Feb. 4, 2009

COMPUTER ENGINEERING DEPARTMENT

ICS 233

COMPUTER ARCHITECTURE & ASSEMBLY LANGUAGE

Final Exam

First Semester (081)

Time: 7:30-10:30 AM

Student Name : _KEY_____

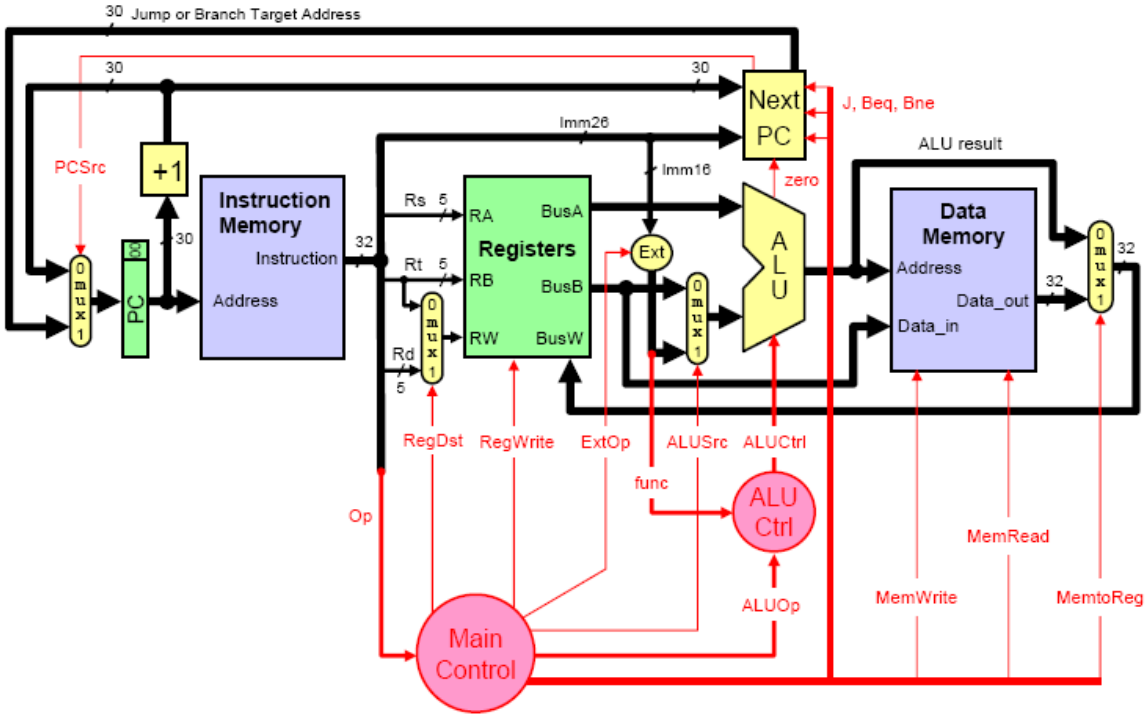
