



Основи рачунарске технике 2

- пројекат 2022/2023 -

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

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

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

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

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

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

За адресне инструкције се битовима 7 или 7, 6, 5 и 4 другог бајта инструкције се специфицира начин адресирања. Процесор поседује следеће начине адресирања: кратко непосредно адресирање (shortimmed), регистарско директно адресирање (regdir), регистарско индиректно адресирање (regind), меморијско директно адресирање (memdir), меморијско индиректно адресирање (memind), непосредно адресирање (immed), регистарско индиректно са предекрементирањем (predec) и базно-индексног адресирање са постинкрементирањем индексног регистра (bxpostinc). Бит 7 другог бајта инструкције има вредност 0 за кратко непосредно адресирање, док код осталих адресирања бит 7 другог бајта инструкције има вредност 1. Код кратког непосредног адресирања битови 6 до 0 другог бајта инструкције садржи неозначени податак. Дужина инструкција је два бајта. Код регистарског директног, регистарског индиректног адресирања и регистарског индиректног адресирања са предекрементирањем се битовима 7, 6, 5 и 4 дефинише сам начин адресирања 1000, 1001 и 1010, респективно, при чему се користе неки од регистра опште намене R[0] до R[15] специфициран битовима 3 до 0 другог бајта инструкције. Дужина инструкција је два бајта. Меморијско директно, меморијско индиректно, непосредно адресирање и базно-индексног адресирање са постинкрементирањем индексног регистра се дефинише битовима 7, 6, 5 и 4 другог бајта инструкције, и то 1100, 1101, 1110, 1111, респективно. Код меморијског директног и меморијског индиректног адресирања трећи и четврти бајт инструкције садрже адресу меморијске локације. Битови 3 до 0 другог бајта инструкције се не

користе. Дужина инструкција је четири бајта. Код непосредног адресирања трећи и четврти бајт инструкције садрже податак. Дужина инструкција је четири бајта. Код базно-индексног адресирања са постинкрементирањем индексног регистра, базни регистар представља регистар опште намене R[13], а индексни регистар представља регистар опште намене R[14]. Битови 3 до 0 другог бајта инструкције се не користе. Дужина инструкција је два бајта.

Стек расте према вишим меморијским локацијама, а за регистар који указује на врх стека се користи регистар опште намене R[15]. Регистар R[15] указује на последњу заузету меморијску локацију. Сматрати да је регистар SP у симулатору помоћни регистар који није део архитектуре.

Безадресне инструкције су инструкција повратка из потпрограма (RTS), инструкција повратка из прекидне рутине (RTI), инструкција аритметичког померања садржаја акумулатора удесно за једно место (ASR), инструкција логичког померања садржаја акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција ротирања садржаја акумулатора и индикатора C удесно за једно место (RORC), инструкција аритметичког померања садржаја акумулатора улево за једно место (ASL), инструкција логичког померања садржаја акумулатора улево за једно место (LSL), инструкција ротирања садржаја акумулатора улево за једно место (ROL), инструкција ротирања садржаја акумулатора и индикатора C улево за једно место (ROLC), инструкција постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржаја акумулатора A у регистар IVTP (STIVTP), инструкција стављања садржаја акумулатора A на стек (PUSH), инструкција скидања податка са стека и смештање у акумулатор A (POP), инструкција поствљања вредности акумулатора A на 0 (CLR) и инструкција заустављања рада процесора (HALT).

Инструкције условног скока су:

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

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

Адресне инструкције су инструкција преноса у акумулатор (LD), инструкција преноса из акумулатора (ST), инструкција која рачуна дужину знаковног низа (сматрати да се знаковни низ завршава нулом) и резултат смешта у акумулатор (LEN), аритметичка инструкција сабирања (ADD), инструкција инкрементирања операнда при чему се **садржај акумулатора не мења**

(INC), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), аритметичка инструкција негирања у другом комплементу (NEG), инструкција одређивања максимума између акумулатора и операнда (MAX) при чему се већа вредност смешта у акумулатор, инструкција упоређивања операнда са акумулатором А без измене садржаја акумулатора (CMP). Потребно је поставити индикаторе PSWN, PSWZ, PSWC и PSWV на основу резултата инструкције CMP.

Спољашњи маскирајући захтеви за прекид долазе од 7 улазно/излазна уређаја по линијама intr7 до intr1. 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 PSW15...0 налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра PSW15...0 уписују вредности 1 или 0, респективно. Уколико се јави непостојећи код операције или грешка у начину адресирања, инструкција треба да буде без дејства.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач PC15...0 и програмска статусна реч PSW15...0. У оквиру друге групе корака утврђује се адреса прекидне рутине. Утврђивање адресе прекидне рутине се реализује на основу садржаја табеле адреса прекидних рутина, која се обично назива IV табела (*Interrupt Vector Table*), и броја улаза у IV табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра IVTP15...0 (*Interrupt Vector Table Pointer*), креирана IV табела са 16 улаза, тако да се у улазима од 15 до 9 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама intr7 до intr1, респективно. Како је меморијска реч 8 битна величина, а адреса прекидне рутине 16 битна величина, то сваки улаз у IV табели заузима по две суседне меморијске локације. Због тога се најпре број улаза (UEXT) множењем са два претвара у померај, па затим померај сабира са садржајем регистра IVTP15...0 и на крају добијена вредност користи као адреса са које се чита адреса прекидне рутине и уписује у регистар PC15...0.

НАПОМЕНА:

1. Постављање индикатора N, Z, C и V у инструкцијама ASR, LSR, ROR, RORC, ASL, LSL, ROL, RORC, LD, ADD, SUB, AND, OR, XOR треба да буде урађена на исти начин као што је то урађено у случају процесора датог у архиви AOR_X.Y.Z.zip.
2. Реализација процесора треба да буде урађена на исти начин као што је то урађено у случају процесора датог у архиви AOR_X.Y.Z.zip. Посебно водити рачуна о декодовању инструкција тако да групе и подгрупе прате поставку задатка.
3. Студент сам дефинише кодове инструкција у складу са захтевима пројекта.
4. Константе које су део архитектуре процесора није могуће држати у меморији.