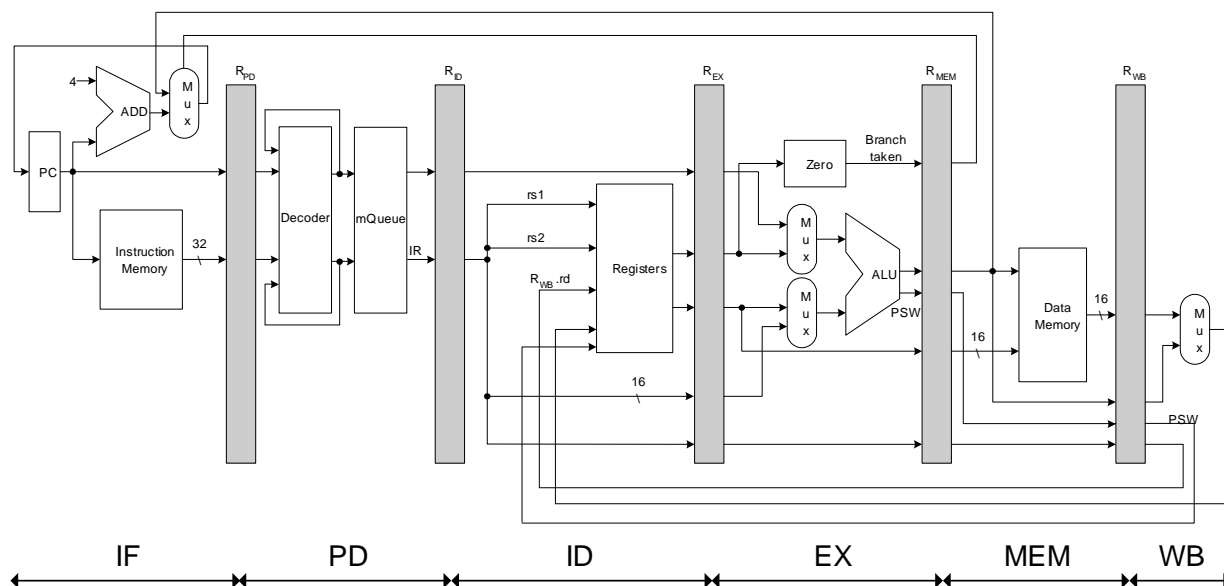


ТРАНСЛАЦИЈА ИНСТРУКЦИЈА

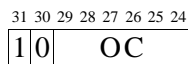
1. Разматра се рачунарски систем у коме се извршавање одређене инструкције одвија у 6 фаза помоћу измењеног процесора са стандардном проточном обрадом (слика 1.1). У процесор са стандардном проточном обрадом је додата као други степен јединица PD (Instruction PreDecode) који обавља трансформацију инструкција задате архитектуре у инструкције RISC архитектуре. Сматрати да приступ меморији траје два сигнала такта.



Слика 1.1 – организација процесора

Формати инструкција изворишне CISC архитектуре су дати у наставку.

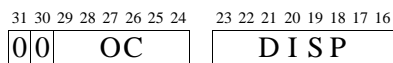
1. Безадресне инструкције – (RTS, RTI, ASR)



Слика 1 Безадресне инструкције

Пољем OC се специфицира операција која се извршава.

2. Инструкције условног скока – (BZ)



Слика 2 Инструкције условног скока

Пољем OC се специфицира операција која се извршава, а пољем DISP 8-то битни померај као целобројна величина са знаком који се сабира са PC да би се добила адреса скока.

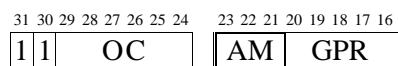
3. Инструкције безусловног скока – (JMP, JSR)



Слика 3 Инструкције безусловног скока

Пољем OC се специфицира операција која се извршава, а пољима ADRH и ADRL 8 старијих и 8 млађих битова 16-то битне адресе скока.

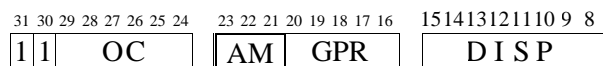
4. Адресне инструкције – (LD, ST, ADD, AND, JADR) са regdir, regind, predecr, postincr



Слика 4 Адресне инструкције са regdir, regind, postdecr, preincr

Пољем OC се специфицира код операције једноадресне инструкције, пољем AM регистарско директно, регистарско индиректно, регистарско индиректно са преинкрементирањем или регистарско индиректно са постдекрементирањем адресирање и пољем GPR један од 32 регистра опште намене.