Student ID. : _____

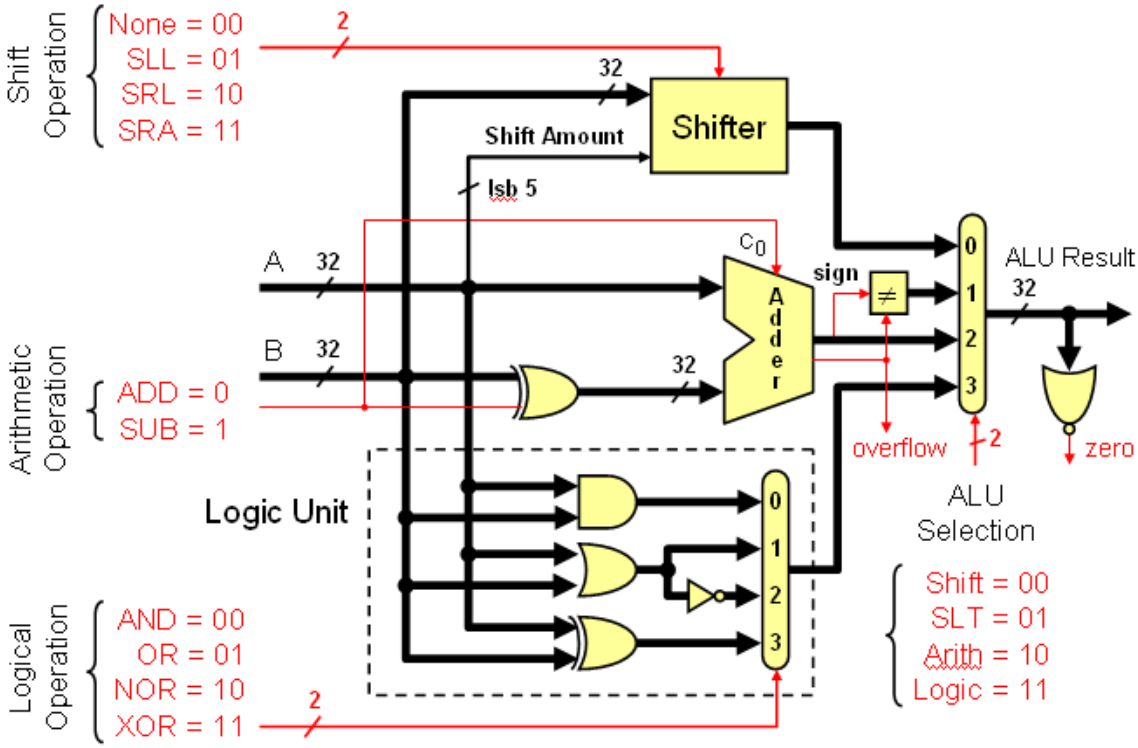
Question	Max Points	Score
Q1	30	
Q2	14	
Q3	15	
Q4	10	
Q5	15	
Q6	16	
Total	100	

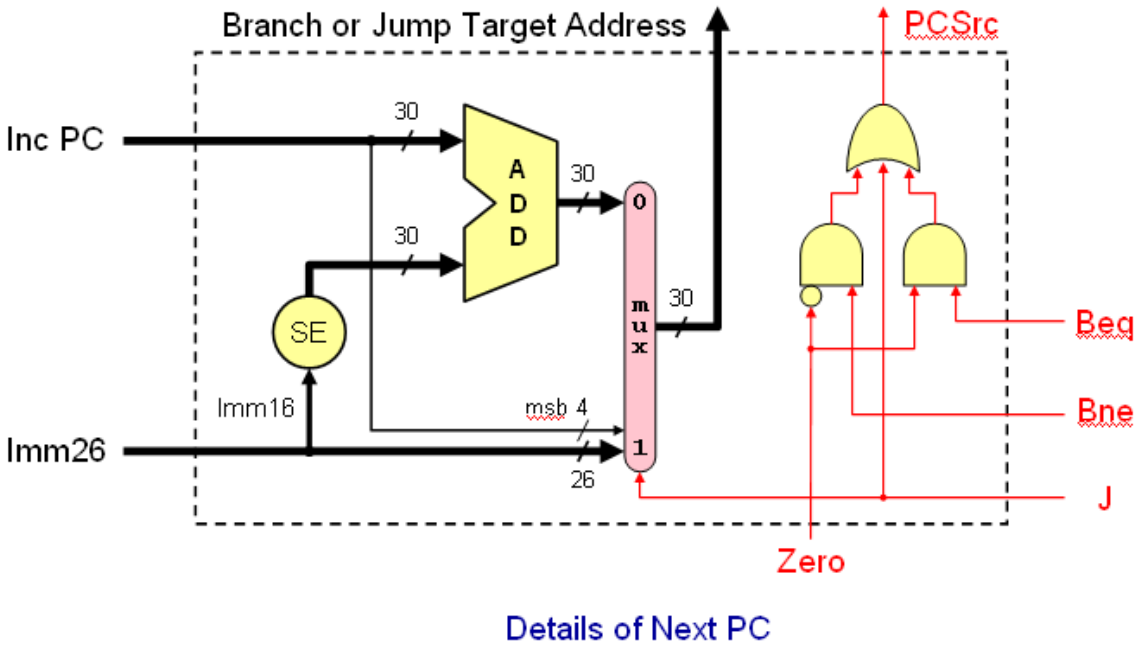
Dr. Aiman El-Maleh

[30 Points]

