:

Фаза читања инструкције:

Рб инст.	Адресе са којих је прочитана инструкција	IR _{23..16}	IR _{15..8}	IR _{7..0}	Прочитана инструкција	Садржај промењених регистара
1.	1000, 1001, 1002	14	10	00	LOAD 10h	PC=1003
2.	1003, 1004, 1005	90	0F	00	SUB #Fh	PC=1006
3.	1006, 1007, 1008	01	0C	10	JC 100Ch	PC=1009
4.	100C, 100D, 100E	68	05	00	ASR (R0)5h	PC=100F

Фаза одређивања адресе операнда и читање операнда:

Рб инст.	Адресе са којих је прочитана адреса операнда	Адресе са којих је прочитан операнд	Операнд	Садржај промењених регистара
1.	-	0010, 0011	0001	-
2.	-	-	000F	-
3.	-	-	-	-
4.	-	0013, 0014	E300	-

Фаза извршавања:

Рб инст.	Адресе којима се приступа у фази извршавања	A	N	Z	V	C	Садржај промењених регистра и меморијских локација
1.	-	0001	0	0	0	0	-
2.	-	FFF2	1	0	0	1	-
3.	-	FFF2	1	0	0	1	PC=100C
4.	-	F180	1	0	0	0	-

Задатак 6.

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 дужине 1 бајт представљени у другом комплементу.

У процесору постоји програмски бројач PC дужине 2 бајта, адресни регистар меморије MAR дужине 2 бајта, прихватни регистар податка меморије MBR дужине 1 бајт, прихватни регистар инструкције IR дужине 3 бајта, акумулатор A дужине 1 бајт, прихватни регистар податка B дужине 1 бајт, регистар податка DR дужине 1 бајт, адресни регистар AR дужине 2 бајта, базни регистар BR дужине 2 бајта, индексни регистар XR дужине 2 бајта, програмска статусна реч PSW дужине 1 бајт, указивач на врх стека SP дужине 2 бајта. Инструкције су дужине 1 или 3 бајта.

Битови 7 до 3 првог бајта инструкције су 00000 за све инструкције скока, док се битовима 2 до 0 првог бајта инструкција специфицира код операције за инструкције скока и то на следећи начин: 001-инструкција условног скока уколико резултат није нула (JNZ), 011-безусловног скока (JMP) и 100-скока на потпрограма (JSR). Адреса скока је дата 2 и 3 бајтом инструкције, при чему је млађи бајт адресе скока дат другим а старији бајт трећим бајтом. Дужина инструкција је 3 бајта.

Битови 7 до 3 првог бајта инструкције су 11111 за безадресне инструкције, док се битовима 2 до 0 првог бајта инструкција специфицира код операције за безадресне инструкције и то на следећи начин: 010-инструкција повратка из потпрограма (RTS) и 110-повратка из прекидне рутине (RTI). Дужина инструкција је 1 бајт.

Битови 7 до 3 првог бајта инструкције у опсегу вредности 00001 до 11110 специфицирају код операције за адресне инструкције и то на следећи начин: 00001-инструкција преноса у акумулатор (LOAD), 00010-инструкција преноса из акумулатора (STORE), 00110-аритметичка инструкција одузимања (SUB), 00111-логичка инструкција комплементирања (NOT) и 01010-инструкција логичког померања улево за једно место место код које резултат остаје само у акумулатору (LSL). Начини адресирања су специфицирани битовима 2 до 0 првог бајта инструкције и то на следећи начин: 000-регистарско директно адресирање, 001-регистарско индиректно адресирање, 010-регистарско индиректно адресирање са постинкрементирањем, 011-регистарско индиректно адресирање са предекрементирањем, 100-меморијско директно адресирање, 101-базно адресирање са померајем, 110-индексно адресирање са померајем и 111-базно индексно адресирање са померајем. Адресирања код којих бит 2 има вредност 0 имплицитно користе регистар DR(000-регистарско директно адресирање) или AR(001-регистарско индиректно адресирање, 010-регистарско индиректно адресирање са постинкрементирањем, 011-регистарско индиректно адресирање са предекрементирањем). Дужина инструкција је 1 бајт. Адресирања код којих бит 2 има вредност 1 имају и други и трећи бајт инструкције. Код меморијског директног адресирања (100) други и трећи бајт инструкције садрже адресу меморијске локације, при чему је млађи бајт адресе меморијске локације дат другим а старији бајт трећим бајтом. Код базног адресирања са померајем (101) имплицитно се користи регистар BR, а други и трећи бајт инструкције садрже 16 битни померај, при чему је млађи бајт помераја дат другим а старији бајт трећим бајтом. Код индексног адресирања са померајем (110) имплицитно се користи регистар XR, а други и трећи бајт инструкције садрже 16 битни померај, при чему је млађи бајт помераја дат другим а старији бајт трећим бајтом. Код базно индексног адресирања са померајем (111) имплицитно се користи регистри BR и XR, а други и трећи бајт инструкције садрже 16 битни померај, при чему је млађи бајт помераја дат другим а старији бајт трећим бајтом.

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

На индикаторе у програмској статусној речи утичу следеће инструкције: LOAD на N и Z, SUB на N, Z, V и C, NOT на N и Z и LSL на N, Z и C.

Садржај дела меморије је:

ЛОКАЦИЈА:	10h	11h	12h	13h	14h	15h	16h	17h	18h	19h
САДРЖАЈ:	00h	01h	05h	00h	E3h	FAh	00h	21h	4Ch	FFh
ЛОКАЦИЈА:	1000h	1001h	1002h	1003h	1004h	1005h	1006h	1007h	1008h	1009h
САДРЖАЈ:	04h	00h	11h	34h	10h	00h	01h	00h	10h	10h
ЛОКАЦИЈА:	1100h	1101h	1102h	1103h	1104h					
САДРЖАЈ:	3Ah	FAh	68h	05h	00h					

Ако је $PC=1000h$, $SP=0100h$, $ACC=F3h$, $AR=15h$, $N=1$, $Z=0$, $V=0$ и $C=0$. Извршити 5 сукцесивних наредби и за сваку наредбу одредити следеће: у фази читања инструкције меморијске адресе са којих је прочитана инструкција, садржај регистра IR, инструкцију која је прочитана и нови садржај регистра PC, у фази одређивања адресе операнда и читања операнда меморијске адресе са којих је прочитана адреса операнда, меморијске адресе са којих је прочитан операнд, вредност операнда и нови садржај регистара опште намене који су у овој фази промењени, у фази извршавања меморијске адресе којима се у овој фази приступа, садржај акумулатора A и индикатора N, Z, V и C након извршавања инструкције и нови садржај регистара и меморијских локација који су у овој фази промењени. Резултате представити у форми једне или више табела које садрже наведене колоне.

Решење:

Фаза читања инструкције:

Рб инст.	Адресе са којих је прочитана инструкција	IR _{23..16}	IR _{15..8}	IR _{7..0}	Прочитана инструкција	Садржај промењених регистара
1.	1000, 1001, 1002	04	00	11	JSR 1100h	PC=1003
2.	1100	3A	-	-	NOT (AR)+	PC=1101
3.	1101	FA	-	-	RTS	PC=1102
4.	1003, 1004, 1005	34	10	00	SUB 10h	PC=1006
5.	1006, 1007, 1008	01	00	10	JNZ 1000h	PC=1009

Фаза одређивања адресе операнда и читање операнда:

Рб инст.	Адресе са којих је прочитана адреса операнда	Адресе са којих је прочитан операнд	Операнд	Садржај промењених регистара
1.	-	-	-	-
2.	-	0015	FA	AR = 16h
3.	-	-	-	-
4.	-	0010	00	-
5.	-	-	-	-

Фаза извршавања:

Рб инст.	Адресе којима се приступа у фази извршавања	A	N	Z	V	C	Садржај промењених регистара и меморијских локација
1.	0100, 0101	F3	1	0	0	0	MEM[0100]=03; MEM[0101]=10, SP=102, PC=1100
2.	-	05	0	0	0	0	-
3.	0101, 0100	05	0	0	0	0	SP=100, PC=1003
4.	-	05	0	0	0	0	-
5.	-	05	0	0	0	0	PC=1000

Задатак 7.

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 дужине 2 бајта представљени у другом комплементу. Подаци у меморији заузимају две суседне меморијске локације, при чему се старији бајт налази на нижој а млађи бајт на вишој адреси.

У процесору постоји програмски бројач РС дужине 2 бајта, адресни регистар меморије MAR дужине 2 бајта, прихватни регистар податка меморије MBR дужине 1 бајт, прихватни регистар инструкције IR дужине 4 бајта, акумулатор А дужине 2 бајта, прихватни регистар податка В дужине 2 бајта, регистри опште намене R0 и R31 дужине 2 бајта, програмска статусна реч PSW дужине 1 бајт, указивач на врх стека SP дужине 2 бајта, регистар броја улаза у табелу са адресама прекидних рутина BR дужине 2 бита и указивач на табелу са адресама прекидних рутина IVTP дужине 2 бајта. Инструкције су дужине 1, 2, 3 или 4 бајта.

Бит 7 првог бајта инструкције има вредност 0 за инструкције скока. Инструкције скока су инструкција условног скока уколико је резултат нула (JZ), безусловног скока (JMP) и скока на потпрограма (JSR). Адреса скока је дата 2 и 3 бајтом инструкције, при чему је старији бајт адресе скока дат другим а млађи бајт трећим бајтом. Битовима 6 до 0 првог бајта инструкције специфицира се код операције за инструкције скока и то са 0000001 за JZ, са 0011100 за JMP и са 1100110 за JSR. Дужина инструкција је 3 бајта.

Бит 7 првог бајта инструкције има вредност 1 за остале инструкције. Бит 6 првог бајта инструкције има вредност 0 за безадресне инструкције. Безадресне инструкције су инструкција повратка из потпрограма (RTS) и повратка из прекидне рутине (RTI). Битовима 5 до 0 првог бајта инструкције специфицира се код операције за безадресне инструкције и то са 000011 за RTS и са 111010 за RTI. Дужина инструкција је 1 бајт.

Бит 6 првог бајта инструкције има вредност 1 за адресне инструкције. Адресне инструкције су инструкција преноса у акумулатор (LOAD), инструкција преноса из акумулатора (STORE), аритметичка инструкција сабирања (ADD), логичка инструкција логички производ (AND) и инструкција аритметичког померања удесно за једно место место код које резултат остаје само у акумулатору (ASR). Битовима 5 до 0 првог бајта инструкција специфицира се код операције за адресне инструкције и то са 000000 за LOAD, са 111111 за STORE, са 001001 за ADD, са 111110 за AND и са 100101 за ASR. Дужина инструкција је 2 или 4 бајта и зависи од специфицираног начина адресирања.

Начини адресирања су специфицирани битовима 7, 6 и 5 другог бајта инструкције и то на следећи начин: 000-регистарско директно адресирање, 001-регистарско индиректно адресирање, 010-регистарско индиректно адресирање са постдекрементирањем, 011-регистарско индиректно адресирање са преинкрементирањем, 100-меморијско директно адресирање, 101-меморијско индиректно адресирање, 110-регистарско индиректно са померајем и 111-непосредно адресирање. Адресирања код којих бит 7 има вредност 0 користе неки од регистара опште намене R0 до R31 специфициран битовима 4 до 0 другог бајта инструкције. Дужина инструкција је 2 бајта. Адресирања код којих бит 7 има вредност 1 имају и трећи и четврти бајт инструкције. Код меморијског директног и меморијског индиректног адресирања трећи и четврти бајт инструкције садрже адресу меморијске локације, при чему је старији бајт адресе меморијске локације дат трећим а млађи бајт четвртим бајтом. Код меморијског индиректног адресирања адреса дужине 16 бита заузима две суседне меморијске локације, при чему се старији бајт налази на нижој а млађи бајт на вишој адреси. Битови 4 до 0 другог бајта инструкције се не користе. Код регистарског индиректног адресирања са померајем трећи и четврти бајт инструкције садрже 16 битни померај, при чему је старији бајт помераја дат трећим а млађи бајт четвртим бајтом. Један од регистара опште намене R0 до R31 који се користи специфициран је битовима 4 до 0 другог бајта инструкције. Код непосредног адресирања трећи и четврти бајт инструкције садрже 16 битни податак, при чему је старији бајт податка дат трећим а млађи бајт четвртим бајтом. Битови 4 до 0 другог бајта инструкције се не користе.

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

На индикаторе у програмској статусној речи утичу следеће инструкције: LOAD на N и Z, ADD на N, Z, V и C, AND на N и Z и ASR на N, Z и C.

Садржај дела меморије је:

ЛОКАЦИЈА:	10h	11h	12h	13h	14h	15h	16h	17h	18h	19h
САДРЖАЈ:	00h	14h	00h	17h	FFh	F0h	00h	01h	F0h	ABh
ЛОКАЦИЈА:	1000h	1001h	1002h	1003h	1004h	1005h	1006h	1007h	1008h	1009h
САДРЖАЈ:	C0h	A0h	00h	10h	C9h	61h	01h	10h	0Dh	FFh
ЛОКАЦИЈА:	100Ah	100Bh	100Ch	100Dh	100Eh					
САДРЖАЈ:	80h	00h	12h	FEh	02h					

а) Ако је PC=1000h, SP=1102h, ACC=10h, R1=14h, R2=07h, N=0, Z=0, V=0 и C=0. Извршити 5 сукцесивних наредби и за сваку наредбу одредити следеће: у фази читања инструкције меморијске адресе са којих је прочитана инструкција, садржај регистра IR, инструкцију која је прочитана и нови садржај регистра PC, у фази одређивања адресе операнда и читања операнда меморијске адресе са којих је прочитана адреса операнда, меморијске адресе са којих је прочитан операнд, вредност операнда и нови садржај регистра опште намене који су у овој фази промењени, у фази извршавања меморијске адресе којима се у овој фази приступа, садржај акумулатора А и индикатора N, Z, V и C након извршавања инструкције и нови садржај регистра и меморијских локација који су у овој фази промењени. Резултате представити у форми једне или више табела које садрже наведене колоне.

Решење:

Фаза читања инструкције:

Рб инст.	Адресе са којих је прочитана инструкција	IR _{31..24}	IR _{23..16}	IR _{15..8}	IR _{7..0}	Прочитана инструкција	Садржај промењених регистара
1.	1000, 1001, 1002, 1003	C0	A0	00	10	LOAD (0010h)	PC=1004
2.	1004, 1005	C9	61	-	-	ADD +(R1)	PC=1006
3.	1006 1007, 1008	01	10	0D	-	JZ 100Dh	PC=1009
4.	1009, 100A, 100B, 100C	FF	80	00	12	STORE 12h	PC=100D
5.	100D, 100E	FE	02	-	-	AND R2	PC=100F

Фаза одређивања адресе операнда и читање операнда:

Рб инст.	Адресе са којих је прочитана адреса операнда	Адресе са којих је прочитан операнд	Операнд	Садржај промењених регистара
1.	0010, 0011	0014, 0015	FFF0	-
2.	-	0016, 0017	0001	(++R1) => R1=16
3.	-	-	-	-
4.	-	-	-	-
5.	-	-	0007	-

Фаза извршавања:

Рб инст.	Адресе којима се приступа у фази извршавања	A	N	Z	V	C	Садржај промењених регистара и меморијских локација
1.	-	FFF0	1	0	?	?	-
2.	-	FFF1	1	0	0	0	-
3.	-	FFF1	1	0	0	0	-
4.	0012, 0013	FFF1	1	0	0	0	-
5.	-	0001	0	0	0	0	-

Задатак 8.

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 без знака дужине 1 бајт.

У процесору постоји програмски бројач PC дужине 2 бајта, адресни регистар меморије MAR дужине 2 бајта, прихватни регистар податка меморије MBR дужине 1 бајт, прихватни регистар инструкције IR дужине 3 бајта, акумулатор A дужине 1 бајт, прихватни регистар податка B дужине 1 бајт, регистри опште намене R0 и R1 дужине 2 бајта, програмска статусна реч PSW дужине 1 бајт, указивач на врх стека SP дужине 2 бајта, регистар броја улаза у табелу са адресама прекидних рутина BR дужине 2 бита и указивач на табелу са адресама прекидних рутина IVTP дужине 2 бајта. Инструкције су дужине један, два или три бајта.

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. Безадресне инструкције су инструкција повратка из потпрограма (RTS) и повратка из прекидне рутине (RTI). Дужина инструкција је 1 бајт. Инструкције скока су инструкција условног скока уколико је резултат нула (JZ), безусловног скока (JMP) и скока на потпрограма (JSR). Адреса скока је дата 2 и 3 бајтом инструкције, при чему је старији бајт адресе скока дат другим а млађи бајт трећим бајтом. Дужина инструкција је 3 бајта. Битовима 5 до 0 првог бајта инструкција специфицира се код операције за безадресне инструкције и инструкције скока и то са 001100 за RTS, са 110011 за RTI, са 000101 за JZ, са 101010 за JMP и са 010101 за JSR.

Бит 7 првог бајта инструкције има вредност 1 за адресне инструкције. Адресне инструкције су инструкције преноса у акумулатор (LOAD) и из акумулатора (STORE), аритметичка инструкција сабирања (ADD), логичка инструкција ексклузивно ИЛИ (XOR) и инструкција логичког померања удесно за једно место место код које резултат остаје само у акумулатору (LSR). Битовима 6 до 3 првог бајта инструкција специфицира се код операције, битовима 2 и 1 начин адресирања и битом 0 регистар опште намене уколико се користи у задатом начину адресирања за адресне инструкције. Битовима 6 до 3 првог бајта инструкција код инструкције се задаје на следећи начин: 0001 за LOAD, 0010 за STORE, 0100 за ADD, 0111 за XOR и 1000 за LSR. Битовима 2 и 1 начин адресирања се задаје на следећи

начин: 00-регистарско индиректно адресирање, 01-непосредно адресирање, 10-PC релативно адресирање са померајем и 11-меморијско директно адресирање. Код регистарског индиректног адресирања регистар опште намене R0 или R1 се специфицира битом 0 првог бајта инструкције. Дужина инструкције је 1 бајт. Код непосредног адресирања други бајт инструкције садрже 8 битни податак, док се бит 0 првог бајта инструкције се не користи. Дужина инструкције је 2 бајта. Код PC релативног адресирања са померајем други и трећи бајт инструкције садрже 16 битни померај, при чему је старији бајт помераја дат другим а млађи бајт трећим бајтом, док се бит 0 првог бајта инструкције се не користи. Дужина инструкције је 3 бајта. Код меморијског директног 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, док се бит 0 првог бајта инструкције се не користи. Дужина инструкције је 3 бајта.