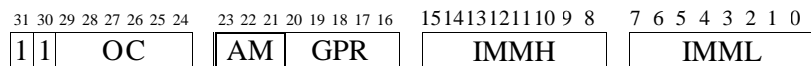
5. Адресне инструкције – (LD, ST, ADD, AND, JADR) са regindpom



Слика 5 Адресне инструкције са regindpom

Пољем OC се специфицира код операције једноадресне инструкције, пољем AM регистарско индиректно са померајем адресирање, пољем GPR један од 32 регистра опште намене и пољем DISP 8-то битни померај као целобројна величина са знаком.

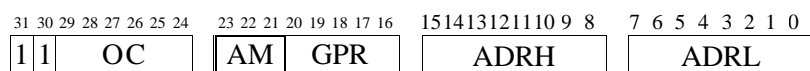
6. Адресне инструкције – (LD, ST, ADD, AND, JADR) са immed



Слика 6 Адресне инструкције са immed

Пољем OC се специфицира код операције једноадресне инструкције, пољем AM непосредно адресирање, поље GPR се не користи и пољима IMMH и IMML 8 старијих и 8 млађих битова непосредне 16-битне величине.

7. Адресне инструкције – (LD, ST, ADD, AND, JADR) са memdir, memind



Слика 7 Адресне инструкције са memdir и memind

Пољем OC се специфицира код операције једноадресне инструкције, пољем AM меморијско директно или меморијско индиректно адресирање, поље GPR се не користи и пољима ADRH и ADRL 8 старијих и 8 млађих битова 16-то битне адресе.

а) Написати секвенцу инструкција (микроинструкција) циљне RISC архитектуре у коју се обавља пресликавање за сваку инструкцију изворишне CISC архитектуре.

б) Нацртати формате инструкција циљне RISC архитектуре.

в) Шта је све потребно променити у процесору са стандардном проточном обрадом да би се омогућило динамичко превођење инструкција.

РЕШЕЊЕ

a)

Асемблерска инструкција	Акција	Микро инструкције	Коментар
ADD Rx	$ACC=ACC+Rx$	ADDS R33, R33, Rx	
ADD (Rx)	$ACC=ACC+MEM[Rx]$	LD R34, (Rx)0 ADDS R33, R33, R34	Хазард по подацима.
ADD (Rx)+	$ACC=ACC+MEM[Rx]$ $Rx=Rx+1$	LD R34, (Rx)0 ADDI Rx, Rx, #1 ADDS R33, R33, R34	
ADD -(Rx)	$Rx=Rx-1$ $ACC=ACC+MEM[Rx]$	LD R34, (Rx)-1 ADDI Rx, Rx, #-1 ADDS R33, R33, R34	
ADD (Rx)disp	$ACC=ACC+MEM[Rx+disp]$	LD R34, (Rx)disp ADDS R33, R33, R34	Хазард по подацима.
ADD adr	$ACC=ACC+MEM[adr]$	LD R34, (R35)adr ADDS R33, R33, R34	Хазард по подацима.
ADD (adr)	$ACC=ACC+MEM[MEM[adr]]$	LD R34, (R35)adr LD R34, (R34)0 ADDS R33, R33, R34	Хазард по подацима.
ADD #imm	$ACC=ACC+imm$	ADDIS R33, R33, imm	
Слично за AND
LD Rx	$ACC=Rx$	ADDS R33, R35, Rx	
LD (Rx)	$ACC=MEM[Rx]$	LDS R33, (Rx)0	
LD (Rx)+	$ACC=MEM[Rx]$ $Rx=Rx+1$	LDS R33, (Rx)0 ADDI Rx, Rx, #1	
LD -(Rx)	$Rx=Rx-1$ $ACC=MEM[Rx]$	LDS R33, (Rx)-1 ADDI Rx, Rx, #1	
LD (Rx)disp	$ACC=MEM[Rx+disp]$	LDS R33, (Rx)disp	
LD adr	$ACC=MEM[adr]$	LDS R33, (R35)adr	
LD (adr)	$ACC=MEM[MEM[adr]]$	LD R34, (R35)adr LDS R33, (R34)0	Хазард по подацима.
LD #imm	$ACC=imm$	ADDS R33, R35, imm	
ST Rx	$Rx=ACC$	ADD Rx, R35, R33	
ST (Rx)	$MEM[Rx]=ACC$	ST R33, (Rx)0	
ST (Rx)+	$MEM[Rx]=ACC$	ST R33, (Rx)0	

