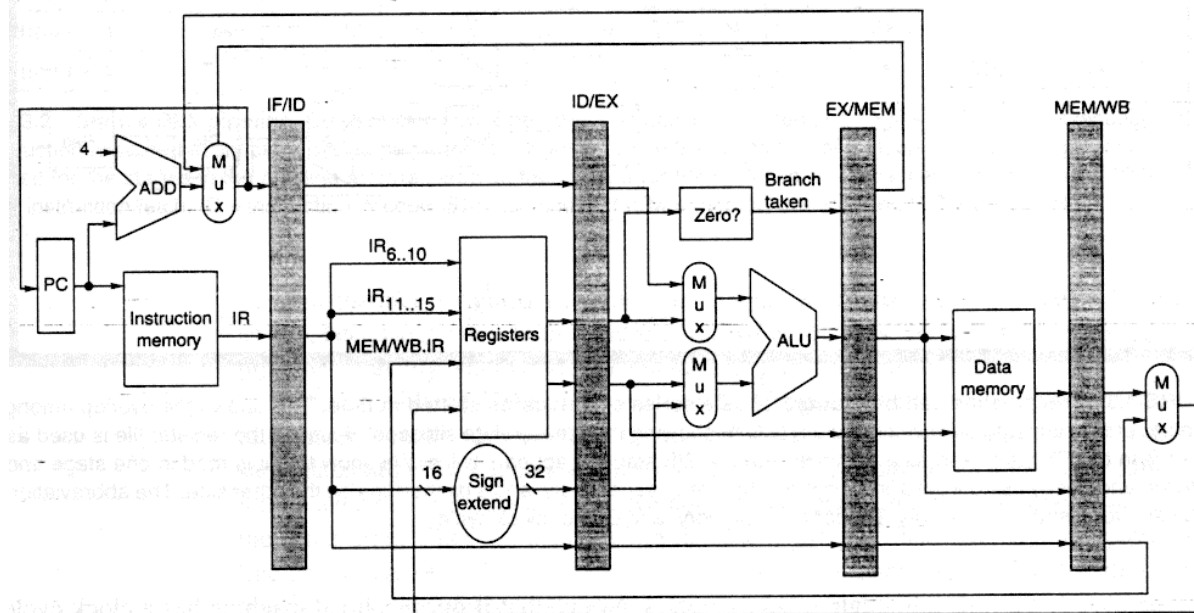




Архитектура и организација рачунара 1 – КЗ

1. (15) Посматра се процесор pipeline организације дат на слици 1. Свака фаза извршавања инструкције траје једну периоду сигнала такта укључујући и фазу 2 у којој се чита из регистарског фајла (Registers) и фазу 5 у којој се уписује у регистарски фајл (Registers).



Слика 1. Pipeline са реализацијом скока у четвртм степену

На процесору се извршава следећа секвенца инструкција:

```
lw    R1, 1(R1);    R1 = Mem[R1+1]
lw    R1, 2(R1);    R1 = Mem[R1+2]
addi  R1, R1, #1;   R1 = R1+1
sw    R1, 3(R1);    Mem[R1+3] = R1
```

У овој секвенци инструкција постоје хазарди података. У неким ситуацијама је могуће реализовати додатни хардвер за прослеђивање којим се елиминишу хазарди података и обезбеђује извршавање инструкција без заустављања pipeline-a. Међутим у неким ситуацијама хазард података је могуће решити једино заустављањем pipeline-a.

а) (8) У датој секвенци инструкција навести све ситуације у којима постоје хазарди података. За сваку од ситуација, навести између које две инструкције се јавља, објаснити зашто постоји хазард података и навести која од њих се може решити заустављањем pipeline-a а која прослеђивањем. Тамо где се зауставља навести колико периода сигнала такта се зауставља, а тамо где се прослеђује (могуће после заустављања) навести шта (који је изворишни регистар) се куда прослеђује (шта је одредиште). Одговор дати табеларно.