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

На индикаторе у програмској статусној речи утичу следеће инструкције: LOAD на N и Z, ADD на N, Z, V и C, XOR на N и Z и LSR на N, Z и C.

Садржај дела меморије је:

ЛОКАЦИЈА:	10h	11h	12h	13h	14h	15h	16h	17h	18h	19h
САДРЖАЈ:	80h	14h	FFh	12h	13h	7Fh	54h	67h	10h	ABh
ЛОКАЦИЈА:	1000h	1001h	1002h	1003h	1004h	1005h	1006h	1007h	1008h	1009h
САДРЖАЈ:	8Ch	F0h	0Fh	B8h	6Ah	10h	0Ah	96h	00h	14h
ЛОКАЦИЈА:	100Ah	100Bh	100Ch	100Dh	100Eh					
САДРЖАЈ:	C2h	80h	A6h	00h	15h					

а) Ако је PC=1000h, SP=1102h, ACC=10h, R0=14h, R1=07h, N=0, Z=0, V=0 и C=0. Извршити 5 сукцесивних наредби и за сваку наредбу одредити следеће: у фази читања инструкције меморијске адресе са којих је прочитана инструкција, садржај регистра IR, инструкцију која је прочитана и нови садржај регистра PC, у фази одређивања адресе операнда и читања операнда меморијске адресе са којих је прочитана адреса операнда, меморијске адресе са којих је прочитан операнд, вредност операнда и нови садржај регистара опште намене који су у овој фази промењени, у фази извршавања меморијске адресе којима се у овој фази приступа, садржај акумулатора A и индикатора N, Z, V и C након извршавања инструкције и нови садржај регистара и меморијских локација који су у овој фази промењени. Резултате представити у форми једне или више табела које садрже наведене колоне.

Решење:

Фаза читања инструкције:

Рб инст.	Адресе са којих је прочитана инструкција	IR _{23..16}	IR _{15..8}	IR _{7..0}	Прочитана инструкција	Садржај промењених регистара
1.	1000, 1001, 1002	8C	F0	0F	LOAD (PC)F00Fh	PC=1003
2.	1003	B8	-	-	XOR (R0)	PC=1004
3.	1004 1005, 1006	6A	10	0A	JMP 100A	PC=1007
4.	100A, 100B	C2	80	-	LSR #80h	PC=100C
5.	100C, 100D, 100E	A6	00	15	ADD 15h	PC=100F

Фаза одређивања адресе операнда и читање операнда:

Рб инст.	Адресе са којих је прочитана адреса операнда	Адресе са којих је прочитан операнд	Операнд	Садржај промењених регистара
1.	-	0012 (1003+F00F)	FF	-
2.	-	0014	13	-
3.	-	-	-	-
4.	-	-	80	-
5.	-	0015	7F	-

Фаза извршавања:

Рб инст.	Адресе којима се приступа у фази извршавања	A	N	Z	V	C	Садржај промењених регистара и меморијских локација
1.	-	FF	1	0	0	0	-

2.	-	ЕС	1	0	0	0	-
3.	-	ЕС	1	0	0	0	PC=100A
4.	-	40	0	0	0	0	-
5.	-	BF	1	0	1	0	-

Задатак 9.

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 дужине 2 бајта представљени у другом комплементу. Подаци у меморији заузимају две суседне меморијске локације, при чему се млађи бајт налази на нижој а старији бајт на вишој адреси.

У процесору постоји програмски бројач PC дужине 2 бајта, адресни регистар меморије MAR дужине 2 бајта, прихватни регистар податка меморије MBR дужине 1 бајт, прихватни регистар инструкције IR дужине 3 бајта, акумулатор A дужине 2 бајта, прихватни регистар податка B дужине 2 бајта, регистри опште намене R0 и R3 дужине 2 бајта, програмска статусна реч PSW дужине 1 бајт. Инструкције су дужине 1 или 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције скока, док се битовима 3 до 0 првог бајта инструкција специфицира код операције за инструкције скока и то на следећи начин: 0001-инструкција условног скока уколико је резултат негативан (JN), 0010-инструкција безусловног скока (JMP) и 0100-инструкција скока на потпрограма (JSR). Адреса скока је дата 2 и 3 бајтом инструкције, при чему је млађи бајт адресе скока дат другим а старији бајт трећим бајтом. Дужина инструкција је 3 бајта.

Битови 7, 6, 5 и 4 првог бајта инструкције су 1111 за безадресне инструкције, док се битовима 3 до 0 првог бајта инструкција специфицира код операције за безадресне инструкције и то на следећи начин: 0010-инструкција повратка из потпрограма (RTS) и 0101-инструкција повратка из прекидне рутине (RTI). Дужина инструкција је 1 бајт.

Битови 7, 6, 5 и 4 првог бајта инструкције у опсегу вредности 0001 до 1110 специфицирају код операције за адресне инструкције и то на следећи начин: 0001-инструкција преноса у акумулатор (LOAD), 0011-инструкције преноса из акумулатора (STORE), 1001-аритметичка инструкција одузимања (SUB), 1100-логичка инструкција логичка сума (OR) и 0110-инструкција аритметичког померања удесно за једно место место код које резултат остаје само у акумулатору (ASR). Начини адресирања су специфицирани битовима 3 и 2 првог бајта инструкције и то на следећи начин: 00-непосредно адресирање, 01-меморијско директно адресирање, 10-регистарско индиректно адресирање са померајем и 11-PC релативно адресирање. Код непосредног адресирања 16 битни операнд је дат другим и трећим бајтом инструкције, при чему је млађи бајт податка дат другим а старији бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се не користе. 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. Код регистарског индиректног адресирања са померајем 16 битни померај је дат другим и трећим бајтом инструкције, при чему је млађи бајт помераја дат другим а старији бајт трећим бајтом. Битови 1 и 0 првог бајта инструкције се користе за адресирање једног од регистара опште намене R0 до R3. Код PC релативног адресирања 16 битни померај је дат другим и трећим бајтом инструкције, при чему је млађи бајт помераја 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 адресама, а SP показује на последњу заузету локацију.

На индикаторе у програмској статусној речи утичу следеће инструкције: LOAD и STORE на N и Z, SUB на N, Z, V и C, OR на N и Z и ASR на N, Z и C.

а) Ако је PC=1000h, ACC=CEh, SP = 4000h, R0=05h, R1=01h, R2=07h, R3=08h, mem(0h)=0h, mem(1h)=1h, mem(2h)=2h, mem(3h)=3h, mem(4h)=4h, mem(5h)=5h, mem(6h)=6h, mem(7h)=7h, mem(8h)=8h, N=0, Z=0, V=0 и C=0 одредити садржај релевантних меморијских локација ако се извршава следећи програмски сегмент: LOAD #5h, STORE (R1)03h, JN 100Ch, SUB 7h.

б) Извршити 4 сукцесивних наредби и за сваку наредбу одредити следеће: у фази читања инструкције меморијске адресе са којих је прочитана инструкција, садржај регистра IR, инструкцију која је прочитана и нови садржај регистра PC, у фази одређивања адресе операнда и читања операнда меморијске адресе са којих је прочитана адреса операнда, меморијске адресе са којих је прочитан операнд, вредност операнда и нови садржај регистара опште намене који су у овој фази промењени, у фази извршавања меморијске адресе којима се у овој фази приступа, садржај акумулатора A и индикатора N, Z, V и C након извршавања инструкције и нови садржај регистара и меморијских локација који су у овој фази промењени. Резултате представити у форми једне или више табела које садрже наведене колоне.

Решење:

а)

ЛОКАЦИЈА:	00h	01h	02h	03h	04h	05h	06h	07h	08h
САДРЖАЈ:	00h	01h	02h	03h	05h	00h	06h	07h	08h

ЛОКАЦИЈА:	1000h	1001h	1002h	1003h	1004h	1005h	1006h	1007h	1008h	1009h
САДРЖАЈ:	10h	05h	00h	39h	03h	00h	01h	0Ch	10h	94h
ЛОКАЦИЈА:	100Ah	100Bh								
САДРЖАЈ:	07h	00h								

б)

Фаза читања инструкције:

Рб инст.	Адресе са којих је прочитана инструкција	IR _{23..16}	IR _{15..8}	IR _{7..0}	Прочитана инструкција	Садржај промењених регистара
1.	1000, 1001, 1002	10	05	00	LOAD #5h	PC=1003
2.	1003, 1004, 1005	39	03	00	STORE (R1)03h	PC=1006
3.	1006, 1007, 1008	01	0C	10	JN 100Ch	PC=1009
4.	1009, 100A, 100B	94	07	00	SUB 7h	PC=100C

Фаза одређивања адресе операнда и читање операнда:

Рб инст.	Адресе са којих је прочитана адреса операнда	Адресе са којих је прочитан операнд	Операнд	Садржај промењених регистара
1.	-	-	0005	-
2.	-	-	-	-
3.	-	-	-	-
4.	-	0007, 0008	0807	-

Фаза извршавања:

Рб инст.	Адресе којима се приступа у фази извршавања	A	N	Z	V	C	Садржај промењених регистара и меморијских локација
1.	-	0005	0	0	0	0	-
2.	0004(1+3), 0005	0005	0	0	0	0	-
3.	-	0005	0	0	0	0	-
4.	-	F7FE	1	0	0	1	-

МЕХАНИЗАМ ПРЕКИДА

Задатак 10.

Посматра се процесор са векторисаним механизмом прекида. Адресе прекидних рутина се чувају у табели адреса прекидних рутина (IV табели) која почиње на адреси 0 и има 256 улаза. Капацитет оперативне меморије је 64KB, ширина речи меморије је један бајт, а 16 битни подаци се у меморију смештају тако да се на нижој адреси налази виши бајт, а на вишој адреси нижи бајт. Прекидне рутине за периферије PER1, PER2 и PER3 почињу на адресама 5678h, 3456h и 1234h, респективно. Улази 5, 3 и 1 у IV табели су додељени периферијама PER1, PER2 и PER3, респективно, а периферије су иницијализоване да као импулс шаљу захтев за прекидом, као и да шаљу број улаза у табели прекидних рутина.

- Нацртати део оперативне меморије на којима се налазе улази 0 до 5 у IV табели, означити адресе релевантних меморијских локација и попунити их одговарајућим вредностима.
- Набројати корак по корак шта се све дешава у процесору од тренутка када је стигао захтев за прекид од периферије до тренутка када се у регистру PC налази почетна адреса прекидне рутине.
- Објаснити за сваки корак набројан у тачки б) да ли га обавља хардвер или софтвер.
- Објаснити како се обавља повратак из прекидне рутине и набројати кораке који се том приликом изводе.

Решење:

- Садржај дела оперативне меморије је приказан на слици.

број улаза у IV табелу	периферија	адреса	садржај
5	PER1	11	78h
		10	56h
4		9	
		8	
3	PER2	7	56h
		6	34h
2		5	
		4	
1	PER3	3	34h
		2	12h
0		1	
		0	

- Када стигне захтев за прекид најпре се заврши са извршавањем текуће инструкције. Након тога се наставља са извршавањем додатних корака који су потребни да би се ① сачувао контекст процесора и ② израчунала адреса прекидне рутине. Контекст процесора (PC, PSW и програмски доступни регистри) се чува на врху стека. Адреса прекидне рутине се рачуна следећом секвенцом акција:

процесор шаље **inta** сигнал (interrupt acknowledge) периферији,
 периферија одговара тако што процесору пошаље свој број улаза у IV табелу,
 процесор израчуна адресу улаза у IV табелу (број улаза се претвори у померај и дода се на садржај IVTP регистра) и
 прочита се адреса прекидне рутине из IV табеле и смести се у PC.
 Треба додати да веома често након корака ① и ② процесор обавља и корак ③. У овом кораку:
 бит I (маскирајући прекиди дозвољени) у регистру PSW се ресетује,
 бит T (прекид после сваке инструкције) у регистру PSW се ресетује и
 у битове L (приоритет текућег програма) у регистру PSW се уписује приоритет прекидне рутине на коју се скаче.

- Одговор у тачки б) подразумева да се сви кораци изводе хардверски. Међутим, неко ид корака под ① могу да се изведу софтверски. Ово је случај код програмски доступних регистара и понекад код регистра PSW које у том случају треба програмски сачувати на почетку прекидне рутине. Програмски доступни регистри се обично чувају софтверски ако их је много. У супротном, они се чувају хардверски. Регистар PSW се мора чувати хардверски ако постоји корак ③. У супротном, може се чувати или хардверски или софтверски. У већину случајева регистар PSW се чува хардверски. Кораци ② и ③ се увек изводе хардверски.

- Повратак из прекидне рутине се изводи посебном инструкцијом **RTI** (return from interrupt). Ова инструкција естаурира са стека контекст процесора:

програмски доступни регистри се рестаурирају ако су сачувани хардверски,

регистар PSW се рестаурира ако је сачуван хардверски и регистар PC је рестауриран.

Ако се програмски доступни регистри и PSW не рестаурирају инструкцијом RTI, морају се рестаурирати софтверски на крају прекидне рутине пре инструкције RTI.

Задатак 11.

Меморија неког рачунара је капацитета 4G (giga) 16 битних речи. Адресибилна јединица је 16 битна реч, а 32 битни бројеви се у меморију смештају тако да је на нижој адреси нижих 16 бита. Процесор је једноадресни, улазно-излазни и меморијски адресни простори су раздвојени, а механизам прекида је векторисан. Интерапт вектор (IV) табела заузима најнижи део меморијског адресног простора. На процесор су везане три периферије, PER1, PER2 и PER3 којима треба доделити улазе 3, 5 и 7 у вектор табели, и којима одговарају прекидне рутине на адресама 12345678h, 23456789h и 3456789Ah, респективно. Адресе 16 битних регистра у којима се чувају бројеви улаза су 0h, 4h и 8h, респективно.

- а) Написати део програма којим се додељују бројеви улаза наведеним периферијама.
- б) Нацртати изглед првих 8 улаза у вектор табели, означити адресе релевантних локација и уписати садржаје у њих.
- в) Написати део програма којим се иницијализује улаз 5 у вектор табели.
- г) Коју вредност шаље процесору периферија PER2 када јој процесор одобри захтев за прекидом?
- д) Описати и представити програмом поступак којим се и периферији PER2 додељује иста прекидна рутина као и периферији PER3.
- ђ) Колико улаза има IV табела?

Решење

- | | | | | |
|--|--|-------------|---|------------------------|
| <p>а) LOAD #3
 OUT 0h
 LOAD #5
 OUT 4h
 LOAD #7
 OUT 8h</p> | <p>в) LOAD #2345h
 STORE 11
 LOAD #6789h
 STORE 10</p> | <p>г) 5</p> | <p>д) LOAD #3456h
 STORE 11
 LOAD #789Ah
 STORE 10
 ili:
 LOAD #7
 OUT 4h</p> | <p>ђ) 64К улаза</p> |
|--|--|-------------|---|------------------------|

б)

број улаза у IV табелу	периферија	адреса	садржај
7	PER3	15	3456h
		14	789Ah
6		13	
		12	
5	PER2	11	2345h
		10	6789h
4		9	
		8	
3	PER1	7	1234h
		6	5678h
2		5	
		4	

Задатак 12.

Адресни простор неког рачунара је величине 16GB, адресибилна јединица је 32 битна реч, подаци су дужине 32 бита. Процесор је једноадресни, улазно-излазни и меморијски адресни простори су раздвојени, а механизам прекида је векторисан. IV (Interrupt Vector) табела има 4 улаза и почиње од адресе 0. Процесор поседује три улаза за спољне маскирајуће прекиде, IRQ1, IRQ2 и IRQ3, при чему је IRQ3 највишег приоритета а IRQ1 најнижег приоритета, на које су везане периферије PER1, PER2 и PER3, респективно. Прекидне рутине ових периферија започињу на адресама 1000h, 1007h и 1014h, респективно. У PSW постоји бит I (Interrupt Enable) у разреду 2 који се брише у корака за обраду прекида и бити L1..0 у разредима 1 и 0 који садржи ниво приоритета текућег извршавања. Не постоји селективно маскирање прекида. Процесор не прихвата прекид истог нивоа као што је текући. Инструкције INT, INTE и RTI не реагују на прекид. При прекиду се на стеку чувају ACC, PC и PSW тим редом. Адресе регистра у којима се чувају бројеви улаза PER1, PER2 и PER3 су 8h, 20h и 32h, респективно. Главни програм је дат на слици 1, прекидне рутине на Слици 2, а садржај дела меморијског адресног простора почев од адресе 0 дат је на Слици 3. Инструкција на адреси 0100h означена је као 1. (прва) по редоследу извршавања, а свака следећа инструкција која се извршава означена је следећим редним бројем. У току извршавања 1. инструкције стиже захтев за прекид од периферије PER1, у току 5. од периферије PER2, а у току 11. од периферије PER3. На почетку су сви бити PSW-а постављени на 0. Почетни садржај акумулатора је 2h, локације 13h је 6h, а локације 23h је 1h. Стек расте од виших ка нижим адресама, а SP указује на прву слободну локацију. Све вредности на сликама су хексадецималне.