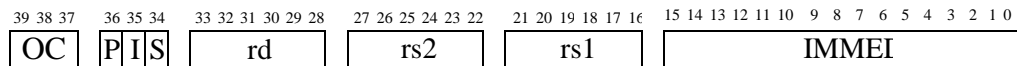
Асемблерска инструкција	Акција	Микро инструкције	Коментар
	$R_x = R_x + 1$	ADDI $R_x, R_x, \#1$	
ST -(R_x)	$R_x = R_x - 1$ MEM[R_x]=ACC	ST R33, (R_x)-1 ADDI $R_x, R_x, -1$	
ST (R_x)disp	MEM[R_x +disp]=ACC	ST R33, (R_x)disp	
ST adr	MEM[adr]=ACC	ST R33, (R35)adr	
ST (adr)	MEM[MEM[adr]]=ACC	LD R34, (R35)adr ST R33, (R34)0	Хазард по подацима.
ST #imm JADR R_x JADR imm	MEM[++SP]=PC MEM[++SP]=PSW AND PSW, #maskIT PC= MEM[IVTP + prekid]	ADDIP R34, PC, #0 ST R34, (R36)2 ST R37, (R36)4 ADDI R36, R36, #4 LD R34, (R38)prekid ANDI R37, R37, #maskIT BEQZP R35, (R34)0	Хазард по подацима.
JADR (R_x)	PC= R_x	BEQZ R35, (R_x)0	
JADR (R_x)+	PC= R_x $R_x = R_x + 1$	BEQZ R35, (R_x)0 ADDI $R_x, R_x, \#1$	
JADR -(R_x)	$R_x = R_x - 1$ PC= R_x	ADDI $R_x, R_x, \#-1$ BEQZ R35, (R_x)0	
JADR (R_x)disp	PC = R_x +disp	ADDI R34, R34, disp BEQZ R35, (R34)0	
JADR adr	PC = adr	BEQZ R35, (R35)adr	
JADR (adr)	PC = MEM[adr]	LD R34, (R35)adr BEQZ R35, (R34)0	Хазард по подацима.
BZ (PC)disp	If (PSWZ==0) PC=PC+disp	ANDI R34, R37, #maskZ BEQZP R34, (PC)disp	
JMP adr	PC=adr	BEQZ R35, (R35)disp	
RTI	PSW=MEM[SP--] PC=MEM[SP--]	LDS R37, (R36)0 LD R34, (R36)-2 ADDI R36, R36, #4 BEQZ R35, (R34)0	
RTS	PC=MEM[SP--]	LD R34, (R36)0 ADDI R36, R36, #-2 BEQZ R35, (R34)0	
ASR	ACC=ACC>>1	ASRIS R33, R33, #1	

ACC → R33
 TMP → R34
 0 → R35
 SP → R36
 PSW → R37
 IVTP → R38

CODE[[P] | [I] | [S]]

P – користи копију регистра PC (У EX степену проточне обраде утиче на MUX)
 I – користи непосредну величину (У EX степену проточне обраде утиче на MUX)
 S – поставља регистар PSW

б) Све инструкције RISC архитектуре су фиксне дужине 40 бита. Инструкције које се користе у разматрањима *pipeline* организације процесора имају следећи формат инструкција:



в)

Кеш меморију за инструкције приступ се обавља на нивоу 32 битне речи.

Кеш меморију за податке приступ се обавља на нивоу 16 битне речи.