Ситуација	Инструкција 1	Инструкција 2	Објашњење	З/П	Заустављање	Прослеђивање Извориште	Одредиште

б) (7) Нацртати модификовану слику процесора са слике 1 која треба да садржи само оне делове који се користе за реализацију прослеђивања која се јављају у овом задатку и за сваки део назначити ситуацију за коју се елиминише хазард.

2. (15) Посматра се систем са стандардном проточном обрадом са слике 1 код кога постоји хардвер за прослеђивање (forwarding) и који користи статичко предвиђање да скок није прихваћен (branch not taken). Извршава се следећи програмски сегмент (све вредности су хексадецималне).

```

Loop:  LW      R1, (R3)0      ; R1 = Mem[R3+0]
        SUB    R5, R1, R2    ; R5 = R1 - R2
        SW     R5, (R3)8     ; Mem[R3+8] = R5
        BEQZ   R5, Exit     ; if (R5 = 0) goto Exit
        ADDI   R3, R3, #1    ; R3 = R3 + 1
        SUBI   R7, R7, #1    ; R7 = R7 - 1
        BNEZ   R7, Loop     ; if (R7 != 0) goto Loop
Exit:   ADDI   R4, R1, #1    ; R4 = R1 + 1
        ADDI   R5, R1, #2    ; R5 = R1 + 2
        ADDI   R6, R1, #3    ; R6 = R1 + 3

```

Почетна вредност регистра R2 је 2h, R3 је 100h и R7 је 7h а изглед дела меморије почев од адресе 100h је приказан на слици 3. Из меморије се чита и у њу уписује 32-битна реч по реч.

Адреса		100	101	102	103	104	105	106	107	108	109	10A	10B	
Садржај		4	3	2	1	0	108	109	10A	3	6	6	9	

Слика 3. Изглед дела меморије

а) Приказати табеларно шта се дешава у којој фази за првих 15 инструкција датог програма које се изврше. Дати вредности које се налазе у регистрима након завршетка 15 инструкција датог програма.

б) Претпоставити да је процесор са слике 1 модификован тако да се утврђивање да ли је до скока дошло или не обавља у другом степену. Приказати табеларно шта се дешава у којој фази за првих 15 инструкција датог програма које се изврше.

Напомене: На испиту нису дозвољена никаква помоћна средства, ни калкулатори ни литература. Испит траје 3 сата.



Архитектура и организација рачунара 1 – К2

1. (15) Рачунар поседује виртуелну меморију сегментне организације и јединицу за убрзавање пресликавања виртуелних у физичке адресе (TLB јединица). Виртуелни адресни простор има 2^5 сегмената, максималне величине 2 Кбајта. Физички адресни простор је величине 512 Мбајта. Адресе у виртуелном и физичком адресном простору се односе на речи ширине 1 бајт. TLB јединица је реализована са асоцијативним пресликавањем, користи LRU алгоритам замене, и може да садржи делове дескриптора 256 сегмената који припадају различитим процесима. Број процеса је 16.

а) (6) Нацртати табелу сегмената и TLB јединицу и означити све капацитете и ширине поља. Усвојити такве величине улаза у табели сегмената, тако да претварање бројева улаза у помераје може да се реализује померањем бројева улаза улево за одређени број места.

б) (9) Посматра се превођење виртуелне у физичку адресу. За сваку карактеристичну ситуацију дати:

- назив;
- за TLB јединицу (ако јој се приступа) назначити на како је добијен број улаза, коју вредност има бит V, а коју бити у пољу таг, да ли се приступ ради хардверски или софтверски;
- за табелу сегмената (ако јој се приступа) назначити како је добијена адреса улаза одакле је прочитан дескриптор сегмента и коју вредност има бит V у дескриптору сегмента, да ли се приступ ради хардверски или софтверски;
- за диск (ако му се приступа) назначити како је добијена адреса на диску.

Карактеристичне ситуације би биле:

1. постоји сагласност у TLB јединици,
2. не постоји сагласност у TLB јединици, а постоји сегмент у оперативној меморији, и
3. не постоји сагласност у TLB јединици нити је сегмент у оперативној меморији.

