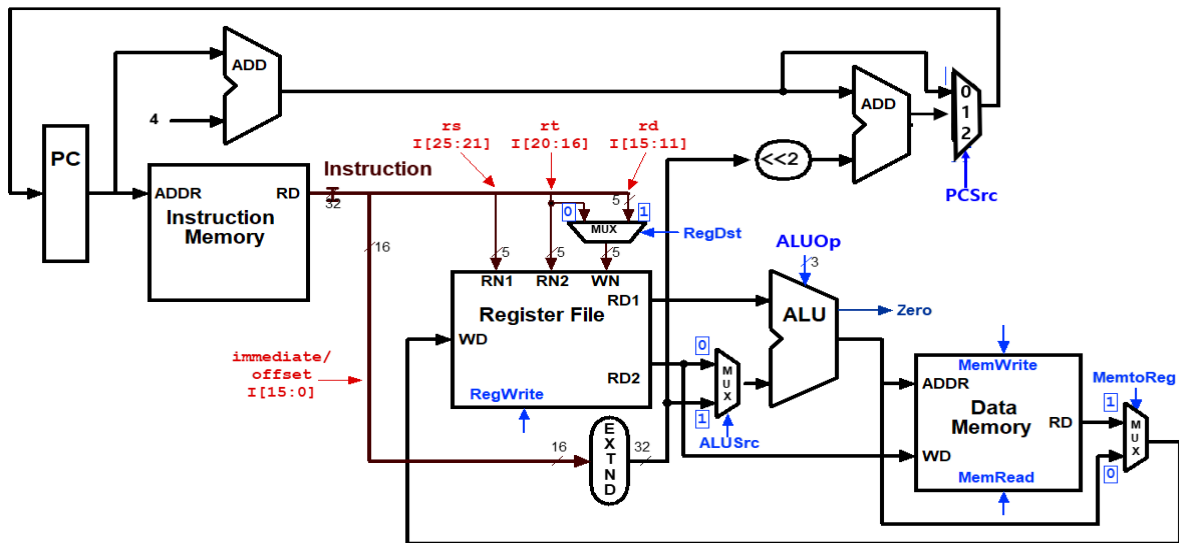


Problem 1) (20 Pts) Answer the questions for a single-cycle datapath.

- (a) The single-cycle datapath from lecture appears below. Clearly mark all paths that are active during the execution of lw instruction (lw 실행을 위하여 요구되는 선을 붉게 표시하시오).



On the table below, write (next to the signal's name) values of all non-0 control signals required for the lw instruction (ALUOp: Add = 010, Sub = 110).

RegDst	ALUSrc	MemtoReg	RegWrite	MemRead	MemWrite	ALUOp	PCSrc

Problem 2) (20 Pts) When the execution time in each of the following components is given as follows, answer the questions (각 컴포넌트에서의 실행시간이 아래 표와 같을 때 물음에 답하시오).

Functional Unit	Time delay
Memory	5 ns
ALU	4 ns
Register File	3 ns

Hint) **A clock cycle time** is determined by the instruction that has the longest execution time in single-cycle datapath, and is determined by the component in which the instruction has the longest execution time (클럭 사이클 타임은 싱글 사이클 데이터패스에서는 가장 긴 실행시간을 갖는 명령어에 의해 결정되고 멀티사이클 데이터패스에서는 가장 긴 실행시간을 갖는 컴포넌트에 의해서 결정된다).

- (a) What is the clock cycle time in a single-cycle datapath?

- (b) What is the clock cycle time in a multiple-cycle datapath?

- (c) Compute time to perform the following instructions in the table below: lw, sw, add, beq.

	Time to perform the instruction:	
Instruction	In single-cycle (ns)	In multi-cycle (ns)
lw		
sw		
add		

beq		
-----	--	--

- (d) How much faster than a 1GHz single-cycle MIPS processor would a 3.0GHz Pentium4 x86 processor be if it achieved a CPI of 0.5 on a workload? (번역: CPI = 0.5인 3GHz 펜티엄 프로세서는 1GHz 싱글 사이클 프로세서 (즉, CPI = 1.0)에 몇배나 빠른지를 계산하시오)

Problem 4) (20 Pts) Here is the final pipelined datapath that we discussed in class.

- Forwarding is performed from the EX/MEM and MEM/WB latches to the ALU inputs. The control equation for the Rs register input to the ALU is shown at the bottom of the page. The Rt input is similar.
- Branches are assumed to be not taken.
- A hazard unit inserts stalls for lw instructions.