Слика 1	Адреса	Инструкција	Слика 2	Адреса	Инструкција	Адреса	Инструкција	Слика 3	Адреса	Садржај
	0100	INTE	1000	INT #3	100A	STORE 13h		0000h	1000h	
	0101	INCA	1001	INTE	100C	RTI		0001h	1007h	
	0102	AND #6	1002	INC	100D	POP		0002h	1014h	
	0104	OR #6	1003	INC	100E	AND #FFFFFFCh		0003h	100Dh	
			1004	STORE 15h	1010	OR 23h		0004h	1004h	
			1006	RTI	1012	PUSH		0005h	1002h	
			1007	LOAD 13h	1013	RTI				
			1009	INC	1014	INC				
					1015	RTI				

- а) По којим линијама периферија PER1 шаље број улаза у IV табелу?
- б) Која вредност се налази на магистрала података у циклусу из тачке а)?
- в) Написати секвенцу инструкција којом се периферији PER3 додељује број улаза у IVT.
- г) Дати секвенцу адреса инструкција које се редом извршавају по датом сценарију. Резултат дати табеларно тако да табела садржи редни број интрукције, адресу на којој започиње инструкција, саму инструкцију, садржај акумулатора накој извршење инструкције, вредности свих познатих бита унутар програмске статусне речи.
- д) Поновити поступак под г) уколико је почетни садржај локације 23h имао вредност 2.

Решење

а) Ова периферија шаље преко магистрале података свој број улаза у IVT.

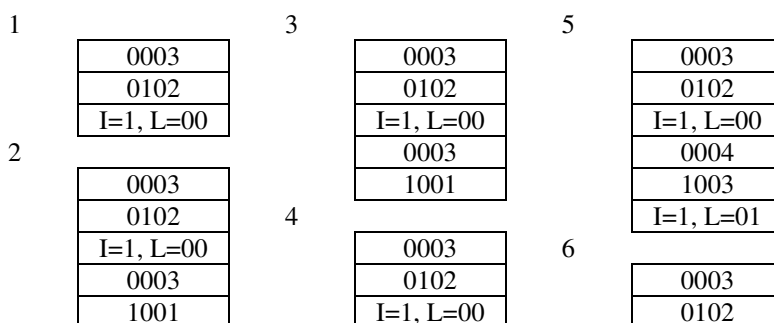
б) 0

в) LOAD #2; OUT 32h

г)

MEM[23h] = 1h

Рб	Адреса	Инструкција	ACC	Стек	I	L	PRIRR3	PRIRR2	PRIRR1
1	0100	INTE	2	-	1	00	-	-	1
2	0101	INCA	3	-	1	00	-	-	1
			3	1	1	01	-	-	-
3	1000	INT #3	3	2	0	01	-	-	-
4	100D	POP	X001b	3	0	01	-	-	-
5	100E	AND #FFFFFFCh	X000b	3	0	01	-	1	-
6	1010	OR 23h	X001b	3	0	01	-	1	-
7	1012	PUSH	X001b	4	0	01	-	1	-
8	1013	RTI	3	1	0	01	-	1	-
9	1001	INTE	3	1	1	01	-	1	-
10	1002	INC	4	1	1	01	-	1	-
			4	5	0	10	-	-	-
11	1007	LOAD 13h	6	5	0	10	1	-	-
12	1009	INC	7	5	0	10	1	-	-
13	100A	STORE 13h	7	5	0	10	1	-	-
14	100C	RTI	4	1	1	01	1	-	-
15	1003	INC	5	1	1	01	1	-	-
			5	6	0	11	-	-	-
16	1014	INC	6	6	0	11	-	-	-
17	1015	RTI	5	1	1	01	-	-	-
18	1004	STORE 15h	5	1	1	01	-	-	-
19	1006	RTI	3	-	1	00	-	-	-
20	0102	AND #6	2	-	1	00	-	-	-
21	0104	OR #6	6	-	1	00	-	-	-



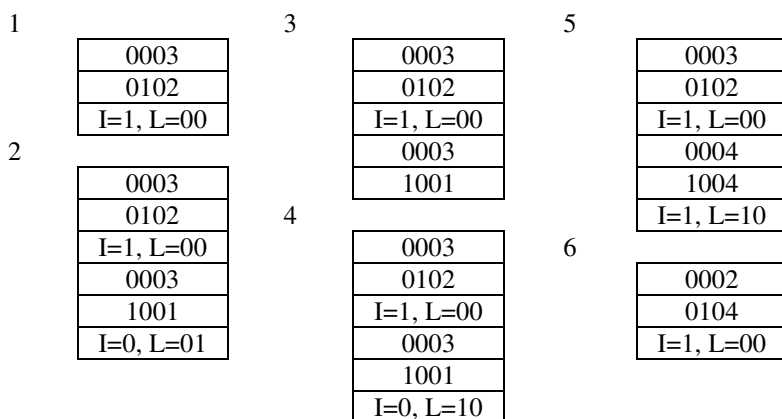
I=0, L=01

0003
1001
I=0, L=01

I=1, L=00
0005
1004
I=1, L=01

д)
MEM[23h] = 2h

Рб	Адреса	Инструкција	ACC	Стек	I	L	PRIRR3	PRIRR2	PRIRR1
1	0100	INTE	2	-	1	00	-	-	1
2	0101	INCA	3	-	1	00	-	-	1
			3	1	1	01	-	-	-
3	1000	INT #3	3	2	0	01	-	-	-
4	100D	POP	X01b	3	0	01	-	-	-
5	100E	AND #FFFFFFCh	X00b	3	0	01	-	1	-
6	1010	OR 23h	X10b	3	0	01	-	1	-
7	1012	PUSH	X10b	4	0	01	-	1	-
8	1013	RTI	3	1	0	10	-	1	-
9	1001	INTE	3	1	1	10	-	1	-
10	1002	INC	4	1	1	10	-	1	-
11	1003	INC	5	1	1	10	1	1	-
			5	5	0	11	-	1	-
12	1014	INC	6	5	0	11	-	1	-
13	1015	RTI	5	1	1	10	-	1	-
14	1004	STORE 15h	5	1	1	10	-	1	-
15	1006	RTI	3	-	1	00	-	1	-
16	0102	AND #6	2	-	1	00	-	1	-
			2	6	0	10	-	-	-
17	1007	LOAD 13h	6	6	0	10	-	-	-
18	1009	INC	7	6	0	10	-	-	-
19	100A	STORE 13h	7	6	0	10	-	-	-
20	100C	RTI	2	-	1	10	-	-	-
21	0104	OR #6	6	-	1	10	-	-	-



Задатак 13.

Меморија неког рачунара је капацитета 64 KB. Адресибилна јединица је бајт, а 16-битни бројеви се у меморију смештају тако да је на нижој адреси виши бајт. Процесор је једноадресни и има наредбе које обављају пренос само једног бајта ка/из меморије, улазно/излазни и меморијски адресни простори су раздвојени, а механизам прекида је векторисан. Интерапт вектор табела има 8 улаза и почиње од адресе 0 оперативне меморије. Процесор има 4 улазне линије IRQ0 до IRQ3 за маскирајуће прекиде који се приоритирају, при чему је улаз IRQ0 највишег приоритета. Овим линијама додељени су фиксно улази 4 до 7 у вектор табели, респективно. Немаскирајући и интерни прекиди су вишег приоритета од маскирајућих, и заузимају улазе 0 до 3 у табели. У PSW процесора постоји бит I (Interrupt Enable) који се брише у микропрограму за обраду прекида, а програмски контролише наредбама INTE (Interrupt Enable, поставља I на 1) и INTD (Interrupt Disable, брише I), као и бити L2...0 који садрже ниво прекида који се тренутно опслужује. Такође постоји 4-битни регистар IMR (Interrupt Mask Register) код кога јединица на биту i значи да је дозвољен прекид

са линије IRQi. Овај регистар иницијално садржи све јединице. На линије IRQ1 и IRQ3 су везане периферије PER1 и PER3, респективно, а остале линије су слободне. Треба везати још уређај T—временски бројач који периодично генерише сигнал захтева за прекид. Потребно је обезбедити да се прекид од временског бројача може опслужити и у току извршавања неке од прекидних рутина за периферије PER1 и PER3. Ове рутине почињу на адресама 100h и 300h, респективно.

- а) На који улаз IRQ_{3...0} треба везати сигнал захтева за прекид временског бројача?
 б) Написати део програма којим се иницијализује улаз периферије PER1 у вектор табели.
 в) Прекидна рутина периферије PER1 је приказана на слици.

```
INTH1: IN    FF00h
        STORE (Dest)+
        ...
        RTI
```

Ако у току извршавања инструкције STORE стигне сигнал прекида од временског бројача, да ли ће се овај прекид одмах опслужити? Образложити одговор.

- г) Изменити дату прекидну рутину периферије PER1 тако да одговор на претходно питање буде супротан.
 д) Шта треба урадити у прекидној рутини периферије PER3 тако да се обезбеди да она не буде прекидана од захтева са периферије PER1, али да буде прекидана од захтева са временског бројача?

Решење

- | | | |
|---------|-----------|--|
| а) IRQ0 | б) | LOAD #01
STORE 0Ah
LOAD #00
STORE 0Bh |
| в) Не. | г) INTH1: | INTE
IN FF00h
STORE (Dest)+
...
RTI |

- д) Постављања бита I на 1, у IMR треба уписати bb01 (бинарно, b означава било коју вредност).

Задатак 14.

Адресни простор процесора је величине 16GB, адресибилна јединица је 32 битна реч, а 64 битни бројеви се смештају тако да је на нижој адреси нижа реч. Процесор је једноадресни, а механизам прекида је векторисан. Интерапт вектор (IV) табела има 4 фиксна улаза и почиње од адресе 2h. Процесор има једну улазну линију IRQM за спољне маскирајуће прекиде и једну улазну линију IRQN за спољне немаскирајуће прекиде. Њима су придружени улази 0 и 1 у IV табелу, респективно. Немаскирајући прекиди су вишег приоритета од маскирајућих. Улаз 2 у IV табели се употребљава у случају прекида после сваке инструкције (TRAP), а улаз 3 у свим осталим случајевима. PSW постоји бит I (Interrupt Enable) који се брише у кораку за обраду прекида. При прекиду се на стеку чувају PSW и PC тим редом. Стек расте према нижим локацијама. Акумулатор је 32 битни. Инструкције INTE, INTD, RTI, TRPE и TRPD не реагују на прекиде. Дат је део главног програма на слици 1, прекидне рутине на слици 2, изглед дела меморије почев од адресе 0 дат је на слици 3. Инструкција INTE на адреси 0100h означена је као 1. (прва) по редоследу извршавања, а свака следећа инструкција која се извршава означена је следећим редним бројем. У току извршавања 2. инструкције стиже захтев за прекид по линији IRQN, а у току 5. по линији IPQM. На почетку су сви бити PSW-а постављени на 0.

Слика 1	Адреса	Инструкција	Слика 2	Адреса	Инструкција	Адреса	Инструкција	Слика 3	Адреса	Садржај
	0100h	INTE		1000h	INC		1006h	STORE 1h	0000h	1001h
	0101h	LOAD #1h		1001h	PUSH		1007h	RTI	0001h	100Ah
	0102h	INC		1002h	POP		1008h	POP	0002h	1000h
	0103h	STORE #1h		1003h	RTI		1009h	INC	0003h	1004h
	0104h	DEC		1004h	LOAD 1h		100Ah	PUSH	0004h	1006h
	0105h	INTD		1005h	OR #FFh		100Bh	RTI	0005h	1008h

- а) На којим адресама започињу прекидне рутине за линије IRQM и IRQN, респективно?
 б) Написати секвенцу адреса наредби које се редом извршавају, почев од адресе 0100h. Резултат дати табеларно тако да табела садржи редни број инструкције, адресу на којој започиње инструкција, саму инструкцију, садржај акумулатора на кој извршење инструкције, вредности свих познатих бита унутар програмске статусне рехи, и вредности на линијама IRQ.
 в) Приказати садржај свих познатих локација на врху стека након извршавања 7. инструкције. За сачувану вредност PSW дати само вредност бита I. Назначити у коме смеру расте стек.

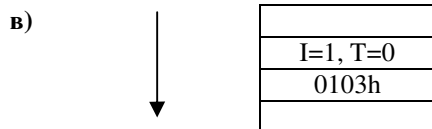
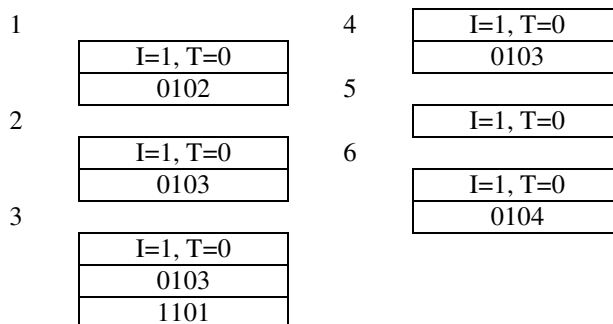
г) Која ће се вредност налазити на локацији 1h након извршења секвенце под б)?

Решење

а) IRQM – 1000h, IRQN – 1004h

б)

P6	Адреса	Инструкција	ACC	Стек	I	T	PRIRRN	PRIRRM
0	-	-	?	-	0	0	-	-
1	0100	INTE	?	-	1	0	-	-
2	0101	LOAD #1	1	-	1	0	1	-
			1	1	0	0	-	-
3	1004	LOAD 1	100A	1	0	0	-	-
4	1005	OR #FFh	10FF	1	0	0	-	-
5	1006	STORE 1	10FF	1	0	0	-	1
6	1007	RTI	10FF	-	1	0	-	1
7	0102	INC	1100	-	1	0	-	1
			1100	2	0	0	-	-
8	1000	INC	1101	2	0	0	-	-
9	1001	PUSH	1101	3	0	0	-	-
10	1002	POP	1101	2	0	0	-	-
11	1003	RTI	1101	-	1	0	-	-
12	0103	STORE #1	1101	4	0	0	-	-
13	1008	POP	0103	5	0	0	-	-
14	1009	INC	0104	5	0	0	-	-
15	100A	PUSH	0104	6	0	0	-	-
16	100B	RTI	0104	-	1	0	-	-
17	0104	DEC	0103	-	1	0	-	-
18	0105	INTD	0103	-	0	0	-	-



г) MEM[1] = 10FFh

Задатак 15.

Адресни простор процесора је величине 16GB, адресибилна јединица је 32 битна реч, а 64 битни бројеви се смештају тако да је на нижој адреси нижа реч. Процесор је једноадресни, а механизам прекида је векторисан. Интерапт вектор (IV) табела има 4 фиксна улаза и почиње од адресе 2h. Процесор има две улазне линију IRQM1 и IRQM2 за спољне маскирајуће прекиде, при чему је IRQM2 вишег приоритета, и једну улазну линију IRQN за спољне немаскирајуће прекиде. Њима су придружени улази 0, 1 и 2 у IV табелу, респективно. Немаскирајући прекиди су вишег приоритета од маскирајућих. Улаз 3 се користи у свим осталим случајевима. У PSW-у постоје бити I (Interrupt Enable) и T (Trap) који се бришу у кораку за обраду прекида, као и одређен број L бита. При прекиду се на стеку чувају PSW и PC тим редом. Стек расте према нижим локацијама. Акумулатор је 32 битни. Инструкције INTE, INTD, RTI, TRPE и TRPD не реагују на прекиде. Не прихвата се прекид истог нивоа приоритета. Не постоји регистар маске IMR. Дат је део главног програма на слици 1, прекидне рутине на слици 2, изглед дела меморије почев од адресе 0 дат је на слици 3. Инструкција TRPE на адреси 0100h означена је као 1. (прва) по редоследу извршавања, а свака следећа инструкција која се извршава означена је следећим редним бројем. У току извршавања 2. инструкције стиже захтев за

прекид по линији IRQM1, у току 5. по линији IRQN, а у току 9. по линији IRQM2. На почетку су сви бити PSW-а постављени на 0.

Слика 1	Адреса	Инструкција	Слика 2	Адреса	Инструкција	Адреса	Инструкција	Слика 3	Адреса	Садржај
	0100h	TRPE		1000h	INC		1006h	INC	0000h	1002h
	0101h	INTE		1001h	RTI		1007h	RTI	0001h	1008h
	0102h	LOAD #1h		1002h	POP		1008h	LOAD 1h	0002h	1009h
	0103h	INTD		1003h	PUSH		1009h	INC	0003h	1006h
	0104h	STORE 1h		1004h	DEC		100Ah	INC	0004h	1004h
	0105h	TRPD		1005h	RTI		100Bh	RTI	0005h	1000h

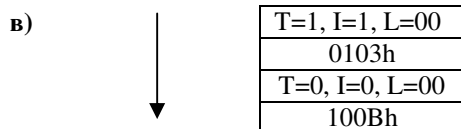
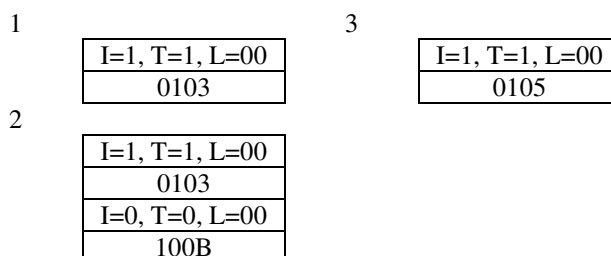
- а) На којим адресама започињу прекидне рутине за линије IRQM1, IRQM2 и IRQN, респективно?
- б) Написати секвенцу адреса наредби које се редом извршавају, почев од адресе 0100h. Резултат дати табеларно тако да табела садржи редни број интрукције, адресу на којој започиње инструкција, саму инструкцију, садржај акумулатора након извршења инструкције, вредности свих познатих бита унутар програмске статусне речи, и вредности на линијама IRQ.
- в) Приказати садржај свих познатих локација на врху стека након извршавања 6. инструкције. За сачувану вредност PSW дати само вредности бита I, T и L. Назначити у коме смеру расте стек.
- г) Која ће се вредност налазити на локацији 1h након извршења секвенце под б)?
- д) Нацртати структурну шему мреже за разрешавање захтева за маскирајућим прекидом у процесору. Ова мрежа треба да генерише сигнал INTRQ логичког услова који говори да постоји маскирајући прекид који треба опслужити, а помоћу вредности бита из регистра PSW и вредности са улаза IRQM1 и IRQM2.

