

ПРОЈЕКАТ 2

АРХИТЕКТУРА И ОРГАНИЗАЦИЈА РАЧУНАРА 2

ШКОЛСКА 2015/2016

Пројекат 2 из предмета Архитектура и организација рачунара 2 се школске 2015/2016 ради у групама од до 4 студената, а брани се самостално. Пројекат носи до 20.

Посматра се део рачунара који чине меморија, процесор, магистрала и периферије.

Процесор је са раздвојеним меморијским и улазно/излазним адресним простором. Меморија је капацитета 2^{32} бајтова. Ширина меморијске речи је 4 бајта.

Процесор је са двоадресним форматом инструкција. Подаци су целобројне величине са знаком дужине 4 бајта. Адресе и подаци у меморији заузимају једну меморијску локацију. У процесору постоји 32 регистара опште намене R0 до R31 дужине 32 бита, програмски бројач PC дужине 32 бита, указивач на врх стека SP дужине 32 бита, и програмска статусна реч PSW дужине 32 бита.

Стек расте од виших ка нижим адресама, а SP показује на заузету локацију на врху стека.

Све инструкције су дужине 4 бајта. Битови 31 до 26 инструкције специфицирају код операције, док преостали бити инструкције специфицирају коришћене регистре и начине адресирања.

Аритметичке инструкције процесора су сабирање (ADD), одузимање (SUB) и поређење (CMP), и за њих су усвојени кодови операција 0, 1 и 2. Први, изворишни и у исто време одредишни, операнд је дат регистром опште намене R0 до R31 а специфициран је битовима 25 до 21 инструкције. Други изворишни операнд је дат садржајем једног од регистара опште намене R0 до R31 а специфициран је битовима 20 до 16 инструкције. Најнижих 16 бита инструкције представљају непосредну величину са знаком дату у другом комплементу коју треба третирати као трећи операнд (reg1 operation reg2 operation imm16). Инструкције постављају аритметичке индикаторе N, Z, V и C.

Логичке инструкције процесора су логички производ (AND), логичка сума (OR), комплементирање бит по бит (NOT), провера логичког производа (TST), и за њих су усвојени кодови операција 8, 9, 10 и 11. Први, изворишни и у исто време одредишни, операнд је дат регистром опште намене R0 до R31 а специфициран је битовима 25 до 21 инструкције. Други изворишни операнд је дат садржајем једног од регистара опште намене R0 до R31 а специфициран је битовима 20 до 16 инструкције. Најнижих 16 бита инструкције се не користе. Инструкције постављају аритметичке индикаторе N и Z.

Померачке инструкције процесора су логичко померање у десно (SHR) и логичко померање улево (SHL), и за њих су усвојени кодови операција 16 и 17. Одредишни операнд је дат регистром опште намене R0 до R31 а специфициран је битовима 25 до 21 инструкције. Изворишни операнд је дат садржајем једног од регистара опште намене R0 до R31 а специфициран је битовима 20 до 16 инструкције. Број места за колико треба померити садржај изворишног регистра је дат као непосредна величина без знака битовима 4 до 0 инструкције. Бити 16 до 5 инструкције се не користе. Инструкције постављају аритметичке индикаторе N, Z и C.

Инструкције за приступ меморији су инструкција преноса у регистар (LD), и инструкција преноса из регистра (ST), и за њих су усвојени кодови операција 24 и 25. Инструкције користе регистарско индиректно са померајем адресирање. Први операнд је дат регистром опште намене R0 до R31 а специфициран је битовима 25 до 21 инструкције. Други операнд је дат садржајем једног од регистара опште намене R0 до R31 а специфициран је битовима 20

до 16 инструкции. Померај је дат са најнижих 16 бита инструкции који представљају непосредну величину са знаком дату у другом комплементу. Код инструкции LD операнд се чита са меморијске локације чија се адреса добија сабирањем другог операнда и помераја, а резултат смешта у регистар специфициран првим операндом. Инструкција поставља аритметичке индикаторе N и Z. Код инструкции ST други операнд се смешта на меморијску локацију почев од адресе која се добија сабирањем првог операнда и помераја. Инструкција не поставља аритметичке индикаторе.