- 포워딩은 EX/MEM과 MEM/WB 레지스터로부터 ALU입력단자로 이루어진다. ALU에 대한 Rs 레지스터 입력 (즉, A단자)을 선택하는 제어방정식은 아래와 같으며, Rt레지스터 입력은 유사함.
- 브랜치는 일어나지 않는 것으로 가정한다.
- Hazard 유닛은 lw 명령에 대하여 stall을 삽입한다.

Selection Equation for A (A를 위한 제어방정식)

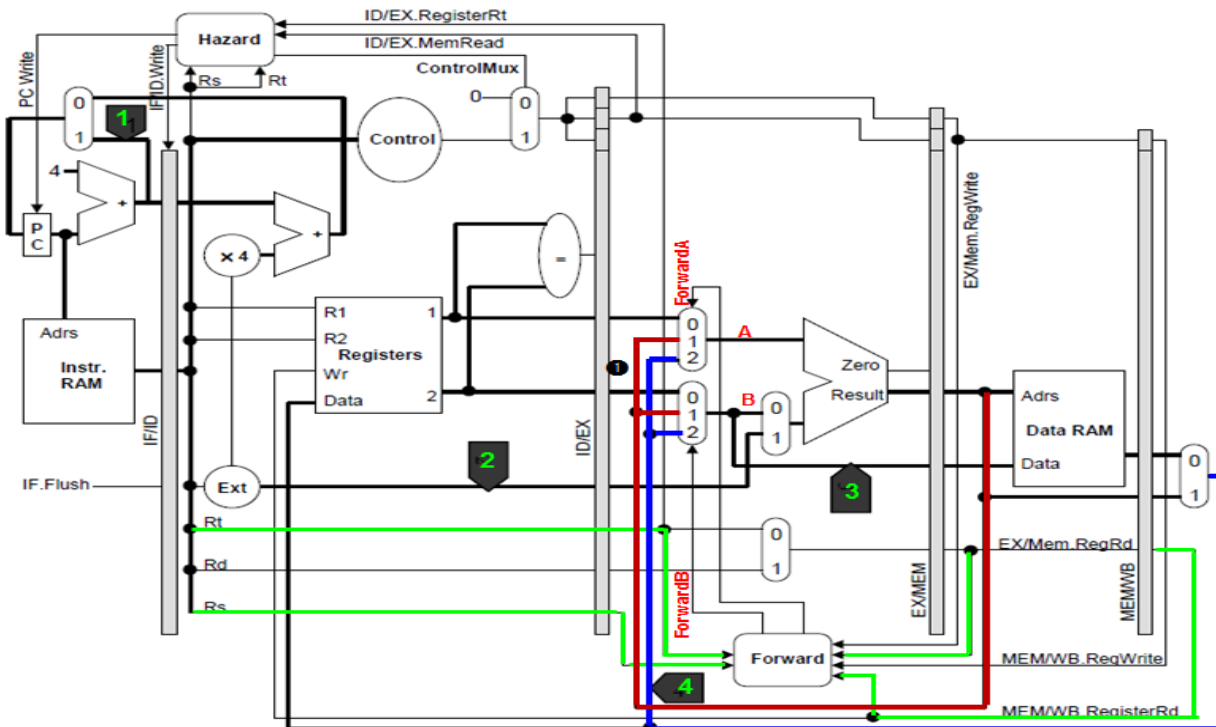
if ((EX/MEM.RegWrite == 1) and (EX/MEM.RegisterRd == ID/EX.RegisterRs)) then

ForwardA = 1

else if ((MEM/WB.RegWrite == 1) and (MEM/WB.RegisterRd == D/EX.RegisterRs))

then

ForwardA = 2



- (a) Pipelining primarily improves one of the followings. Select one. (10 Pts)

파이프라인을 사용하여 주로 개선되는 것을 선택하시오 (동그라미).

(1) Instruction latency (2) Instruction throughput

(b) Describe the situations that require a load stall.

load stall은 어떤 경우에 발생하는지 그 상황을 설명하시오..

(c) Complete the control equation in parenthesis for load stalls using the hint below.

Load stall에 대하여 아래 힌트를 사용하여 괄호 안에 프로그램을 완성하시오.

If () then

PC_Write = 0, IF/ID.Write = 0, and ControlMux = 0; //insert bubble in load stall

*Note: by default, PC_Write, IF/ID.Write, and ControlMux are 1.

Hint: In the following instruction sequence, one bubble between lw and add must be inserted at the ID stage.