Решење

а) IRQM1 – 1009h, IRQM2 – 1006h, IRQN – 1004h

б)

Pб	Адреса	Инструкција	ACC	Стек	I	T	L	PRIRRN	PRIRRM2	PRIRRM1
0	-	-	?	-	0	0	00	-	-	-
1	100	TRPE	?	-	0	1	00	-	-	-
2	101	INTE	?	-	1	1	00	-	-	1
3	102	LOAD #1	1	-	1	1	00	-	-	1
			1	1	0	0	10	-	-	-
4	1009	INC	2	1	0	0	10	-	-	-
5	100A	INC	3	1	0	0	10	1	-	-
			3	2	0	0	10	-	-	-
6	1004	DEC	2	2	0	0	10	-	-	-
7	1005	RTI	2	1	0	0	10	-	-	-
8	100B	RTI	2	-	1	1	00	-	-	-
9	103	INTD	2	-	0	1	00	-	1	-
10	104	STORE 1h	2	-	0	1	00	-	1	-
			2	3	0	0	00	-	1	-
11	1000	INC	3	3	0	0	00	-	1	-
12	1001	RTI	3	-	0	1	00	-	1	-
13	105	TRPD	3	-	0	0	00	-	1	-



г) MEM[1] = 2h

д) Погледати приручник за лабораторијске вежбе.

Задатак 16.

Адресни простор неког рачунара је величине 64 КВ, адресибилна јединица је бајт, а 16 битни бројеви се у меморију смештају тако да је на нижој адреси нижи бајт. Механизам прекида је векторисан. IV (Interrupt Vector) табела почиње од адресе на коју указује регистар IVTP. Трап прекиду припада фиксно улаз 2 у IVT. При прекиду се на стеку чувају PSW и PC тим редом. Стек расте према нижим локацијама. Сви регистри опште намене (Ri) су 8 битни. Процесор поседује инструкције TRPE (Trap Enable) и TRPD (Trap Disable) за дозволу, односно забрану trap-a. Инструкције RTI, TRPE и TRPD не реагују на прекиде (ни на trap). Дат је део програма на слици 1, trap прекидна рутина на слици 2, и део меморије почев од адресе 0 на слици 3. За време извршавања датог дела програма, нема других прекида осим trap-a.

Слика 1	Адреса	Инструкција	Слика 2	Адреса	Инструкција	Слика 3	Адреса	Садржај	Слика 4	Адреса	Инструкција
	FF00h	TRPD		00A0h	POP R1		0000h	00h		00A0h	POP R1
	FF01h	MOV R0, #2		00A1h	PUSH R1		0001h	A0h		00A1h	PUSH #2
	FF03h	TRPE		00A2h	RTI		0002h	00h		00A3h	RTI
	FF04h	DEC R0					0003h	0Ah			
	FF05h	JNZ FF03h					0004h	00h			
	FF08h	TRPD					0005h	A0h			
							0006h	00h			

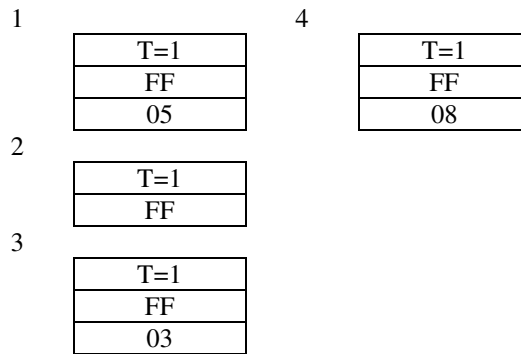
- а) Која се вредност налази на адреси FF07h?
 б) Која се вредност налази у регистру IVTP (та вредност је већа или једнака од 0, а мања од 7)?
 в) Написати секвенцу адреса наредби које се редом извршавају, почев од FF00h, закључно са FF08h. Резултат дати табеларно тако да табела садржи редни број интрукције, адресу на којој започиње инструкција, саму инструкцију, садржај акумулатора након извршења инструкције, вредности свих познатих бита унутар програмске статусне речи, и вредности на линијама IRQ.
 г) Која се вредност налази у регистру R1 непосредно пре првог извршавања наредбе RTI на адреси 00A2h?
 е) Написати секвенцу адреса наредби које се редом извршавају, почев од FF00h, све док су адресе познате, ако се уместо trap рутине са слике 2 користи рутина са слике 4.

Решење:

- а) FF07h садржај FFh.
 б) IVTP једнако 1.
 в)

Рб	Адреса	Инструкција	R0	R1	Стек	T
0	-	-	?	?	-	0
1	FF00	TRPD	?	?	-	0
2	FF01	MOV R0, #2	2	?	-	0
3	FF03	TRPE	2	?	-	1
4	FF04	DEC R0	1	?	-	1
			1	?	1	0
5	00A0	POP R1	1	5	2	0
6	00A1	PUSH R1	1	5	1	0
7	00A2	RTI	1	5	-	1
8	FF05	JNZ FF03h	1	5	-	1
			1	5	3	0
9	00A0	POP R1	1	3	2	0
10	00A1	PUSH R1	1	3	3	0
11	00A2	RTI	1	3	-	1
12	FF03	TRPE	1	3	-	1
13	FF04	DEC R0	0	3	-	1
			0	3	1	0
14	00A0	POP R1	0	5	2	0
15	00A1	PUSH R1	0	5	1	0
16	00A2	RTI	0	5	-	1
17	FF05	JNZ FF03h	0	5	-	1
			0	5	4	0
18	00A0	POP R1	0	8	2	0
19	00A1	PUSH R1	0	8	4	0

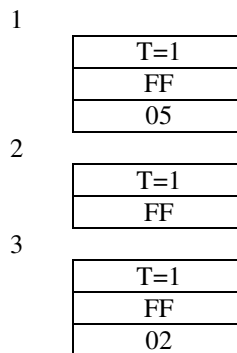
20	00A2	RTI	0	8	-	1
21	FF08	TRPD	0	8	-	0



г) R1 једнака 5.

д) Секвенца адреса је тако: FF00, FF01, FF03, FF04, 00A0, 00A1, 00A3, FF02, даље непознато.

Рб	Адреса	Инструкција	R0	R1	Стек	T
0	-	-	?	?	-	0
1	FF00	TRPD	?	?	-	0
2	FF01	MOV R0, #2	2	?	-	0
3	FF03	TRPE	2	?	-	1
4	FF04	DEC R0	1	?	-	1
			1	?	1	0
5	00A0	POP R1	1	5	2	0
6	00A1	PUSH #2	1	5	3	0
7	00A3	RTI	1	5	-	1
8	FF02	???				



Задатак 17.

Адресни простор процесора је величине 128KB, адресибилна јединица је 16 битна реч, а вишеречни бројеви се смештају тако да је на вишој адреси нижа реч. Процесор је једноадресни са раздвојеним меморијским и улазно/излазним адресним просторима, а механизам прекида је векторисан. IV (Interrupt Vector) табела почиње од адресе на коју указује регистар IVTP (Interrupt Vector Table Pointer), а регистар IVTP има вредност 2. Процесор има три улазне линије IRQ0, IRQ1 и IRQ2 за спољне маскирајуће прекиде, при чему је IRQ0 највишег приоритета, а IRQ2 најнижег приоритета, на које су везане периферије PER0, PER1 и PER2, респективно. Њима су придружени улази 2, 3 и 4 у IV табелу, респективно. Не прихвата се прекид истог нивоа приоритета. Адресе 16 битних регистара у којима се чувају бројеви улаза су 10h, 20h и 30h, респективно. У PSW-у постоји бит I (Interrupt Enable) који се брише у кораку за обраду прекида, као и одређен број L бита. При прекиду се на стеку чувају PSW и PC тим редом. Стек расте према нижим локацијама. Акумулатор је 16 битни. Инструкције INTE, INTD, RTI и INT не реагују на прекиде. Инструкција INT не мења ниво приоритета текућег програма. Дат је део главног програма на слици 1, прекидне рутине на слици 2, изглед дела меморије почев од адресе 0 дат је на слици 3. Инструкција на адреси 0100h означена је као 1. (прва) по редоследу извршавања, а свака следећа инструкција која се извршава означена је следећим редним бројем. У току извршавања 2. инструкције стиже захтев за прекид по линији IRQ2, а у току 5. по линији IRQ0. На почетку су сви бити PSW-а постављени на 0. Не постоји IMR регистар.

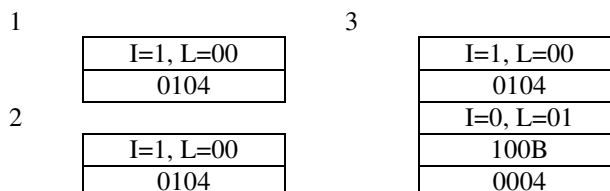
Слика 1	Адреса	Инструкција	Слика 2	Адреса	Инструкција	Адреса	Инструкција	Слика 3	Адреса	Садржај
	0100h	LOAD #2h		1000h	PUSH		1008h	RTI	0000h	0000h
	0102h	INTE		1001h	INTE		1009h	INTE	0001h	0001h
	0103h	INC		1002h	LOAD 1h		100Ah	INC	0002h	1000h
	0104h	ADD #2h		1004h	INC		100Bh	RTI	0003h	1008h
	0106h	INT #3h		1005h	STORE 1h		100Ch	INC	0004h	1000h
	0108h	INTD		1007h	POP		100Dh	RTI	0005h	100Ch
									0006h	1009h

- а) На којим адресама започињу прекидне рутине за линије IRQ0, IRQ1 и IRQ2, респективно?
- б) Написати део програма којим се иницијализује улаз 3 у вектор табели.
- в) Написати део програма којим се додељују бројеви улаза наведеним периферијама.
- г) Дати секвенцу адреса инструкција које се редом извршавају по датом сценарију. Резултат дати табеларно тако да табела садржи редни број интрукције, адресу на којој започиње инструкција, саму инструкцију, садржај акумулатора накој извршење инструкције, вредности свих познатих бита унутар програмске статусне реше, и вредности на линијама IRQ.
- д) Приказати садржај свих познатих локација на врху стека након извршавања б. инструкције. За сачувану вредност PSW дати само вредности бита I и L. Назначити у коме смеру расте стек.

Решење:

- а) IRQ0 – 1000h, IRQ1 – 100Ch, IRQ2 – 1009h
- б) LOAD #100Ch
STORE 5h
- в) LOAD #2h
OUT 10h
LOAD #3h
OUT 20h
LOAD #4h
OUT 30h
- г)

Рб	Адреса	Инструкција	ACC	Стек	I	L	PRIRR0	PRIRR1	PRIRR2
0	-	-	?	-	0	00	-	-	-
1	0100	LOAD #2	2	-	0	00	-	-	-
2	0102	INTE	2	-	1	00	-	-	1
3	0103	INC	3	-	1	00	-	-	1
			3	1	0	01	-	-	-
4	1009	INTE	3	1	1	01	-	-	-
5	100A	INC	4	1	1	01	1	-	-
			4	2	0	11	-	-	-
6	1000	PUSH	4	3	0	11	-	-	-
7	1001	INTE	4	3	1	11	-	-	-
8	1002	LOAD 1h	1	3	1	11	-	-	-
9	1004	INC	2	3	1	11	-	-	-
10	1005	STORE 1h	2	3	1	11	-	-	-
11	1007	POP	4	2	1	11	-	-	-
12	1008	RTI	4	1	1	01	-	-	-
13	100B	RTI	4	-	1	00	-	-	-
14	0104	ADD #2	6	-	1	00	-	-	-
15	0106	INT #3	6	4	0	00	-	-	-
16	100C	INC	7	4	0	00	-	-	-
17	100D	RTI	7	-	1	00	-	-	-
18	0108	INTD	7	-	0	00	-	-	-



I=1, L=01
100B

4

I=1, L=00
0108

д)



I=1, L=00
0104h
I=1, L=01
100Bh
0004h

МЕМОРИЈА

Задатак 18.

Посматра се рачунар код кога је улазно/излазни адресни простор меморијски пресликан.

Адресни простор рачунара је 64К адреса, при чему је ширина адресибилне локације 8 бита. Нижих 32К адреса адресног простора је резервисано за RAM меморију, а виших 32К адреса за ROM меморију и контролере периферија. У оквиру виших 32К адреса адресног простора резервисаног за ROM меморију и контролере периферија, нижих 16К адреса је резервисано за ROM меморију и виших 16К адреса за контролере периферија.

Од целокупног опсега адреса резервисаног за RAM меморију попуњено је само најнижих 8К адреса, а од целокупног опсега адреса резервисаног за ROM меморију попуњено је само најнижих 8К адреса. Контролери периферија имају по 16 локација (регистара). У рачунару постоји само један контролер периферије, чије локације (регистри) попуњавају само најнижих 16адреса из целокупног опсега адреса резервисаног за контролере периферија.

а) Назначити опсег адреса у адресном простору, опсег адреса резервисан за RAM меморију, опсег адреса резервисан за ROM меморију и контролере периферија, опсег адреса резервисан за ROM меморију, опсег адреса резервисан за контролере периферија, опсег адреса попуњен RAM меморијом, опсег адреса попуњен ROM меморијом и опсег адреса попуњен контролером периферије. Резултат представити табеларно.

б) Реализовати модул RAM меморије који попуњава само најнижих 8К адреса од 32К адреса опсега адреса резервисаног за RAM меморију користећи чипове 256x4 бита (управљачки улази су RD, WR и CS).

в) Реализовати модул ROM меморије који попуњава само најнижих 8К адреса од 16К адреса опсега адреса резервисаног за ROM меморију користећи чипове 1024x2 бита (управљачки улази су RD и CS).

г) Реализовати део контролера периферије са локацијама (регистрима) које попуњавају само најнижих 16 адреса из целокупног опсега адреса резервисаног за контролере периферија.

д) Генерисати сигнал којим се детектује да је дата адреса из опсега адреса који нису попуњени RAM меморијом, ROM меморијом и контролером периферије.

Решење:

а)

A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	опсег адреса резервисан за RAM меморију
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	опсег адреса резервисан за ROM меморију
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	опсег адреса попуњен контролером периферије
1	1	0	0	0	0	0	0	0	0	0	0	0	0	1		
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	
1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	опсег адреса не попуњен контролером периферије
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	опсег адреса не попуњен контролером периферије
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

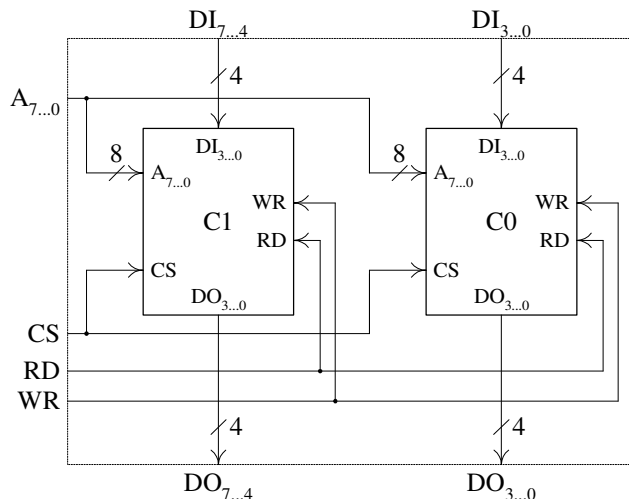
Адресе у опсегу адреса попуњеном контролером периферије

Опсежи адреса	Прва адреса	Последња адреса
---------------	-------------	-----------------

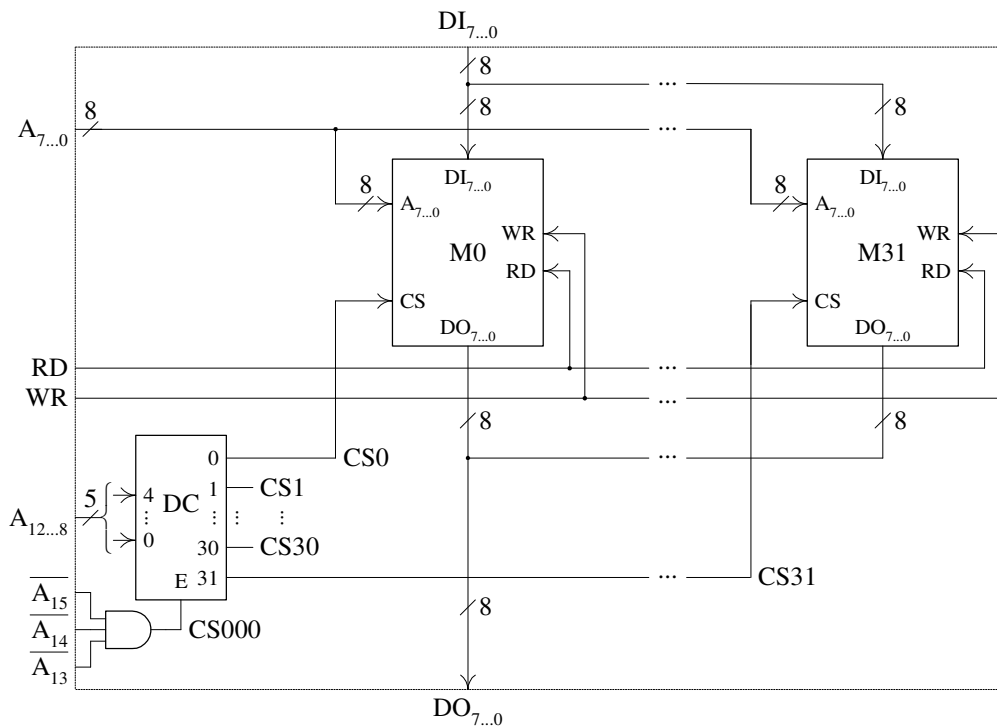
Опсег адреса адресног простора рачунара	0000h	FFFFh
Опсег адреса резервисан за RAM меморију	0000h	7FFFh
Опсег адреса резервисан за ROM меморију	8000h	BFFFh
Опсег адреса резервисан за контролере периферија	C000h	FFFFh
Опсег адреса попуњен RAM меморијом	0000h	1FFFh
Опсег адреса попуњен ROM меморијом	8000h	9FFFh
Опсег адреса попуњен контролером периферије	C000h	C00Fh