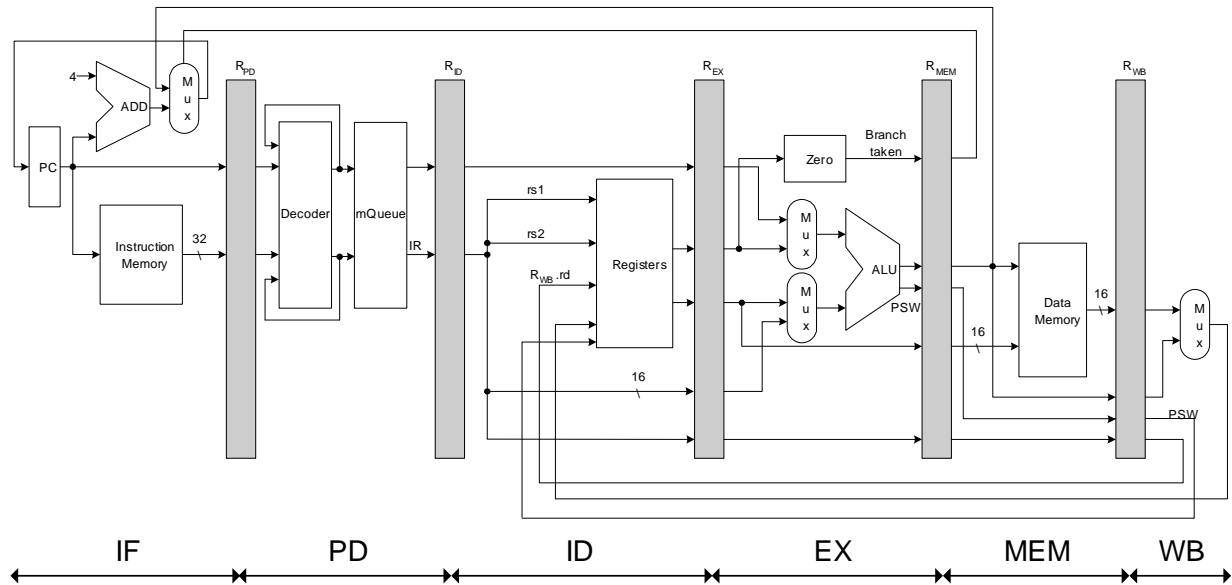
Декодовање (транслација) инструкција се обавља на основу 32 битне речи. Пошто постоје инструкције мање дужине (1, 2 и 3 бајта) приликом декодовање инструкције може се десити да унутар једне речи 32 битне речи буде учитан већи број инструкција (или делова инструкција). Да би се овај проблем решио реч на основу које се декодује инструкција може се састојати од делова две узастопне 32 битне речи.

Већи број прослеђивања.

Проблем око одређивања следеће вредности регистра PC.

Како би се обавило убрзање рада система могуће је коришћење посебне кеш меморије у којој би се уместо инструкција CISC процесора чувале декодоване секвенце инструкције циљног RISC процесора заједно са условима (Trace Cache).

2. Разматра се рачунарски систем у коме се извршавање одређене инструкције одвија у 6 фаза помоћу измењеног процесора са стандардном проточном обрадом (слика 2.1.). У процесор са стандардном проточном обрадом је додата као други степен јединица PD (Instruction PreDecode) који обавља трансформацију инструкција задате архитектуре у инструкције RISC архитектуре. Сматрати да приступ меморији траје два сигнала такта. Архитектура процесора дефинише 16 регистра опште намене. Адресе и подаци су величине 16 бита.



Слика 2.1. – организација процесора

а) Написати секвенцу инструкција (микроинструкција) циљне RISC архитектуре у коју се обавља пресликавање за део инструкцијског сета из табеле 2.1. изворишне CISC архитектуре. Уколико је потребно проширити број регистра опште намене у регистарском фајлу, онда треба за сваки додат регистар написати чему служи. Регистар R13 представља указивач на врх стека (SP) и показује на последњу слободну локацију. Стек расте према вишим адресама. Регистар R14 представља указивач на базну адресу стека (BP). Регистар R0 се користи као акумулатор, регистар R12 се користи као бројачки регистар и као такав има посебне инструкције које га користе. У табели 2.1. акције нису оптимизовано написане, већ описно.

Асемблерска инструкција	Акција	Микро инструкције
SWP (Rx)	MEM[Rx] ↔ ACC	
LEAVE	SP=BP POP BP	
RTI	PSW=MEM[SP--] PC=MEM[SP--]	
LOOP disp	R12=R12-1 IF R12==0 THEN PC=PC+disp	
FASTCALL addr	R10=R0 R11=R1 PUSH PC PC=addr	
ADD (adr)	ACC =ACC+ MEM[MEM[adr]]	

Табела 2.1. – део инструкцијског сета процесора

б) Нацртати формат инструкција циљне RISC архитектуре (на основу инструкција из табеле 2.1.).