Инструкције за приступ улазно/излазном адресном простору су инструкција преноса у регистар (IN), и инструкција преноса из регистра (OUT), и за њих су усвојени кодови операција 26 и 27. Први операнд је дат регистром опште намене R0 до R31 а специфициран је битовима 25 до 21 инструкции. Други операнд је дат садржајем једног од регистра опште намене R0 до R31 а специфициран је битовима 20 до 16 инструкции. Најнижих 16 бита инструкции се не користе. Код инструкции IN операнд се чита из улазно/излазног адресног простора се адреса специфициране другим операндом, а резултат смешта у регистар специфициран првим операндом. Инструкција не поставља аритметичке индикаторе. Код инструкции OUT други операнд се смешта улазно/излазни адресни простор почев од адресе дате првим операндом. Инструкција не поставља аритметичке индикаторе.

Битови 31 до 26 инструкции имају вредност у интервалу од 32 до 51 за све инструкции условног скока. Начин адресирања је специфициран битом 26 инструкции. Померај је 21 битна целобројна величина са знаком дата битима 20 до 0 инструкции. Уколико бит 26 има вредност 0 ради се PC релативном адресирању код кога се адреса локације на којој се налази адреса скока добија сабирањем регистра PC и помераја. Бити 25 до 21 се не користе. Уколико бит 26 има вредност 1 ради се о регистарски индиректном начину адресирања код кога се адреса скока добија сабирањем специфицираног регистра и помераја. Регистар опште намене R0 до R31 ја специфициран је битовима 25 до 21 инструкции. Инструкције не постављају индикаторе. Инструкције условног скока су:

Инструкција	Код	Значење	Услов
JE	32/33	скок на једнако	$Z = 1$
JNE	34/35	скок на неједнако	$Z = 0$
JGE	36/37	скок на веће него или једнако (са знаком)	$N \oplus V = 0$
JG	38/39	скок на веће него (са знаком)	$(N \oplus V) \vee Z = 0$
JLE	40/41	скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$
JL	42/43	скок на мање него (са знаком)	$(N \oplus V) = 1$
JP	44/45	скок на $N = 1$	$N = 1$
JNP	46/47	скок на $N = 0$	$N = 0$
JO	48/49	скок на $V = 1$	$V = 1$
JNO	50/51	скок на $V = 0$	$V = 0$

Инструкције безусловног скока је и инструкција скока на потпрограм (CALL) за коју су усвојени кодови 64 и 65. Начин адресирања је специфициран битом 26 инструкции, а померај је 21 битна целобројна величина са знаком дата битима 20 до 0 инструкции. Уколико бит 26 има вредност 0 ради се PC релативном адресирању код кога се адреса локације на којој се налази адреса скока добија сабирањем регистра PC и помераја. Бити 25 до 21 се не користе. Уколико бит 26 има вредност 1 ради се о регистарски индиректном начину адресирања код кога се адреса скока добија сабирањем специфицираног регистра и помераја. Регистар опште намене R0 до R31 ја специфициран је битовима 25 до 21 инструкции.

Инструкција скока је и инструкција прекида (INT) за коју је усвојен код 66. Број улаза у табелу са адресама прекидних рутина је 5 битна целобројна величина без знака дата битима 25 до 21 инструкции. Најнижих 21 бит инструкции се не користе.

Битови 31 до 26 инструкции имају вредност 67 за безадресне инструкције. Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из

прекидне рутине (RTI) и инструкција безусловног скока (JMP). Битовима 25 до 21 инструкције специфицира се код операције за безадресне инструкције. Усвојени су кодови операција 0, 1 и 2, респективно. Код инструкција RTS и RTI најнижих 21 бита инструкције се не користе. Код инструкције JMP адреса на коју се скаче се добија сабирањем регистра PC и помераја. Померај је дат са најнижих 21 бита инструкције који представљају непосредну величину са знаком дату у другом комплементу.