Опсези адреса

б)

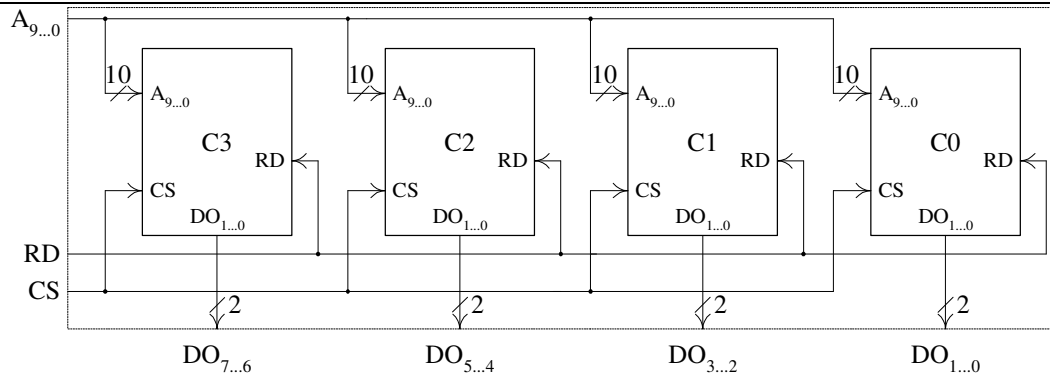


Модул RAM меморије капацитета 256x8 битова реализован чиповима 256x4 бита

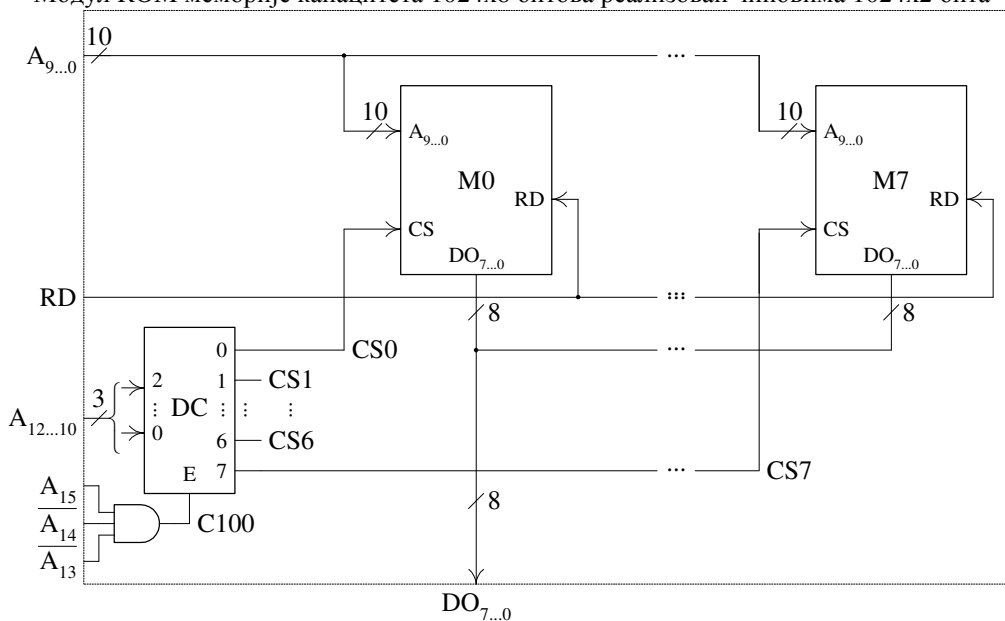


Модул RAM меморије капацитета 8Кx8 битова који попуњава само најнижих 8К адреса од 32К адреса опсега адреса резервисаног за RAM меморију користећи модуле RAM меморије капацитета 256x8 битова

в)

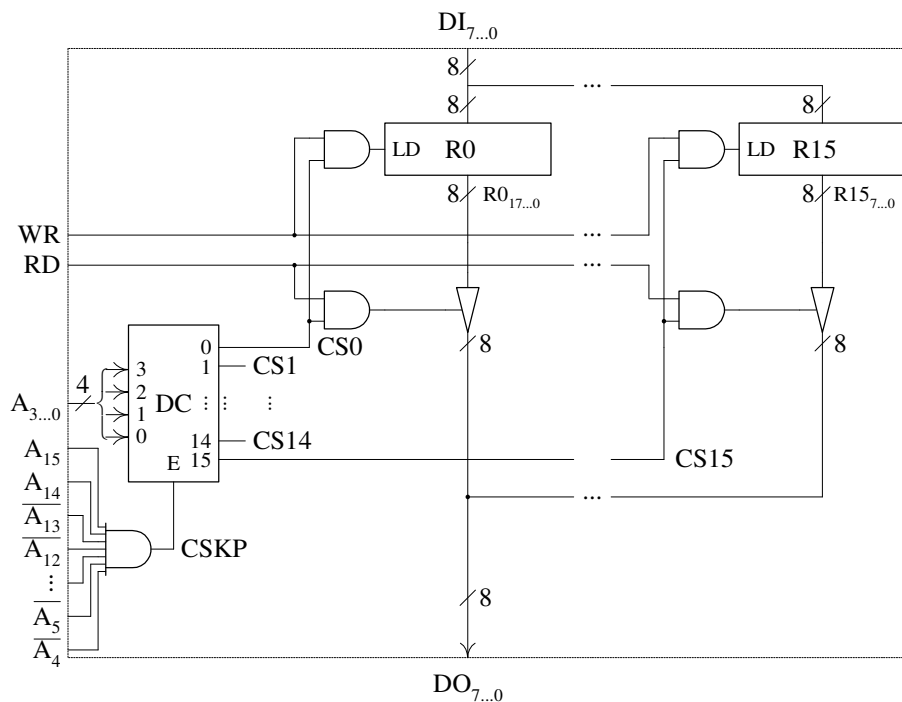


Модул ROM меморије капацитета 1024x8 бита реализован чиповима 1024x2 бита



Модул ROM меморије капацитета 8Kx8 бита која попуњава само најнижих 8K адреса од 32K адреса опсега адреса резервисаног за ROM меморију користећи модуле ROM меморије капацитета 1024x8 бита

г)



Део контролера са периферије

д) ...

Задатак 19.

Посматра се рачунар код кога су меморијски и улазно/излазни адресни простори раздвојени. Стога на магистрали постоји сигнал $\overline{M/IO}$ који вредностима 1 и 0 одређује да ли је адреса из меморијског или улазно/излазног адресног простора, респективно.

Меморијски адресни простор је 4G адреса, при чему је ширина адресбилне локације 16 бита. Нижих 3G адреса меморијског адресног простора је резервисано за RAM меморију, а виших 1G адреса меморијског адресног простора је резервисано за ROM меморију. Од целокупног опсега адреса резервисаног за RAM меморију попуњено је само виших 2G адреса, а од целокупног опсега адреса резервисаног за ROM меморију попуњено је само нижих 512M адреса.

Улазно/излазни адресни простор је 256 адреса, при чему је ширина адресбилне локације 16 бита.

а) Назначити опсег адреса у меморијском адресном простору, опсег адреса резервисан за RAM меморију, опсег адреса резервисан за ROM меморију, опсег адреса попуњен RAM меморијом и опсег адреса попуњен ROM меморијом. Назначити опсег адреса у улазно/излазном адресном простору рачунара, опсег адреса резервисан за контролере без директног приступа меморији, опсег адреса резервисан за контролере са директним приступом меморији, опсег адреса попуњен локацијама (регистрима) контролера без директног приступа меморији и опсег адреса попуњен локацијама (регистрима) контролера са директним приступом меморији. Резултат представити табеларно.

б) Реализовати модул RAM меморије који попуњава само виших 2G адреса од 3G адреса опсега адреса резервисаног за RAM меморију користећи чипове 512Mx8 битова (управљачки улази су RD, WR и CS).

в) Реализовати модул ROM меморије који попуњава само нижих 512M адреса од 1G адреса опсега адреса резервисаног за ROM меморију користећи чипове 128Mx1 бита (управљачки улази су RD и CS).

Решење:

а)

A ₃₁ A ₃₀ A ₂₉ A ₂₈		A ₇ A ₆ A ₅ A ₄	A ₃ A ₂ A ₁ A ₀	опсег адреса у меморијском адресном простору
0 0 0 0		0 0 0 0	0 0 0 0	
0 0 0 0		0 0 0 0	0 0 0 1	
- - - -	...	- - - -	- - - -	
1 1 1 1		1 1 1 1	1 1 1 0	
1 1 1 1		1 1 1 1	1 1 1 1	

Адресе у опсегу адреса меморијског адресног простора

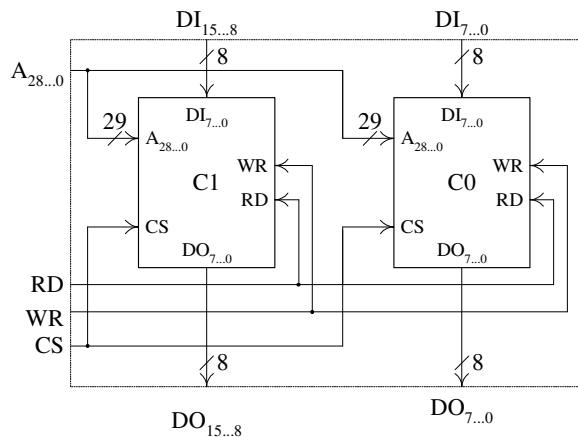
A ₇ A ₆ A ₅ A ₄	A ₃ A ₂ A ₁ A ₀	опсег адреса у улазно/излазном адресном простору рачунара
0 0 0 0	0 0 0 0	
0 0 0 0	0 0 0 1	
- - - -	- - - -	
1 1 1 1	1 1 1 0	
1 1 1 1	1 1 1 1	

Адресе у опсегу адреса улазно/излазног адресног простора

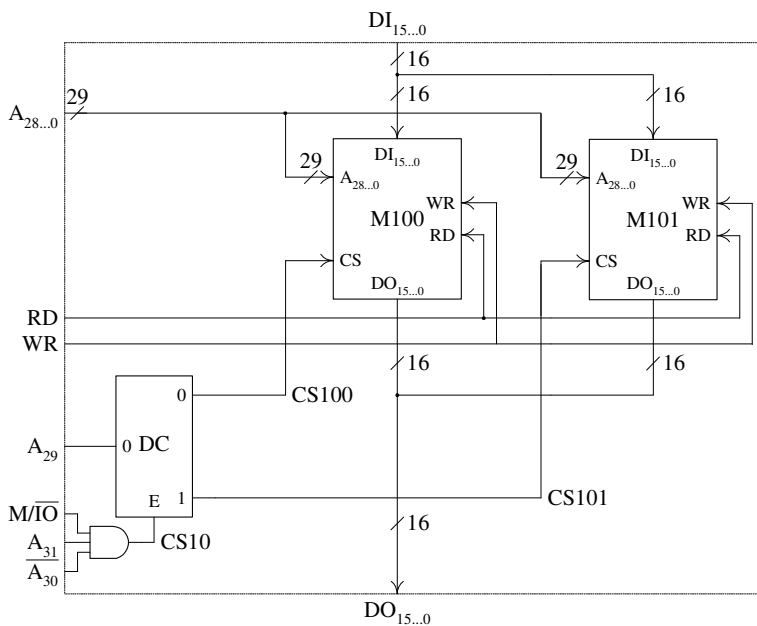
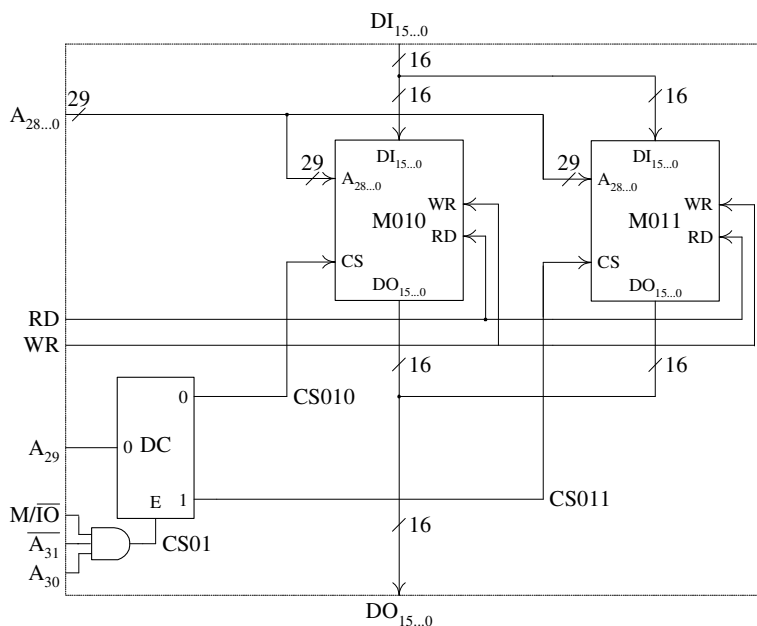
Опсези адреса	$\overline{M/IO}$	Прва адреса	Последња адреса
Опсег адреса меморијског адресног простора	1	0000 0000h	FFFF FFFFh
Опсег адреса резервисан за RAM меморију	1	0000 0000h	BFFF FFFFh
Опсег адреса резервисан за ROM меморију	1	C000 0000h	FFFF FFFFh
Опсег адреса попуњен RAM меморијом	1	4000 0000h	BFFF FFFFh
Опсег адреса попуњен ROM меморијом	1	C000 0000h	DFFF FFFFh
Опсег адреса улазно/излазног адресног простора	0	00h	FFh

Опсези адреса

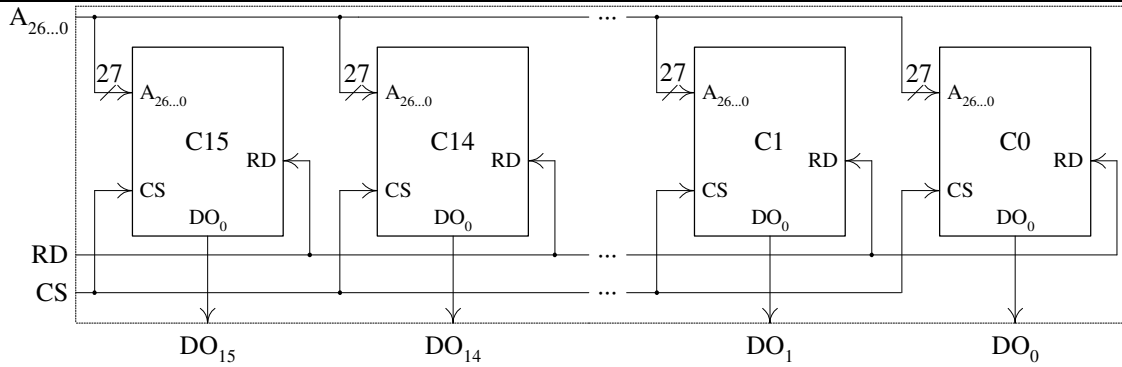
б)



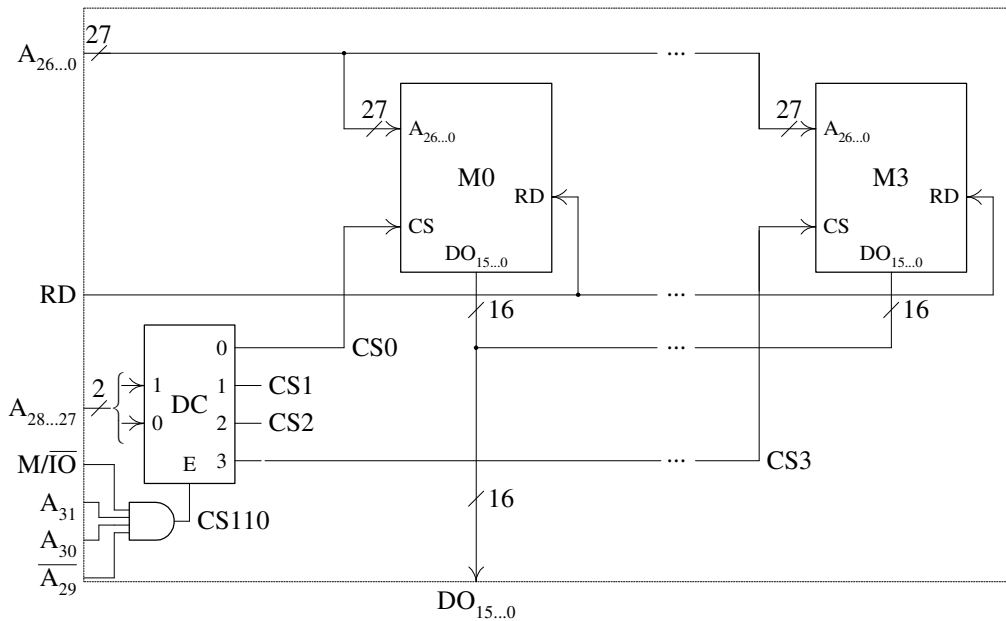
Модул RAM меморије капацитета 512Mx16 битава
реализован чиповима 512Mx8 битава



Два модула RAM меморије капацитета 1Gx16 бита која попуњавају само највиших 2G адреса од 3G адреса опсега адреса резервисаног за RAM меморију користећи модуле RAM меморије капацитета 512Mx16 бита