(Q1) Consider the single-cycle datapath and control given below along with ALU and Next PC blocks design for the MIPS processor implementing a subset of the instruction set:







Details of Next PC

(i) Show the control signals generated for the execution of the following instructions by filling the table given below:

Op	RegDst	RegWrite	ExtOp	ALUSrc	ALUOp	Beq	Bne	J	MemRead	MemWrite	MemtoReg
R-type	1 = Rd	1	x	0=BusB	R-type	0	0	0	0	0	0
ori	0 = Rt	1	0=zero	1=Imm	OR	0	0	0	0	0	0
sw	x	0	1=sign	1=Imm	ADD	0	0	0	0	1	x
bne	x	0	x	0=BusB	SUB	0	1	0	0	0	x
j	x	0	x	x	x	0	0	1	0	0	x

The format of these instructions is given below for your reference:

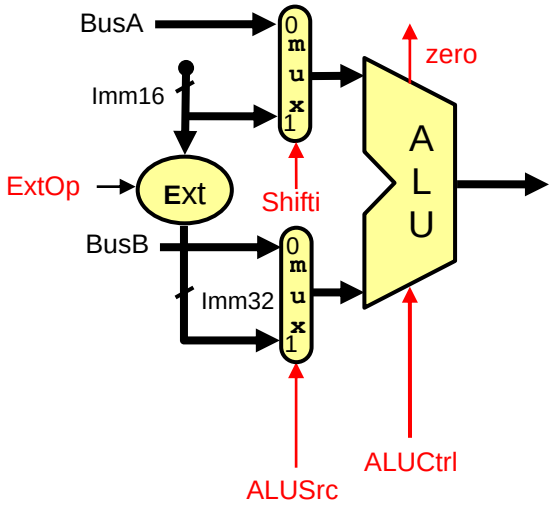
Instruction		Meaning	Format					
add	rd, rs, rt	rd = rs + rt	$Op^6 = 0$	rs^5	rt^5	rd^5	0	0x20
ori	rt, rs, imm^{16}	rt = rs | imm^{16}	0x0d	rs^5	rt^5	imm^{16}		
sw	rt, imm^{16}(rs)	MEM[rs+imm16]=rt	0x2b	rs^5	rt^5	imm^{16}		
bne	rs, rt, label	branch if (rs != rt)	0x05	rs^5	rt^5	imm^{16}		
j	label	Jump to label	0x02	imm^{26}				

(ii) We wish to add the following instructions to the MIPS single-cycle datapath. Add any necessary datapath modifications and control signals needed for the implementation of these instructions. Show only the **modified** and **added** components to the datapath. Show the values of the control signals to control the execution of each instruction.

a. sra

Instruction	Meaning	Format					
sra rd, rt, imm^5	rd= rt>>imm^{16}	$Op^6 = 0$	0	rt^5	rd^5	Imm^5	f^5=3

For the sra instruction, examining the ALU one can see that the shift amont is coming through the A-input of the ALU and the operand to be shifted comes through the B input of the ALU. Thus, we need-to add a MUX on the A-input to select between the output of a register and the immediate values. This MUX needs to select only between the least significant 5 bits. The modified part in the datapath is shown below:



The values of the control signals to control the execution of this instruction are given below:

Op	Shifti	RegDst	RegWrite	ExtOp	ALUSrc	ALUOp	Beq	Bne	J	MemRead	MemWrite	MemtoReg
sra	1	1 = Rd	1	x	0=BusB	sra	0	0	0	0	0	0

