

## 7 ЗАДАТАК

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

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

Битови 7, 6, 5 и 4 првог бајта инструкције су 0000 за све инструкције условног скока. Битовима 3 до 0 првог бајта инструкције специфицира 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 инструкције је два бајта.

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

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

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

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

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

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

акумулатора удесно за једно место (LSR), инструкција ротирања садржаја акумулатора удесно за једно место (ROR), инструкција 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 постављања индикатора I на 1 (INTE), инструкција постављања индикатора I на 0 (INTD), инструкција преноса садржаја акумулатора A у регистар IVTP (STIVTP) и инструкција преноса садржаја акумулатора A у регистар SP (STSP), инструкција стављања свих програмски доступних регистара на стек (PUSHALL), инструкција скидања садржаја са стека у све програмски доступне регистре (обрнут редослед у односу на постављање) (POPALL).

Инструкције условног скока су:

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

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

Адресне инструкције су инструкција преноса у акумулатор (LD), учитавање операнда дужине 1 бајт у нижи бајт акумулатора без промене вишег бајта акумулатора (LOADL), инструкција преноса из акумулатора (ST), учитавање адресе операнда у 16-битни акумулатор (LEA), аритметичка инструкција сабирања (ADD), аритметичка инструкција одузимања (SUB), логичка инструкција И (AND), логичка инструкција ИЛИ (OR), логичка инструкција ексклузивно ИЛИ (XOR), логичка инструкција инвертовања (NOT).

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  $PSW_{15...0}$  налази вредност 1 или 0, респективно. Дозвољавање и маскирање прекида се реализује програмским путем извршавањем инструкција INTE и INTD којима се у разред PSWI регистра  $PSW_{15...0}$  уписују вредности 1 или 0, респективно. Уколико се јави непостојећи код операције или грешка у начину адресирања (непосредно адресирање код инструкције преноса из акумулатора, или непосредно/регистарско директно адресирање код инструкције срачунатог скока (ако постоји)) инструкција треба да буду без дејства.

Опслуживање захтева за прекид се састоји из две групе корака.

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

табелу. Стога је у поступку иницијализације целог система у меморији, почев од адресе на коју указује садржај регистра  $IVTP_{15...0}$  (*Interrupt Vector Table Pointer*), креирана IV табела са 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. 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  $IVTP_{15...0}$  и на крају добијена вредност користи као адреса са које се чита адреса прекидне рутине и уписује у регистар  $PC_{15...0}$ .

Магистрала је синхрона са атомским циклусима читања и уписа. Процесор обавља функцију арбитратора са једним паром линија за захтев и потврду. Процесор и меморија имају исти сигнал такта, не постоји посебан сигнал такта магистрале. Не постоји сигнал заузећа магистрале.

Реализовати процесор, меморију и арбитратор 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.

#### НАПОМЕНА:

1. Постављање индикатора N, Z, C и V у инструкцијама ASR, LSR, ROR, ASL, LSL, ROL, LD, ADD, SUB, AND, OR, XOR, NOT треба да буде урађено 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.