Модул ROM меморије капацитета 128Mx16 битова реализован чиповима 128Mx1 бита

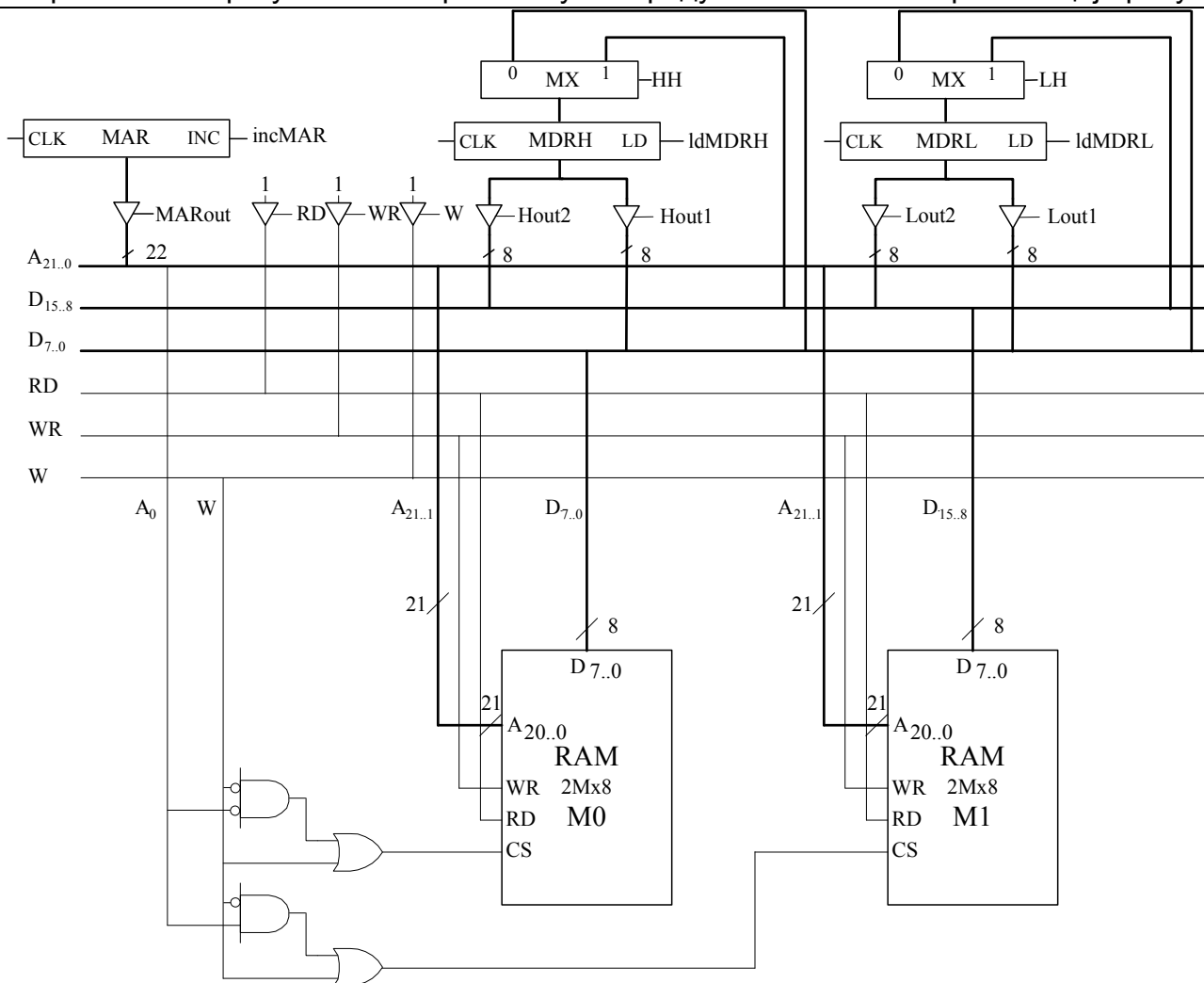


Модул ROM меморије капацитета 512Mx16 битова који попуњава само најнижих 512M адреса од 1G адреса опсега адреса резервисаног за ROM меморију реализован модулима ROM меморије капацитета 128Mx16 битова

Задатак 20.

Једноадресни процесор са раздвојеним адресним просторима комуницира са меморијом преко синхроне магистрале приказане на слици. Управљачка линија W има вредност 1 када се обавља трансфер 16-битне речи. Инструкција STOREB преноси садржај регистра MDRL у меморију на адресу која се налази у регистру MAR. Инструкција STOREW преноси садржај из регистра MDRH (виши бајт) и MDRL (нижи бајт) у меморију на адресу која се налази у регистру MAR. 16-битна реч се смешта тако да се нижи бајт смешта на нижу адресу. Претпоставити да циклус уписа у меморију траје један циклус такта.

- а) Колика је величина целог меморијског адресног простора? Колика је величина меморијских модула M0 и M1?
- б) Које адресе обухвата меморијски модул M0, а које меморијски модул M1?
- в) Колико циклуса такта траје фаза извршења следећих инструкција: (i) STOREW 8h, (ii) STOREW 5h.



Решење

- а) 4MB, 2MB, 2MB
- б) M0: 0h, 2h, ... 3FFFFFFh (парне адресе), M1: 1h, 3h,... 3FFFFFFh (непарне адресе).
- в) 1; 2.

МАГИСТРАЛА

Задатак 21.

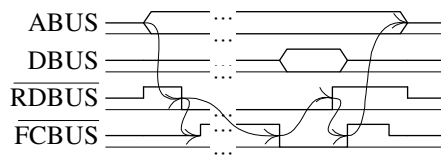
За асинхрону магистралу нацртати и објаснити временски облик свих сигнала на магистрали (адресних, управљачких и линија података) за случај:

- а) циклуса читања,
- б) циклуса уписа и
- в) циклуса прихватања броја улаза.

Сигнали на контролној магистрали су активни у логичкој нули.

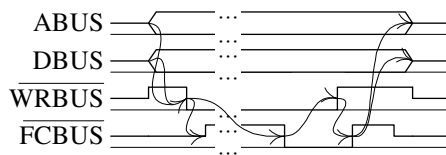
Решење:

- а) Временски облици сигнала ABUS, DBUS, \overline{RDBUS} и \overline{FCBUS} , које на магистрали размењују газда и слуга приликом реализације циклуса читања, дати су на слици.



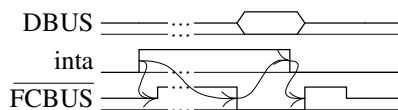
Временски облици сигнала за циклус читања на асинхроној магистрали

- б) Временски облици сигнала ABUS, DBUS, \overline{WRBUS} и \overline{FCBUS} , које на магистрали размењују газда и слуга приликом реализације циклуса уписа, дати су на слици.



Временски облици сигнала за циклус уписа на асинхроној магистрали

- в) Временски облици сигнала DBUS и \overline{FCBUS} , које на магистрали размењују процесор као газда и уређај као слуга, и сигнала потврде *inta*, који по посебној линији која не припада магистрали процесор шаље улазно/излазном уређају приликом реализације циклуса прихватања броја улаза, дати су на слици.



Временски облици сигнала за циклус прихватања броја улаза на асинхроној магистрали

Задатак 22.

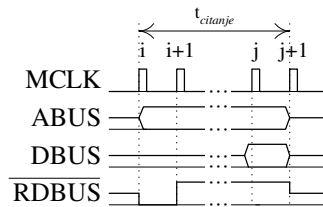
За синхрону магистралу нацртати и објаснити временски облик свих сигнала на магистрали (адресних, управљачких и линија података) за случај:

- а) циклуса читања,
- б) циклуса уписа и
- в) циклуса прихватања броја улаза.

Сигнали на контролној магистрали су активни у логичкој нули.

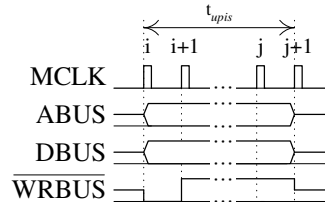
Решење

- а) Временски облици сигнала ABUS, DBUS и \overline{RDBUS} , које на магистрали размењују газда и слуга приликом реализације циклуса читања, дати су на слици.



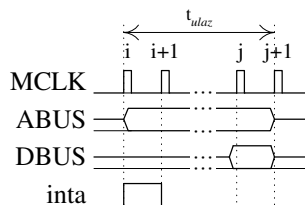
Временски облици сигнала за циклус читања на синхроној магистрали

б) Временски облици сигнала ABUS, DBUS и WRBUS, које на магистрали размењују газда и слуга приликом реализације циклуса уписа, дати су на слици.



Временски облици сигнала за циклус уписа на синхроној магистрали

в) Временски облици сигнала DBUS које на магистрали размењују процесор као газда и уређај као слуга и сигнала потврде inta, који по посебној линији која не припада магистрали, процесор шаље улазно/излазном уређају приликом реализације циклуса прихватање броја улаза, дати су на слици.



Временски облици сигнала за циклус прихватање броја улаза на синхроној магистрали.

Задатак 23.

Једноадресни процесор, меморија и периферија повезани су 16-битном адресном и 8-битном магистралом података. Садржај дела оперативне меморије дат је на слици.

Адреса	0000h	0001h	0002h	0003h	0004h	0005h	0006h	0007h	0008h	0009h	000Ah
Садржај	14h	0Ah	05h	00h	32h	15h	31h	FFh	0Ah	00h	16h
Адреса	000Bh	000Ch	000Dh	000Eh							
Садржај	01h	F0h	00h	10h							

Процесор поседује 16-битне регистре PC (програмски бројач), SP (показивач на прву слободну локацију стека који расте према нижим адресама), A (акумулатор) и AR (адресни регистар). При позиву потпрограма на стеку се чува само PC. Први бајт инструкције увек садржи само код операције, а други начин адресирања. Главни програм и потпрограм који се извршава позивом JSR дати су на слици. Претпоставити да је пре почетка извршавања главног програма SP = DEE0h и AR = 9, а да се виши бајт 16-битне речи смешта на нижу адресу.

```

адреса   инструкција   коментар
0000h   LOADW PC(05h)      ; relativno adresiranje sa 8-bitnim pomerajem
0003h   ADDB (AR) +        ; autoinkrement adresiranje
0005h   JSR (AR)           ; skok u potprogram, registarsko indirektno
0007h   HALT              ; zaustavljanje procesora

; potporogram
X       OUTB F000h    ; memorijsko direktno
X+4    RTS           ; povratak iz potprograma
    
```

а) Чему је једнако X?

б) Навести секвенцу садржаја на адресној магистралу, магистралу података и контролној магистралу за сваки циклус на магистралу при извршавању програма приказаног на слици.

Решење

а) Потпрограм почиње на адреси која представља садржај адресног регистра AR при извршавању инструкције JSR. Претходна инструкција је инкрементирала овај регистар, док је AR на почетку имао вредност 9. Према томе, $X = 10 = 000Ah$.

б) Секвенца адреса које се генеришу на адресној магистралу при извршавању датог дела програма је дата у табели.

Г.	Адресна маг.	Маг. података	R	W	M/ \overline{IO}	Коментар
1	0000h	14h	1	0	1	Фаза читања инструкције LOADW PC(5), први бајт
2	0001h	0Ah	1	0	1	Фаза читања инструкције LOADW PC(5), други бајт
3	0002h	05h	1	0	1	Фаза читања инструкције LOADW PC(5), трећи бајт
4	0008h	0Ah	1	0	1	Фаза читања операнда LOADW PC(5), виши бајт
5	0009h	00h	1	0	1	Фаза читања операнда LOADW PC(5), нижи бајт ACC=0A00h
6	0003h	00h	1	0	1	Фаза читања инструкције ADDB(AR)+, први бајт
7	0004h	32h	1	0	1	Фаза читања инструкције ADDB(AR)+, други бајт
8	0009h	00h	1	0	1	Фаза читања операнда ADDB(AR)+
9	0005h	15h	1	0	1	Фаза читања инструкције JSR(AR), први бајт
10	0006h	31h	1	0	1	Фаза читања инструкције JSR(AR), други бајт
11	DEE0h	07h	0	1	1	Фаза извршавања инструкције JSR(AR), млађи бајт PC на стек
12	DEDFh	00h	0	1	1	Фаза извршавања инструкције JSR(AR), старији бајт PC на стек
13	000Ah	16h	1	0	1	Фаза читања инструкције OUTB F000h, први бајт
14	000Bh	01h	1	0	1	Фаза читања инструкције OUTB F000h, други бајт
15	000Ch	F0h	1	0	1	Фаза читања инструкције OUTB F000h, трећи бајт
16	000Dh	00h	1	0	1	Фаза читања инструкције OUTB F000h, четврти бајт
17	F000h	00h	0	1	0	Фаза извршавања инструкције OUTB F000h
18	000Eh	10h	1	0	1	Фаза читања инструкције RTS
19	DEDFh	00h	1	0	1	Фаза извршавања инструкције RTS, млађи бајт PC са стека
20	DEE0h	07h	1	0	1	Фаза извршавања инструкције RTS, старији бајт PC са стека
21	0007h	FFh	1	0	1	Фаза читања инструкције HALT

Задатак 24.

Посматра се двоадресни процесор који је повезан са меморијом и улазно/излазним уређајима преко асинхроне магистрале. Адресни простор је капацитета 32МВ, а адресибилна јединица је 16-битна реч (W). Улазно/излазни адресни простор је меморијски раздвојен. Садржај дела оперативне меморије дат је на слици.

Адреса	0000h	0001h	0002h	0003h	0004h	0005h	0006h	0007h	0008h	0009h	000Ah
Садржај	A000h	0000h	0000h	E004h	F000h	E028h	0000h	0009h	0500h	0050h	0000h

Навести секвенцу садржаја на адресној магистралаи, магистралаи података и контролној магистралаи за сваки циклус на магистралаи при извршавању програма приказаног на слици. Регистри опште намене су дужине 16 бита.

адреса	инструкција	коментар
0000h	LOAD R1,0	; меморијско директно адресирање
0003h	OUT PC(F000h),R1	; relativно адресирање са 16-bitnim pomerajem
0005h	STORE (9h),R1	; меморијско индиректно адресирање.

Решење

Секвенца адреса које се генеришу на адресној магистралаи при извршавању датог дела програма је дата на слици.

T.	Адресна маг.	Маг. података	R	W	M/ $\bar{I}\bar{O}$	Коментар
1	00 0000h	A000h	1	0	1	Фаза читања инструкције LOAD R1,0, прва реч
2	00 0001h	0000h	1	0	1	Фаза читања инструкције LOAD R1,0, друга реч
3	00 0002h	0000h	1	0	1	Фаза читања инструкције LOAD R1,0, трећа реч
4	00 0000h	A000h	1	0	1	Фаза читања операнда LOAD R1,0
5	00 0003h	E004h	1	0	1	Фаза читања инструкције OUT PC(F000h),R1, прва реч
6	00 0004h	F000h	1	0	1	Фаза читања инструкције OUT PC(F000h),R1, друга реч
7	FF F005h	A000h	0	1	0	Фаза извршавања инструкције OUT PC(F000h),R1
8	00 0005h	E028h	1	0	1	Фаза читања инструкције STORE (9h),R1, прва реч
9	00 0006h	0000h	1	0	1	Фаза читања инструкције STORE (9h),R1, друга реч
10	00 0007h	0009h	1	0	1	Фаза читања инструкције STORE (9h),R1, трећа реч
11	00 0009h	0050h	1	0	1	Фаза дохватања адресе операнда STORE (9h),R1, старија реч
12	00 000Ah	0000h	1	0	1	Фаза дохватања адресе операнда STORE (9h),R1, млађа реч
13	50 0000h	A000h	0	1	1	Фаза извршавања инструкције STORE (9h),R1

Задатак 25.

Двоадресни процесор са 4 16-битна регистра опште намене (R0-R3) повезан је са меморијом преко асинхроне магистрале са 16 адресних линија, 8 линија за податке и одговарајућим управљачким линијама. Адресибилна јединица је бајт. Садржај дела оперативне меморије дат је на слици.

Адреса	0000h	0001h	0002h	0003h	0004h	0005h	0006h	0007h		
Садржај	00h	30h	30h	0Eh	0Eh	30h	30h	00h		
Адреса	3000h	3001h	3002h	3003h	3004h	3005h	3006h	3007h	3008h	3009h
Садржај	05h	11h	0Ah	30h	05h	22h	0Ch	44h	02h	FFh
Адреса	300Ah	300Bh	300Ch	300Dh	300Eh	300Fh	3010h	3011h	3012h	3013h
Садржај	00h	00h	05h	30h	00h	02h	04h	F0h	12h	C0h

При позиву потпрограма или прекидне рутине на стеку се чува само РС. Стек расте ка вишим меморијским адресама, а SP указује на прву слободну локацију на врху стека. Претпоставити да је пре почетка извршавања сегмента програма са слике, SP=4000h, IVTP=0h, PC=3000h. Навести секвенцу садржаја на адресној магистралаи, магистралаи података и контролној магистралаи за сваки циклус на магистралаи при извршавању тог програма.

