169 ЗАДАТАК

Посматра се део рачунара који чине меморија и процесор.
Меморија је капацитета 2^{16} бајтова. Ширина меморијске речи је 1 бајт.

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

У процесору постоје шеснаест регистара опште намене који се налазе у регистарском фајлу, програмска статусна реч PSW, регистар IVTP (*Interrupt Vector Table Pointer*), адресни регистар меморије MAR, прихватни регистар податка меморије MDR и прихватни регистар инструкције IR. Процесор не поседује посебан регистар акумулатора A, указивача на врх стека SP и програмског бројача PC, већ за сврху акумулатора користи нулти регистар регистара опште намене, за сврху указивача на врх стека користи се први регистар регистара опште намене, а за програмски бројач користи се други регистар регистара опште намене.

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

1) Безадресне инструкције

Инструкција	Значење	$IR_{31..24}$	$IR_{23..16}$	$IR_{15..8}$	$IR_{7..0}$	Дужина
HALT	заустављање рада процесора	0000 0000b	/	/	/	1B
RTS	повратак из потпрограма	0000 0001b	/	/	/	1B
RTI	повратак из прекидне рутине	0000 0010b	/	/	/	1B
ASL	аритметичко померање улево за једано место	0000 0011b	/	/	/	1B
ROLC	ротирање садржаја акумулатора улево кроз индикатор C	0000 0100b	/	/	/	1B
PUSHGPR	стављање садржаја свих регистара опште намене на стек (R0-R15)	0000 0101b	/	/	/	1B
POPGPR	пуњење садржаја свих регистара опште намене на стек (R15-R0)	0000 0110b	/	/	/	1B

2) Инструкције условног скока (попуњавају се само прва три бајта IR регистра)

Инструкција	Значење	Услов	$IR_{31..24}$	$IR_{23..16}$	$IR_{15..8}$	Дужина
BLSS	скок на мање него (са знаком)	$(N \oplus V) = 1$	0001 0000b	PPPP PPPPb	/	2B
BNCR	скок на C = 0	C = 0	0001 0001b	PPPP PPPPb	/	2B
BGRTU	скок на веће него (без знака)	$C \vee Z = 0$	0001 0010b	PPPP PPPPb	/	2B
BNEG	скок на N = 1	N = 1	0001 0011b	PPPP PPPPb	/	2B
JLEQ	апсолутни скок на мање него или једнако (са знаком)	$(N \oplus V) \vee Z = 1$	0001 0100b	адреса скока		3B
LOOPGRE	декрементира садржај акумулатора и скаче на адресу уколико је вредност у акумулатору већа или једнака 0.	$N \oplus V = 0$	0001 0101b	адреса скока		3B

3) Инструкције безусловног скока

Инструкција	Значење	$IR_{31..24}$	$IR_{23..16}$	$IR_{15..8}$	$IR_{7..0}$	Дужина
JMP	апсолутни скок	0010 0000b	адреса скока		/	3B
JSR	апсолутни скок на потпрограм	0010 0001b	адреса скока		/	3B

4) Адресне инструкције

Инструкција	Значење	$IR_{31..24}$	Дужина
LD	инструкција преноса у акумулатор	0011 0000b	Зависи од начина адресирања
ST	инструкција преноса из акумулатора	0011 0001b	
SUB	аритметичка инструкција одузимања	0011 0010b	

Инструкција	Значење	$IR_{31..24}$	Дужина
AND	логичка инструкција И	0011 0011b	Зависи од начина адресирања
NOT	логичка инструкција инвертовања	0011 0100b	

Начини адресирања:

Адресирање	Значење	$IR_{23..16}$	$IR_{15..8}$	$IR_{7..0}$	Дужина
immed	непосредно адресирање	0010 0000b	податак		4B
memdir	меморијско директно адресирање	0100 0000b	адреса податка		4B
regdir	регистарско директно адресирање	011R RRRRb	/	/	2B
memind	меморијско индиректно адресирања	1000 0000b	адреса податка		4B
postincr	регистарско индиректно са постинкрементирањем адресирање	110R RRRRb	/	/	2B
pcrelpom	PC релативно са померајем адресирање	1110 0000b	PPPP PPPPb	/	3B

X – битови који се не користе.
R – битови који означавају индекс регистра опште намене који се користи.
P – битови који представљају померај са знаком.

Формат PSW регистра:

15	14	13	12	11	10	9	8
PSWI	/	/	/	/	/	/	/

7	6	5	4	3	2	1	0
/	/	/	PSWC	PSWV	PSWN	PSWZ	PSWSTART

Неактивна бредност бита PSWSTART зауставља рад процесора, док активна вредност враћа процесор у рад.