Одговор дати табеларно. Водити рачуна да за одређене ситуације може бити више редова.

Прилико попуњавања табеле користити ознаке дате под а).

	TLB јединица				Табела сегмената			Диск	
Ситуација	Улаз	V	Таг	X/C	Адреса улаза	V	X/C	Адреса на диску	X/C

2. (5) Посматра се виртуелна меморија и TLB јединица из претходног задатка. Кориснички програм са бројем 5h генерише следећу секвенцу виртуелних адреса (све вредности су хексадецималне): 1134h(Ex), 2145h(Wr), 1137h(Ex), 3167h(Rd) и 3169h(Wr). Сви сегменти се налазе у оперативној меморији, и смештени су почев од локација 10000h, 10200h и 10400h физичке меморије, респективно. Величина сваког сегмента је 512 бајта, а пре дате секвенце TLB јединица је била празна.

За сваки приступ меморији означити: виртуелну адресу којој се приступа, тип операције, идентификацију корисника, вредност поља Segment и поља Word, вредност поља Tag у TLB јединици, коментар да ли је било сагласности у TLB јединици, почетну адресу сегмента у меморији и физичку адресу којој се приступило. Одговор дати табеларно.

3. (10) Рачунарски систем се састоји из три процесора, DMA контролера и оперативне меморије. Меморијске адресе су ширине 32 бита, ширина магистрале података је 32 бита, адресирање је на нивоу 32-битних речи. Меморијски систем је са преклопљеним приступом меморијским модулима, са 16 модула који покривају цео адресни простор, код кога четири најнижа бита адресе одређују број модула.

Претпоставити да је након иницијализације операције уписа у меморију меморијски модул заузет наредне 4 периоде сигнала такта и да је након иницијализације операције читања из меморије модул заузет 3 периоде сигнала такта и одговор даје тек у четвртој. Када процесор или DMA контролер и модул комуницирају, трајање циклуса на магистрали је једна периода сигнала такта.

Циклус повратка прочитаног податка из меморије идентификује се тако што су обе контролне линије *rd* и *wr* активне. Сви захтеви за приступ магистрали постављају се синхронно, на исти сигнал такта. Арбитрација се обавља комбинационо, тако да се у истом такту арбитражују сви захтеви постављени на тај сигнал такта (или раније), даје дозвола једном *master*-у и тај *master* обавља циклус на магистрали. Међу уређајима (процесори и DMA), виши приоритет има уређај са већим идентификационим бројем (ID). DMA контролер има највиши приоритет, редни број му је 3. Међу модулима, виши приоритет има модул са мањим бројем. Циклус враћања прочитаног податка из меморије има највиши приоритет. Уколико се постави захтев модулу који је заузет обрадом раније задате операције, модул одговара негативно у истом такту (*ack*=0), одустаје се од захтева, паузира 2 такта, а онда понавља исти захтев. Уређаји постављају захтеве за операције са меморијом на следећи начин (формат је: такт:ID-адреса(операција)):

1:0-00001000h(Rd), 1:1-00000100h(Rd), 1:2-00000010h(Wr)

DMA контролер је иницијализован да пребацује 4 речи из меморије на периферију почев од адресе 00000001h почевши од такта 1. DMA контролер ради у режиму циклус по циклус. Уколико DMA контролер добије позитиван одговор од текућег меморијског модула (*ack*=1) у наредном такту генерише следећи захтев, уколико постоји.

Приказати како теку циклуси на магистрали за ову секвенцу. Приказ дати табеларно, тако да се по редовима табеле наводе циклуси на магистрали, прва колона даје редни број такта, друга садржај на адресној магистрали, трећа садржај на магистрали података, а четврта, пета и шеста вредности сигнала *rd*, *wr* и *ack*, редом. За вредност на магистрали података током циклуса уписа податка или враћања прочитаног податка из меморије ставити X.