адреса	инструкција	коментар
3000h	LOAD R1, (300Ah)	; memorijski indirektno adresiranje
3004h	LOAD R2, (R1)0Ch	; registarski indirektno sa 8-bitnim pomerajem
3007h	INT #2	; softverski prekid
3009h	HALT	; zaustavljanje procesora
...		
300Eh	ADD R2, #4	; neposredno adresiranje
3011h	PUSH R2	; registarski direktno adresiranje
3013h	RTI	; povratak iz prekidne rutine

Решење

T.	Адресна маг.	Маг. података	R	W	M/IO	Коментар
1	3000h	05h	1	0	1	Фаза читања инструкције LOAD R1, (300Ah), први бајт
2	3001h	11h	1	0	1	Фаза читања инструкције LOAD R1, (300Ah), други бајт
3	3002h	0Ah	1	0	1	Фаза читања инструкције LOAD R1, (300Ah), трећи бајт
4	3003h	30h	1	0	1	Фаза читања инструкције LOAD R1, (300Ah), четврти бајт
5	300Ah	00h	1	0	1	Фаза читања адресе операнда LOAD R1, (300Ah), први бајт
6	300Bh	00h	1	0	1	Фаза читања адресе операнда LOAD R1, (300Ah), други бајт
7	0000h	00h	1	0	1	Фаза читања операнда LOAD R1, (300Ah), млађи бајт
8	0001h	30h	1	0	1	Фаза читања операнда LOAD R1, (300Ah), старији бајт R1=3000h
9	3004h	05h	1	0	1	Фаза читања инструкције LOAD R2, (R1)0Ch, први бајт
10	3005h	22h	1	0	1	Фаза читања инструкције LOAD R2, (R1)0Ch, други бајт
11	3006h	0Ch	1	0	1	Фаза читања инструкције LOAD R2, (R1)0Ch, трећи бајт
12	300Ch	05h	1	0	1	Фаза читања операнда LOAD R2, (R1)0Ch, млађи бајт
13	300Dh	30h	1	0	1	Фаза читања операнда LOAD R2, (R1)0Ch, старији бајт R2=3005h
14	3007h	44h	1	0	1	Фаза читања инструкције INT #2, први бајт
15	3008h	02h	1	0	1	Фаза читања инструкције INT #2, други бајт
16	4000h	09h	0	1	1	Опслуживање инструкције прекида, млађи бајт РС-а на стек
17	4001h	30h	0	1	1	Опслуживање инструкције прекида, млађи бајт РС-а на стек
18	0004h	0Eh	1	0	1	Читање млађег бајта адресе почетка прекидне рутине
19	0005h	30h	1	0	1	Читање старијег бајта адресе почетка прекидне рутине

20	300Eh	00h	1	0	1	Фаза читања инструкције ADD R2,#4, први бајт
21	300Fh	02h	1	0	1	Фаза читања инструкције ADD R2,#4, други бајт
22	3010h	04h	1	0	1	Фаза читања инструкције ADD R2,#4, трећи бајт R2=3009h
23	3011h	F0h	1	0	1	Фаза читања инструкције PUSH R2, први бајт
24	3012h	12h	1	0	1	Фаза читања инструкције PUSH R2, други бајт
25	4002h	09h	0	1	1	Фаза извршавања инструкције PUSH R2, млађи бајт R2 на стек
26	4003h	30h	0	1	1	Фаза извршавања инструкције PUSH R2, старији бајт R2 на стек
27	3013h	C0h	1	0	1	Фаза читања инструкције RTI
28	4003h	30h	1	0	1	Фаза извршавања инструкције RTI, старији бајт PC са стека
29	4002h	09h	1	0	1	Фаза извршавања инструкције RTI, млађи бајт PC са стека
30	3009h	FFh	1	0	1	Фаза читања инструкције HALT

УЛАЗ/ИЗЛАЗ

Задатак 26.

Адресне линије и линије података магистрале посматраног рачунара су широке по 16 бита. Процесор је једноадресни и оперише само над 16 битним подацима. Улазно излазни и меморијски адресни простори су раздвојени. Периферије PER0 и PER1 имају управљачке, статусне и регистре података редом на следећим адресама: FF00h, FF01h, FF02h (PER0) и FF10h, FF11h, FF12h (PER1). У управљачким регистрима најмлађи бит је бит Enable Interrupt којим се дозвољава прекид, најстарији је бит Start којим се дозвољава почетак операције, а бит 1 је бит смера операције (0—улаз, 1—излаз). У статусним регистрима најмлађи бит је бит спремности Ready. Написати програм којим се блок од 200h података учитава са PER0, смешта у меморију од локације F000h, обрађује процедуром Obrada, и резултат шаље на PER1. Операције улаза и излаза реализовати испитивањем бита спремности. Процедура Obrada не мења место ни дужину блока података. Ову процедуру не треба реализовати, већ је само позвати на одговарајућем месту у програму помоћу наредбе CALL Obrada.

Решење:

```

LOAD #200h           ;brojač u MemCnt
STORE MemCnt
LOAD #F000h         ;adresa u MemDst
STORE MemDst
LOAD #8000h         ;Start=1, Enable=0, Direction=0
OUT FF00h           ;pokreni kontroler
LOOP1:IN FF01h      ;ispitivanje bita spremnosti
AND #1
JZ LOOP1           ;ako nije spreman, čekaj
IN FF02h           ;ulaz podatka
STORE (MemDst)     ;i smeštanje u memoriju
LOAD MemDst        ;ažuriranje pokazivača
INC
STORE MemDst
LOAD MemCnt        ;i brojača
DEC
STORE MemCnt
JNZ LOOP1         ;ako nije poslednji, ponovi
LOAD #0           ;zaustavi kontroler
OUT FF00h         ;Stop PER0
CALL Obrada       ;obrada
LOAD #200h        ;brojač u MemCnt
STORE MemCnt
LOAD #F000h       ;pokazivač u MemSrc
STORE MemSrc
LOAD #8002h       ;Start=1, Enable=0, Direction=1
OUT FF10h         ;pokretanje kontrolera
LOOP2:IN FF11h    ;ispitivanje bita spremnosti
AND #1
JZ LOOP2         ;ako nije spreman, čekaj
LOAD (MemSrc)    ;podatak na izlaz
OUT FF12h
LOAD MemSrc      ;ažuriranje pokazivača
INC
STORE MemSrc
LOAD MemCnt      ;i brojača
DEC
STORE MemCnt
JNZ LOOP2       ;ako nije poslednji, ponovi
LOAD #0         ;zaustavi kontroler
OUT FF10h       ;Stop PER1

```

Задатак 27.

Једноадресни рачунар са 16 битном адресном и 8 битном магистралом података поседује меморијски пресликан улазно/излазни адресни простор. Периферије PER0, PER1 и PER2 имају управљачке, статусне и регистре

података редом на следећим адресама: FF10h, FF11h, FF12h (PER0), FF20h, FF21h, FF22h (PER1) и FF30h, FF31h, FF32h (PER2). У управљачким регистрима најстарији бит је Start којим се дозвољава почетак операције, најмлађи је бит Enable којим се дозвољава прекид, а бит 3 је бит смера операције (1—улаз, 0—излаз). У статусним регистрима најстарији бит је бит спремности Ready. Написати главни програм и одговарајуће прекидне рутине којима се: истовремено са периферија PER0 и PER1 учитавају блокови од по 100h бајтова и смештају у меморију почев од локација 1000h (PER0) и 1100h (PER1), затим изврши обрада унетих блокова података позивом процедуре Obrada (CALL Obrada) и потом врши пренос обрађеног блока од 200h бајтова, почев од адресе 1000h из меморије у PER2. Обе улазне операције извршити коришћењем прекида, а излазну операцију испитивањем бита Ready.

Решење

```

LOAD #1000h ;pokretanje obe ulazne operacije
STORE MemB0
LOAD #1100h
STORE MemB1
LOAD #100h
STORE MemCnt0
STORE MemCnt1
LOAD #0
STORE MemSem0
STORE MemSem1
LOADB #89h
STOREB FF10h
STOREB FF20h
Wait0: LOAD MemSem0;čekanje na završetak sa PER0
AND #1
JZ Wait0
Wait1: LOAD MemSem1;čekanje na završetak sa PER1
AND #1
JZ Wait1
CALL Obrada ;obrada
LOAD #1000h ;izlazna operacija
STORE MemB
LOAD #200h
STORE MemCnt
LOADB #80h
STOREB FF30h
Loop: LOADB FF31h
AND #80h
JZ Loop
LOADB (MemB)
STOREB FF32h

INC MemB
DEC MemCnt
JNZ Loop
LOAD #0
STOREB FF30h
Прекидне рутине:
Per0: PUSHA
LOADB FF12h
STOREB (MemB0)
INC MemB0
DEC MemCnt0
JNZ Back0
LOAD #0
STOREB FF10h
INCA
STORE MemSem0
Back0: POPA
RTI
Per1: PUSHA
LOADB FF22h
STOREB (MemB1)
INC MemB1
DEC MemCnt1
JNZ Back1
LOAD #0
STOREB FF20h
INCA
STORE MemSem1
Back1: POPA
RTI

```

Задатак 28.

Једноадресни процесор са раздвојеним адресним просторима, меморија, периферије PER0 и PER1 повезани су системском магистралом са 16 адресних линија и 16 линија за податке. Адресирање је на нивоу 16-битних речи. Механизам прекида је векторисан а број улаза у IV табелу је одређен фиксно (0 за PER0, 1 за PER1). Адресе релевантних регистара су:

```

PER0_CONTROL FF00h PER1_CONTROL FF10h
PER0_STATUS FF01h PER1_STATUS FF11h
PER0_DATA FF02h PER1_DATA FF12h

```

У управљачким регистрима бит 0 је Start којим се дозвољава почетак операције, бит 1 одређује смер операције (0-улаз, 1-излаз), бит 4 је Enable којим се дозвољава прекид, а у статусним регистрима бит 0 је Ready који сигнализира спремност контролера. Написати главни програм и одговарајућу прекидну рутину којима се: упоредо врши учитавање низа A(i) (i=0...FF) са PER0 у меморијски блок који почиње од адресе 1000h, и низа B(i) (i=0...FF) са PER1 у меморијски блок почев од адресе 1100h, затим изврши сабирање унетих низова (B(i) = A(i) + B(i)) и резултујући низ шаље на периферију PER0. Улаз са PER0 реализовати коришћењем механизма прекида, улаз са PER1 реализовати испитивањем бита спремности, а излаз на PER0 коришћењем механизма прекида. Контролни регистар PER0 може да се чита.

Решење

Главни програм:	DEC MemCnt
	JNZ Loop
LOAD #100h	LOAD #100h
STORE MemCnt0	STORE MemCnt0
STORE MemCnt1	LOAD #1100h
LOAD #1000h	STORE MemAdr0
STORE MemAdr0	LOAD #0
LOAD #1100h	STORE MemSem0
STORE MemAdr1	LOAD #1 ; smer: izlaz
LOAD #0	STORE MemDir0
STORE MemSem0	LOAD #13h
LOAD #0 ; smer: ulaz	OUT FF00h
STORE MemDir0	...
LOAD #11h	Wait1:LOAD MemSem0
OUT FF00h	CMP #1
LOAD #01h	JNZ Wait1
OUT FF10h	HALT
Crdy: IN FF11h	
AND #01h	Прекидна рутина:
JZ Crdy	PUSHA
IN FF12h	LOAD MemDir0 ; koji smer?
STORE (MemAdr1)	CMP #1
INC MemAdr1	JZ Out
DEC MemCnt1	; input
JNZ Crdy	IN FF02h
LOAD #0	STORE (MemAdr0)
OUT FF10h	Inc: INC MemAdr0
Wait: LOAD MemSem0	DEC MemCnt0
CMP #1	JNZ Back
JNZ Wait	LOAD #1
	STORE MemSem0
LOAD #100h	LOAD #0
STORE MemCnt	OUT FF00h
LOAD #1000h	JMP Back
STORE MemA	;output
LOAD #1100h	Out: LOAD (MemAdr0)
STORE MemB	OUT FF02h
Loop: LOAD (MemA)	JMP Inc
ADD (MemB)	Back: POPA
STORE (MemB)	RTI
INC MemA	
INC MemB	

Задатак 29.

Једноадресни процесор са раздвојеним адресним просторима, периферије PER0 и PER1, и меморија повезани су магистралом са 16 адресних линија и 16 линија за податке. Адресирање је на нивоу 16-битних речи. Адресе релевантних регистара дате су на слици.

PER0_CONTROL	FF00h	PER1_CONTROL	FF10h
PER0_STATUS	FF01h	PER1_STATUS	FF11h
PER0_DATA	FF02h	PER1_DATA	FF12h

Бит 0 контролних регистара је Start бит, бит 1 дефинише смер операције (0—улаз, 1—излаз), а бит 2 је Enable којим се омогућује прекид. Бит 4 статусних регистара је Ready бит. Написати програм и одговарајуће прекидне рутине којима се реализује: учитавање низа $A(i)$, $i = 0, \dots, 999$ са PER0 у меморијски блок који почиње од адресе 2000h и слање квадрираног низа ($A(i)*A(i)$) на PER1. Пријем са PER0 и слање на PER1 реализовати упоредо, тј. омогућити слање елемента низа на PER1 чим је то могуће. Улаз реализовати механизмом прекида, а излаз испитивањем бита спремности.

Решење

Главни програм:
LOAD #2000h ;početna adresa ulaznog bafera

```

STORE MemWP      ;pokazivač koji prati učitavanje sa PER0
STORE MemRP      ;pokazivač koji prati slanje na PER1
LOAD #1000       ;broj elemenata niza
STORE CntW
STORE CntR
LOAD #5
OUT FF00h        ;start PER0
LOAD #3
OUT FF10h        ;start PER1
Chck: IN  FF11h
AND #10h
JZ Chck
;провера да ли постоји елемент спреман за слање
LOAD MemWP
SUB MemRP
JLE Chck         ;skok ako je rezultat oduzimanja =< 0
LOAD (MemRP)
MUL (MemRP)     ;kvadriranje elementa niza
OUT FF12h       ;slanje rezultujućeg elementa na PER1
INC MemRP
DEC CntR
JNZ Chck
LOAD #0
OUT FF10h       ;Stop PER1
HALT
Прекидна рутина периферије PER0:
Per1: PUSHA
IN  FF02h
STORE (MemWP)
INC MemWP
DEC CntW
JNZ Back
LOAD #0
OUT FF00h       ;Stop PER0
Back: POPA
RTI

```

Задатак 30.

Двоадресни процесор са 16 регистара опште намене поседује меморијски пресликан улазно/излазни адресни простор. Процесор, меморија, периферије PER0, PER1 и PER2 повезани су системском магистралом са 16 адресних линија и 16 линија за податке. Адресирање је на нивоу 16-битних речи. Адресе релевантних регистара су приказане на слици.

PER0_CONTROL	FF00h	PER1_CONTROL	FF10h	PER2_CONTROL	FF20h
PER0_STATUS	FF01h	PER1_STATUS	FF11h	PER2_STATUS	FF21h
PER0_DATA	FF02h	PER1_DATA	FF12h	PER2_DATA	FF22h

У управљачким регистрима бит 15 је Start којим се дозвољава почетак операције, бит 0 одређује смер операције (0—улаз, 1—излаз), бит 7 је Enable којим се дозвољава прекид, а у статусним регистрима бит 0 је Ready који сигнализира спремност контролера периферије. Написати главни програм и одговарајуће прекидне рутине којима се упоредо: врши читавање низа A(i) (i = 0, ..., 99) са PER0 у меморијски блок који почиње од адресе 1000, и низа B(i) (i = 0, ..., 99) са PER1 у меморијски блок почев од адресе 2000, формирање резултујућег низа C(i) ($C(i) = A(i) + B(i)$, i = 0, ..., 99) који се смешта у меморијски блок почев од адресе 3000, и слање резултујућег низа C на периферију PER2. Формирање низа C тече упоредо са читавањем, тј. чим се прочита i-ти елемент низа A(B) формира се i-ти елемент низа C под условом да је прочитан i-ти елемент низа B(A). Такође, слање низа C на PER2 треба започети пре него што је цео низ C формиран, тј. чим је неки елемент низа C формиран треба омогућити његово слање на PER2. Улаз са PER0 и PER1 реализовати коришћењем механизма прекида, а излаз на PER2 испитивањем бита спремности.

Решење:

```

MOV R0, #1
MOV mcA, R0
MOV mcB, R0
MOV mwcC, R0

```


	MOV	R1,	R0	;R1 is read pointer		MOV	(R2) 3000,	R0
	MOV	FF00h,	#8080h	;start PER0	SkipA:	CMP	R1,	#99
	MOV	FF10h,	#8080h	;start PER1		JNZ	BackA	
	MOV	FF20h,	#8001h	;start PER2		MOV	FF00h,	#0
Wait:	MOV	R0,	FF21h	;read status	BackA:	POP	R2	
	AND	R0,	#1			POP	R1	
	JZ	Wait				POP	R0	
	MOV	R2,	mwcC			RTI		
	CMP	R2,	R1		Прекидна рутина за PER1:	INTD		
	BLE	Wait				PUSH	R0	
	INC	R1				PUSH	R1	
	MOV	FF22h,	(R1)3000			PUSH	R2	
	CMP	R1,	#99			MOV	R0,	FF12h
	JNZ	Wait				MOV	R1,	mcB
	MOV	FF20h,	#0			INC	R1	
	HALT					MOV	mcB,	R1
Прекидна рутина за PER0:						MOV	(R1) 2000,	R0
	INTD					CMP	R1,	mcA
	PUSH	R0				JG	SkipB	
	PUSH	R1				ADD	R0,	(R1) 1000
	PUSH	R2				MOV	R2,	mwcC
	MOV	R0,	FF02h			INC	R2	
	MOV	R1,	mcA			MOV	mwcC,	R2
	INC	R1				MOV	(R2) 3000,	R0
	MOV	mcA,	R1		SkipB:	CMP	R1,	#99
	MOV	(R1)1000,	R0			JNZ	BackB	
	CMP	R1,	mcB			MOV	FF10h,	#0
	JG	SkipA			BackB:	POP	R2	
	ADD	R0,	(R1) 2000			POP	R1	
	MOV	R2,	mwcC			POP	R0	
	INC	R2				RTI		
	MOV	mwcC,	R2					

Задатак 31.

Двоадресни процесор са раздвојеним I/O и меморијско адресним просторима везан је на 16-битну адресу и 8-битну магистралу података. На магистралу су везана два контролера периферије, чији се регистри налазе на следећим адресама: Data (0026h, 0028h), Control (0030h, 0032h) и Status (0020h, 0022h). Сви регистри су 8-битни, адресирање је бајтовско. У статусним регистрима бит 2 је Ready, а у управљачким регистрима бит 0 је Start. Написати програм који учитава вредности са обе периферије, техником испитивања бита Ready, и пристигле вредности сумира, посебно за сваку периферију у локацијама SUM1 и SUM2. Улаз се прекида када са било које периферије стигне вредност 0. Уређаји шаљу податке различитим брзинама, а податак који је пристигао треба учитати што пре.