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

Захтеве за прекид може да генерише четири контролера периферија који су повезани на већ реализован блок INTERRUPT_INTERFACE_4. На улазе $BTN\_INTR_{3..0}$ у блок INTERRUPT_INTERFACE_4 треба довести осам дугмета која симулирају захтеве за прекид контролера периферија. На улаз $UEXT_{1..0}$ треба довести бинарну вредност која представља индекс прихваћеног захтева за прекид. На улаз *inta* треба довести сигнал који је активан у случају да се прихвата неки од захтева за прекид (сигнал за учитавање у регистар BRU). Излаз блока $intr_{3..0}$ представља запамћене захтеве за прекид. Ови прекиди се називају спољашњи маскирајући прекиди јер долазе од уређаја ван процесора и могу бити дозвољени или маскирани јер процесор на њих реагује или не реагује у зависности од тога да ли се у разреду PSWI registra програмске статусне речи PSW налази вредност 1 или 0, респективно. Сматрати да процесор реагује само на ову врсту прекида.

Опслуживање захтева за прекид се састоји из две групе корака.

У оквиру прве групе корака на стеку се чувају програмски бројач PC, акумулатор A и програмска статусна речи PSW. 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 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, креирана IV табела са 4 улаза, тако да се у улазима 3 до 0 налазе адресе прекидних рутина за сваки од прекида који долазе по линијама $intr_7$ до $intr_0$ који долазе из блока INTERRUPT_INTERFACE_4, респективно. Прекиди који долазе по линијама $intr_3$ до $intr_0$ треба уредити по приоритету при чему линија $intr_3$ има највиши, а линија $intr_0$ најнижи ниво приоритета. Број улаза у IV табелу треба да генерише процесор на

основу позиције линије $intr_3$ до $intr_0$ највишег нивоа приоритета на којој постоји захтев за прекид.

Реализовати процесор према задатој спецификацији његове архитектуре, и то помоћу блокова FETCH, ADDR, EXEC, INTR и COMMON:

Блок са заједничким секвенцијалним и комбинационим мрежама (COMMON блок). Блок који садржи помоћне регистре, флип-флопове и комбинационе модуле који се користе у више него једној фази извршавања инструкције.

За симулацију процесора потребно је додати дугме BTN_RST који генерише сигнал *rst*. Активна вредност сигнала *rst* враћа процесор у почетно стање, а у регистар PC уписује вредност 1000h, у регистар PSW 8001h, у регистар SP F000h, у акумулатор A 0h и у регистар IVTP 0h. Сигнал *rst* треба искористити у сваком реализованом блоку.

а) [5 поена] Блок дохватања инструкције (FETCH блок). Блок FETCH креће са фазом читања инструкције уколико се и у флип-флопу FETCH и у биту PSWSTART налази вредност 1. По завршеном читању инструкције уписивањем вредности 1 у флип-флопове ADDR или EXEC стартује се блок ADDR или блок EXEC, док се уписивањем вредности 0 у флип-флоп FETCH зауставља блок FETCH. Дефинисати сигнал *grinst* који је активан уколико је прочитана инструкција са недефинисаним операционим кодом или у случају недефинисаног начина адресирања или у случају недозвољене комбинације операционог кода и начина адресирања. Одмах при активирању сигнала *grinst* прећи на учитавање следеће инструкције.

б) [10 поена] Блок формирање адресе и дохватање операнда (ADDR блок). Блок ADDR креће са формирањем адресе операнда и читањем операнда уколико се у флип-флопу ADDR налази вредност 1. По завршеном формирању адресе и дохватања операнда уписивањем вредности 1 у флип-флоп EXEC стартује се блок EXEC и продужава се са извршавањем фазе извршавања операције, док се уписивањем вредности 0 у флип-флоп ADDR зауставља блок ADDR.

в) [10 поена] Блок извршавања операције (EXEC блок). Блок EXEC креће са фазом извршавања операције уколико се у флип-флоп EXEC налази вредност 1. По завршеном извршавању операције уписивање вредности 1 у флип-флоп INTR стартује се блок INTR и продужава се са извршавањем фазе опслуживања прекида, док се уписивањем вредности 0 у флип-флоп EXEC зауставља блок EXEC.

г) [5 поена] Блок опслуживања прекида (INTR блок). Блок INTR креће са фазом опслуживања прекида уколико се у флип-флопу INTR налази вредност 1. По завршетку опслуживања прекида уписивањем вредности 1 у флип-флоп FETCH стартује се блок FETCH и креће се са фазом читања следеће инструкције, док се уписивањем вредности 0 у флип-флоп INTR зауставља блок INTR.

Операциона јединица сваког блока треба да буде реализована директним повезивањем прекидачких мрежа, а сваки блок осим COMMON блока треба да има управљачку јединицу реализовану микропрограмирањем.

Напомена: Начин функционисања блокова FETCH, ADDR, EXEC и INTR треба да буде имплементиран као у литератури (са тим да се заједнички елементи налазе у блоку COMMON). Студенту се препоручује да направи тест програме који тестирају реализоване блокове.

Линкови:

- https://rti.etf.bg.ac.rs/rti/ir2ort2/literatura/Projektovanje_dela_procesora.pdf
- https://rti.etf.bg.ac.rs/rti/ir2ort2/literatura/Organizacija_procesora.pdf