Безадресне инструкције су и инструкција преноса из регистра на стек (PUSH), и инструкција преноса са стека у регистар (POP), преноса програмске статусне речи на стек (PUSHF), и инструкција преноса са стека у програмску статусну реч (POPF) и за њих су усвојени кодови операција 8, 9, 10 и 11. Код инструкција PUSH и POP регистар опште намене R0 до R31 је специфициран битовима 21 до 16 инструкције. Битови 15 до 0 инструкције се не користе. Код инструкција PUSHF и POPF битови 21 до 0 инструкције се не користе. Инструкције PUSH и PUSHF не поставља аритметичке индикаторе, док инструкција POP поставља аритметичке индикаторе N и Z.

Спољашњи маскирајући захтеви за прекид долазе од 7 улазно/излазна уређаја по линијама intr_7 до intr_1 . Ови прекиди се називају спољашњи прекиди јер долазе од уређаја ван процесора, као и маскирајући прекиди, јер су дозвољени или маскирани и процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI регистра програмске статусне речи PSW налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем тако што се у разред PSWI регистра PSW уписују вредности 1 или 0, респективно. Нема селективног маскирања прекида коришћењем регистра маске IMR. Прекиди који долазе по линијама intr_7 до intr_1 уређени су по приоритетима при чему линија intr_7 има највиши, а линија intr_1 најнижи ниво приоритета. Захтеви за прекид се прихватају уколико је њихов ниво виши од нивоа приоритета текућег програма. Спољашњи немаскирајући захтев за прекид долази по линији intr_m . Унутрашњи прекиди су прекид због грешке у коду операције, прекид због грешке у адресирању и прекид због задатог режима рада прекид после сваке инструкције. У случају прекид због грешке у коду операције и прекид због грешке у адресирању на стеку се чува PC почетка инструкције која је изазвала прекид.

Опслуживање захтева за прекид се састоји из две групе корака. У оквиру прве групе корака на стеку се чувају програмски бројач PC и програмска статусна реч PSW, затим се у разреде PSWI и PSWT програмске статусне речи PSW уписују вредности 0 и у случају маскирајућих прекида у разреде PSWL2, PSWL1 и PSWL0 регистра PSW уписује ниво приоритета прекидне рутине на коју се скаче. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*) која почиње од адресе 0, и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе 0, креирана IV табела са 16 улаза, тако да се у улазима 15 до 9 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr_7 до intr_1 , респективно. Адресе прекидних рутина за прекид због грешке у коду операције, грешке у адресирању, спољашњи немаскирајући прекид и прекид због задатог режима рада прекид после сваке инструкције, налазе се у улазима 3, 2, 1 и 0 табеле са адресама прекидних рутина.

Реализовати процесор и меморију, процесор обавља арбитражију магистрале.

Операциона јединица треба да буде реализована са три магистрале.

Управљачку јединицу процесора реализовати микропрограмском реализацијом са хоризонталним кодирањем управљачких сигнала са једним типом микроинструкција.

Магистрала је синхрона са атомским циклусима читања и уписа.

Процесор, меморија и магистрала имају исти сигнал такта.

Ставке које је потребно обавити пре доласка на одбрану:

1. Пројекат се предаје у електронској верзији
2. Нацртати (без коментара) операциону и управљачку јединицу процесора.
3. Нацртати дијаграм тока фаза извршавања инструкције процесора и то: фазе читања инструкције, фазе формирања адресе и читања операнда, фаза извршавања операција и фазе опслуживања захтева за прекид користећи приложени формулар.
4. Развити и конфигурисати симулатор реализованог рачунарског система на RTL нивоу (по угледу на симулатор дат у архиви AOR_2015_2016.zip).
5. Потребно је написати и дати садржај оперативне меморије за сет програма којима би се обавило тестирање реализованог рачунарског система. Овај сет програма треба да садржи и програме који обављају следеће операције:
 - а) Проналази минималну вредност низа дужине 20 који је задата почев од меморијске адресе 1000h.
 - б) Програм и потпрограм који рачуна аритметичку средину елемента низа дужине 32. Потпрограм почетну адресу низа прима као аргумент преко регистра R0. Резултат остаје у регистру R0.
 - в) Програм који иницијализује улазе у табелу прекидних рутина (IV табела) и који демонстрира рад прекида после сваке инструкције, инструкције прекида INT и грешке усред нелегалног кода операције.