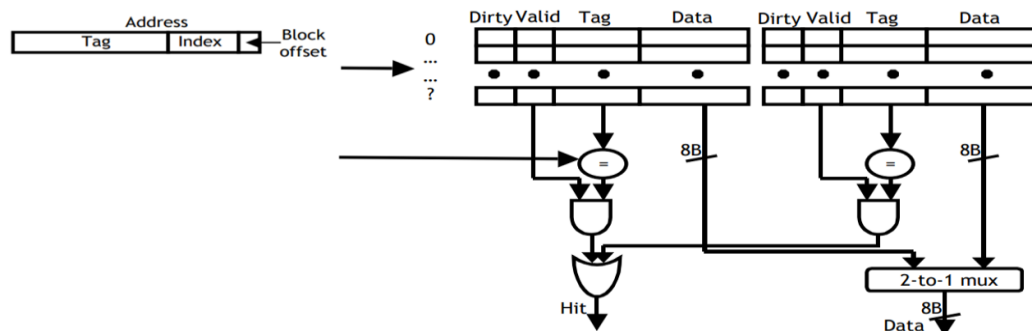
lw \$4, 0(\$7);

add \$3, \$4, \$5

Thus, use the following conditions to complete the above control equation (다음 조건들과 AND/OR을 사용하여 위 빈칸의 프로그램을 완성하시오):

- ID/EX.MemRead == 1
- ID/EX.Rt == IF/ID.Rs
- ID/EX.Rt == IF/ID.Rt

Problem 5) (20 Pts) Answer the questions for the following cache.



(a) Consider the cache drawn above in a processor that uses 32-bit addresses for a byte-addressable memory. If the cache uses 10-bit indices, how big (in bits) are the cache tags? (32비트 주소를 사용하는 프로세서에서 index가 10비트이면 tag는 몇 비트인가?)

(b) How much data can the cache hold? (캐쉬는 얼마나 많은 데이터를 저장할 수 있는가?) KB 단위로 나타내시오 (힌트: 위 그림을 보면 2-way set associative cache를 사용함을 알 수 있으므로 k비트 인덱스를 사용하면 전체 ($2^k \times 2$)블락을 사용하고 있음을 알 수 있다).

- (c) Given a direct-mapped cache with 4 blocks of 8 bytes, which of the following byte accesses hit? Also indicate whether the hit is because of spatial locality or temporal locality. (4개의 블록, 각 블록은 8바이트를 사용하는 direct-mapped 캐쉬 구조에서 주어진 어드레스를 액세스할 때 hit인지 miss인지를 기입하고 hit가 발생하면 spatial locality에 의한 것인지 temporal locality에 의한 것인지를 기입하십시오. (힌트: 어떤 주소 a10abc를 읽으면 해당 주소가 캐쉬에 없으면 블록 주소 (index) a10에 해당하는 블록 즉, a10000, a10001, a10010, a10011, ..., a10111에 해당하는 8개의 데이터를 한 번에 읽음)

Address (= 6bits)

Tag = 1 bit	Index = 2 bits	Block Offset = 3 bits
-------------	----------------	-----------------------

Address (binary)	Hit/Miss	Reason for hit: Spatial or Temporal?
000000	Miss	-
010101	Miss	-
111111	Miss	-
010000		
011111		
000100		
111111		
110111		
000111		
000111		
011100		

- (c) Given a direct-mapped cache with 4 blocks of 8 bytes, which of the following byte accesses hit? Also indicate whether the hit is because of spatial locality or temporal locality. (4개의 블록, 각 블록은 8바이트를 사용하는 direct-mapped 캐쉬 구조에서 주어진 어드레스를 액세스할 때 hit인지 miss인지를 기입하고 hit가 발생하면 spatial locality에 의한 것인지 temporal locality에 의한 것인지를 기입하십시오. (힌트: 어떤 주소 a10abc를 읽으면 해당 주소가 캐쉬에 없으면 블록 주소 (index) a10에 해당하는 블록 즉, a10000, a10001, a10010, a10011, ..., a10111에 해당하는 8개의 데이터를 한 번에 읽음)

Address (= 6bits)

Tag = 1 bit	Index = 2 bits	Block Offset = 3 bits
-------------	----------------	-----------------------

Address (binary)	Hit/Miss	Reason for hit: Spatial or Temporal?
000000	Miss	-
010101	Miss	-
111111	Miss	-
010000	Hit	
011111	Miss	
000100	Hit	
111111		
110111	Miss	
000111	Hit	
000111	Hit	
011100		

0 00
0 10
1 11
0 10
0 11
0 00
1 11
1 10
0 00
0 00
0 11