Решење

```
...
MOV SUM1, #0
MOV SUM2, #0
OUT 30h, #1 ; Start PER1
OUT 32h, #1 ; Start PER2
LOOP: IN R0, 20h ; Test PER1
      AND R0, #4
      JNZ PER1
TSTP2: IN R0, 22h ; Test PER2
      AND R0, #4
      JZ LOOP
PER2:  IN R0, 28h ;Input from PER2
      OR R0, R0
      JZ END
      ADD SUM2, R0
      JMP LOOP
PER1:  IN R0, 26h ;Input from PER1
      OR R0, R0
      JZ END
      ADD SUM1, R0
      JMP TSTP2
END:   OUT 30h, #0; Stop PER1
      OUT 32h, #0; Stop PER2
...
```

Задатак 32.

Једноадресни процесор са меморијски мапираним улазно/излазним адресним простором, меморија, периферија PER0 (adrCR=FF10h, adrSR= FF11h, adrDR=FF12h), периферија PER1 (adrCR=FF20h, adrSR=FF21h, adrDR=FF22h) са придруженим контролором периферије DMA1 (adrCR=FF00h, adrSR=FF01h, adrDR=FF02h, adrCNT=FF03h, adrAs=FF04h, adrAd=FF05h) и периферија PER2 (adrCR=FF30h, adrSR=FF31h, adrDR=FF32h) повезани су системском магистралом са 16-битном адресном и 16-битном магистралом података. Адресирање је на нивоу 16-битних речи. У управљачким регистрима бит 0 је Start којим се дозвољава почетак операције, бит 1 одређује смер операције (0-улаз, 1-излаз), бит 2 је Enable којим се дозвољава прекид, а у статусним регистрима бит 4 је Ready који сигнализира спремност контролера. Бит 3 управљачког регистра DMA контролера задаје режим рада (0-блоковски (burst), 1-циклус по циклус (cycle stealing)).

а) Написати главни програм и одговарајуће прекидне рутине којима се обавља следећи пренос. Са периферије PER1 учитава се низ података A(i) дужине 80h и смешта у меморију почев од адресе 1000h коришћењем DMA контролера у burst режиму рада, па се по завршетку преноса учита низ података B(i) исте дужине са периферије PER1 и смешта у меморију почев од адресе 2000 коришћењем DMA контролера у циклус по циклус режиму рада. По завршетку уноса низова врши се упоредо слање података на периферије PER0 и PER2, и то тако што се на бржу периферију шаље податак A(i)/B(i), а на спорију A(i)-B(i) (i=1, ..., 80h). Излаз на периферију PER0 реализовати коришћењем механизма прекида, а излаз на периферију PER2 испитивањем бита спремности.

б) Да ли процесор може приступити регистру података периферије PER1 током учитавања низа података A(i) са ове периферије? Образложити одговор.

Решење:

а) Главни програм:

	LOAD #0h	STORE FF10h
	STORE MemSem	ChRd: LOAD FF31h
	STORE MemSem0	AND #10h
	STORE MC0	JZ ChRd
	STORE MC2	INTD
	LOAD #80h	LOAD MC2
	STORE FF03h	SUB MC0
	LOAD #1000h	JL Skip
	STORE FF05h	LOAD (MAA2)
	STORE MAA0	DIV (MAB2)
	STORE MAA2	JMP IncMa
	LOAD #1h	Skip: LOAD (MAA2)
	STORE FF20h	SUB (MAB2)
	LOAD #5h	IncMa: STORE FF32h
	STORE FF00h	INC MC2
Wait:	LOAD MemSem	INTE
	CMP #1h	INC MAA2
	JNZ Wait	INC MAB2
	LOAD #0h	LOAD MC2
	STORE MemSem	CMP #80h
	LOAD #80h	JNZ ChRd
	STORE FF03h	LOAD #0h
	LOAD #2000h	STORE FF30h
	STORE FF05h	...
	STORE MAB0	ChPer0: LOAD MemSem0
	STORE MAB2	CMP #1h
	LOAD #1h	JZ ChPer0
	STORE FF20h	HALT
	LOAD #Dh	DMA_Int: PUSHA
	STORE FF00h	LOAD #0h
Wait2:	LOAD MemSem	STORE FF20h
	CMP #1h	STORE FF00h
	JNZ Wait2	INC
	LOAD #3h	STORE MemSem
	STORE FF30h	POPA
	LOAD #7h	RTI
		PER0_Int:PUSHA

LOAD MC0	INC MAB0
SUB MC2	LOAD MC0
JL Skip0	CMP #80h
LOAD (MAA0)	JNZ Back
DIV (MAB0)	LOAD #0h
JMP IncMa0	STORE FF10h
Skip0: LOAD (MAA0)	INCA
SUB (MAB0)	STORE MemSem0
IncMa0: STORE FF12h	Back: POPA
INC MC0	RTI
INC MAA0	

б) HE. Процесор не може извршити циклус читања на магистралаи јер DMA контролер ради у блоковском режиму и држи магистралу заузету до завршетка преноса целог блока података

Задатак 33.

Једноадресни процесор са развојеним адресним просторима, меморија, периферија PER0, периферије PER1 и PER2 са придруженим контролерима за директан приступ меморији DMA1 и DMA2, редом, повезани су системском магистралом са 16-битном адресном и 16-битном магистралом података. Адресирање је на нивоу 16-битних речи. Адресе релевантних регистара су:

PER0_CONTROL	FF10h	PER1_DATA	FF22h
PER0_STATUS	FF11h	PER2_CONTROL	FF30h
PER0_DATA	FF12h	PER2_STATUS	FF31h
PER1_CONTROL	FF20h	PER2_DATA	FF32h
PER1_STATUS	FF21h		
DMA1_CONTROL	FF00h	DMA2_CONTROL	FF06h
DMA1_ADDRESS	FF01h	DMA2_ADDRESS	FF07h
DMA1_COUNT	FF02h	DMA2_COUNT	FF08h
DMA1_DATA	FF03h	DMA2_DATA	FF09h
DMA1_STATUS	FF04h	DMA2_STATUS	FF0Ah

У управљачким регистрима бит 0 је Start којим се дозвољава почетак операције, бит 1 одређује смер операције (0-улаз, 1-излаз), бит 2 је Enable којим се дозвољава прекид, а у статусним регистрима бит 4 је Ready који сигнализира спремност контролера. Бит 3 управљачког регистра DMA контролера задаје режим рада (0-блоковски (burst), 1-циклус по циклус (cycle stealing)). Написати главни програм и одговарајуће прекидне рутине којима се обавља следећи пренос. Са периферије PER0 се прихвата бесконачни низ података и упоредо прослеђује периферијама PER1 и PER2. На располагању су два бафера BP0 и BP1 који се пуне наизменично подацима са периферије PER0. Наиме, упоредо са пуњењем бафера BP0 одвија се пражњење бафера BP1 слањем података на периферије PER1 и PER2 и обратно, док се пуни бафер BP1 празни се бафер BP0 слањем података на периферије PER1 и PER2. Величине бафера су 100h речи а почетне адресе бафера BP0 и BP1 су 1000h и 1100h, редом. Улаз са периферије PER0 реализовати коришћењем механизма прекида, излаз на периферију PER1 коришћењем DMA контролера који ради у циклус-по-циклус режиму, а излаз на периферију PER2 коришћењем DMA контролера који ради у блоковском режиму.

Решење

; initialize input from PER0 (fill BP0)

LOAD #0	STORE Ain
STORE Mfin	LOAD #1
LOAD #1000h	STORE Dir
STORE Ain	Init: LOAD #0
LOAD #100h	STORE Mfin
STORE Cntin	STORE MFout1
LOAD #5	STORE MFout2
OUT FF10h	LOAD #100h
; ...	STORE Cntin
Wait1: LOAD Mfin	; init DMAs
AND #1	LOAD Aout
JZ Wait1	OUT FF01h
; initialize output from BP0 and input to BP1	OUT FF07h
Swap: LOAD #1000h	LOAD #100h
STORE Aout	OUT FF02h
LOAD #1100h	OUT FF08h

; start controllers

```

LOAD #5
OUT FF10h
LOAD #Fh
OUT FF00h
LOAD #7
OUT FF06h
LOAD #3h
OUT FF20h
LOAD #3h
OUT FF30h

```

;

```

Wait2: LOAD Mfin
      AND MFout1
      AND MFout2
      JZ Wait2
      LOAD Dir
      AND #1
      JZ Swap

```

; initialize output from BP1 and input to BP0

```

LOAD #1100h
STORE Aout
LOAD #1000h
STORE Ain
LOAD #0
STORE Dir
JMP Init

```

Прекидна рутине:

DMA1Int:PUSHA

```

LOAD #0
OUT FF20h
OUT FF00h
INC
STORE MFout1
POPA
RTI

```

DMA2Int:PUSHA

```

LOAD #0
OUT FF30h
OUT FF06hh
INC
STORE MFout2
POPA
RTI

```

Per0Int: PUSHA

```

IN FF12h
STORE (Ain)
INC Ain
DEC Cntin
JNZ Back0
LOAD #0
OUT FF10h ;stop PER0
INC
STORE MFin

```

Back0: POPA
RTI

КЕШ МЕМОРИЈА

Задатак 34.

Оперативна меморија рачунара је капацитета 1 МВ, а ширина речи износи 1 бајт.

а) Нацртати структуру кеш меморије, означити ширину у битовима свих релевантних делова и укратко описати њихову намену за кеш меморију са 1024 блока, блоком величине четири бајта и асоцијативним пресликавањем претпостављајући да је ширина речи дела кеш меморије у коме се чува садржај један бајт.

б) Садржај релевантних локација оперативне меморије је дат на слици. Четири захтева за операцијама читања из оперативне меморије са локација 1402h, 401h, 2000h и 1003h се изводе у датом редоследу. За сваки захтев у посебној врсти табеле попунити колоне означене са: Адреса – генерисана адреса, Тип - тип приступа (Rd – читање, Wr – упис), Tag - вредност поља Tag генерисане адресе, Word - вредност поља Word генерисане адресе, Време - време потребно да се обављање приступа и Адресе - адресе локација оперативне меморије којима је приступа у току извршења датог захтева. Навести вредности свих релевантних делова кеш меморије после ове четири операције, претпостављајући да је кеш меморија на почетку била празна.

Адреса	400	401	402	403	...	1000	1001	1002	1003	...	1400	1401	1402	1403	...	2000	2001	2002	2003	
Садржај	0	1	0	0	...	0	0	0	2	...	0	0	3	0	...	4	0	0	0	

Садржај релевантних локација оперативне меморије.

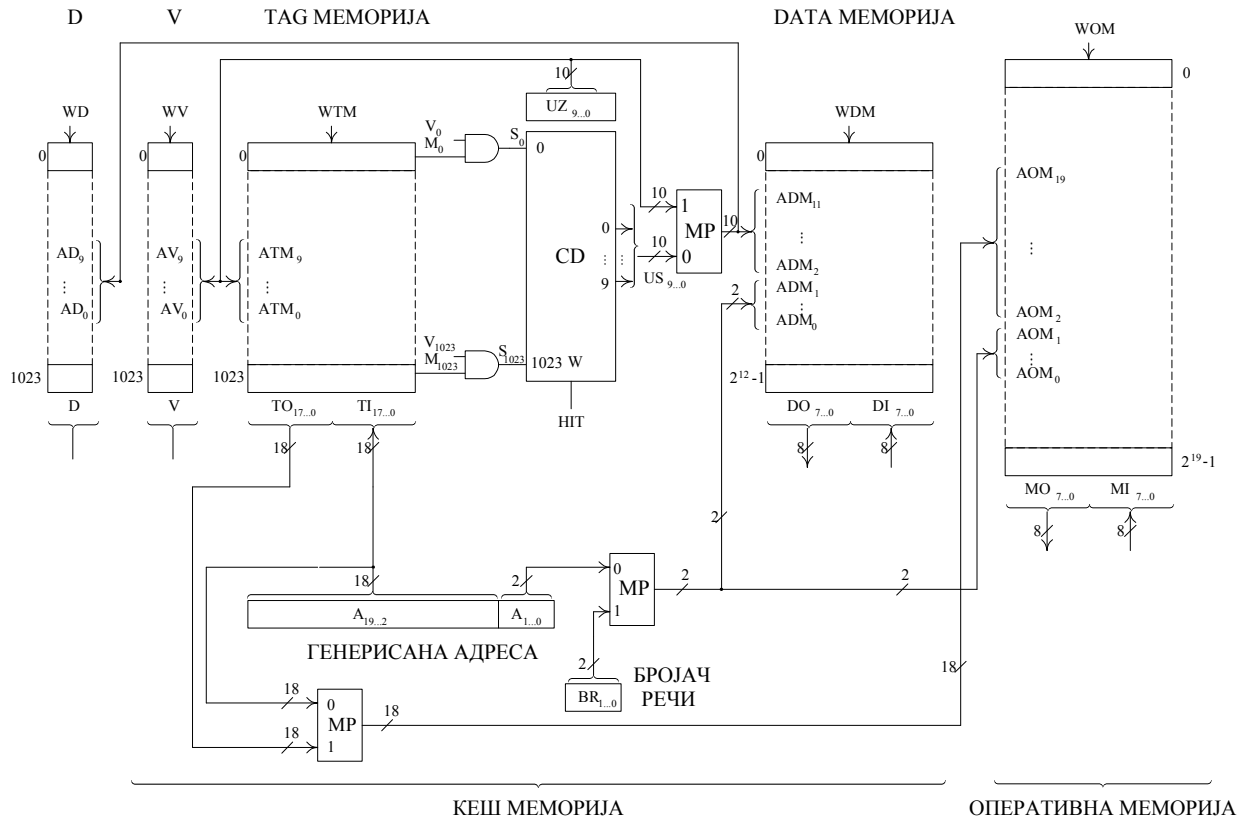
в) Израчунати укупно време потребно за приступ кеш меморији за податке у датој секвенци. Треба претпоставити да се прво пренесе цео блок из оперативне меморије у кеш меморије и обрнуто, па се тек онда приступа локацији, и да се све операције раде секвенцијално. Приликом израчунавања времена потребног да се добије садржај узети у обзир само време утврђивања сагласности у TAG MEMORIJI (t_{SA}), време приступа оперативној меморији (t_{OM}), време приступа DATA MEMORIJI (t_{DM}) и време приступа TAG MEMORIJI (t_{TM}), занемарити времена потребна за остале активности.

Решење:

Структура адреса код кеш меморије:

Број блока	Word (1 бита)
Tag (m бита)	

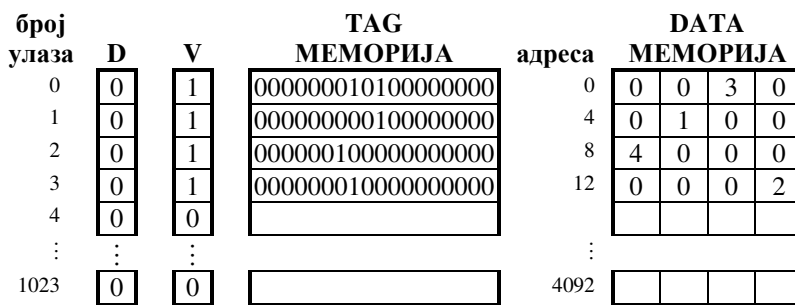
а) Структура кеш меморије приказана је на слици.



Структура кеш меморије са асоцијативним пресликавањем

б) Генерисане адресе су: 1402h, 401h, 2000h и 1003h. Вредности свих релевантних делова кеш меморије после четири операције читања приказани су на слици.

Адреса	Тип	Tag	Word	Време	Адресе
1402h	Rd	000000010100000000	10	$t_{SA}+4 \cdot (t_{OM}+t_{DM})+t_{TM}+t_{SA}+t_{DM}$	1400h-1403h
401h	Rd	000000000100000000	01	$t_{SA}+4 \cdot (t_{OM}+t_{DM})+t_{TM}+t_{SA}+t_{DM}$	400h-403h
2000h	Rd	000000100000000000	00	$t_{SA}+4 \cdot (t_{OM}+t_{DM})+t_{TM}+t_{SA}+t_{DM}$	2000h-2003h
1003h	Rd	000000010000000000	11	$t_{SA}+4 \cdot (t_{OM}+t_{DM})+t_{TM}+t_{SA}+t_{DM}$	1000h-1003h



Садржај релевантних делова кеш меморије

в) Време дохватања блока је:

$$t_B = 4 \cdot (t_{OM} + t_{DM})$$

Укупно време је:

$$t = 4 \cdot (t_{SA} + 4 \cdot (t_{OM} + t_{DM}) + t_{TM} + t_{SA} + t_{DM}) = 20 \cdot t_{DM} + 16 \cdot t_{OM} + 8 \cdot t_{SA} + 4 \cdot t_{TM}$$

ВИРТУЕЛНА МЕМОРИЈА

Задатак 35.

Посматра се рачунар са страничном организацијом виртуелне меморије. Виртуелни адресни простор корисника је 16 МВ и подељен је на странице величине 1 КВ. Реални адресни простор је 1 МВ и подељен је на блокове величине 1 КВ. Виртуелном и реалном адресом се адресирају речи дужине 1 бајт.

Урадити следеће:

- Означити дужине у битовима свих делова виртуелне и реалне адресе.
- Нацртати табелу страница, објаснити како се приступа табелама страница различитих корисника и дати функцију појединих поља дескриптора странице.
- Објаснити како се врши пресликавање виртуелне у реалну адресу за случај да је страница у меморији и навести шта се од тога ради хардверски а шта софтверски.
- Објаснити шта се ради за случај да страница није у меморији и навести шта се од тога ради хардверски а шта софтверски.

Решење:

а) Структура виртуелне и реалне адресе и дужине у битовима делова виртуелне и реалне адресе су приказани на слици.