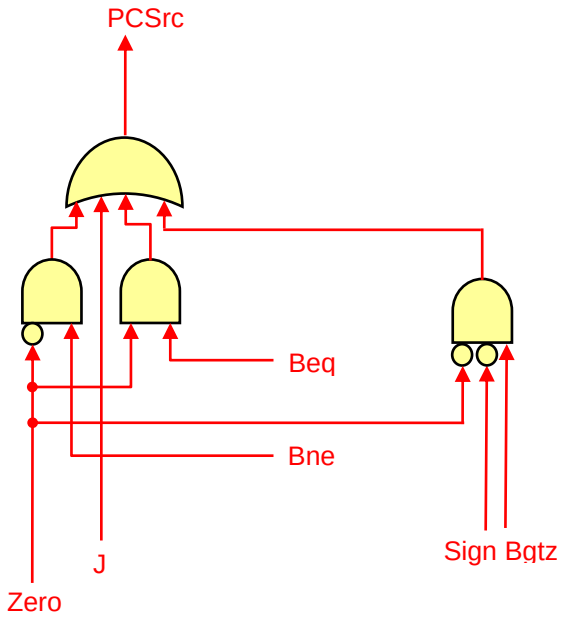
b. bgtz

Instruction	Meaning	Format			
bgtz rs, label	branch if (rs>0)	$Op^6 = 7$	rs^5	0	imm^{16}

Since the first source operand specified by RS comes on BusA and the second operand which is the Zero register specified by the RT filed comes on BusB, all we need is to get the operand on BusA to appear at the output of the ALU as we just need to check the sign bit (i.e. most significant bit of the result).

Performing an addition, subtraction, xoring, oring operations will work. Let us assume that we will do an ALU addition operation.

To check that the result is greater than 0, we need to check that the sign bit is 0 and that the result is not equal to zero. Thus, the changes needed to be done are in the NextPC block as shown below:



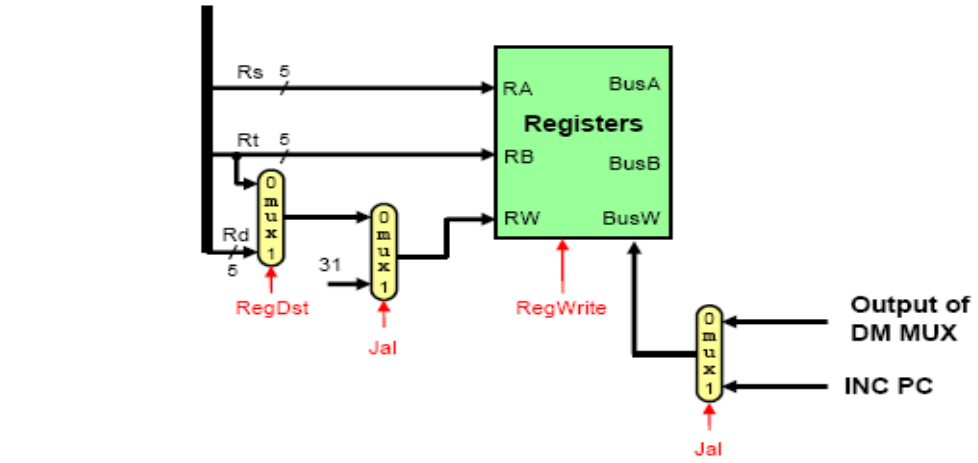
The values of the control signals to control the execution of this instruction are given below:

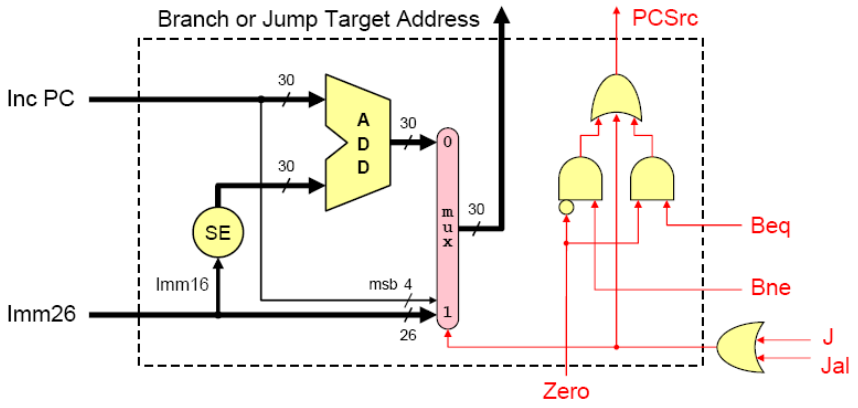
Op	RegDst	RegWrite	ExtOp	ALUSrc	ALUOp	Bgtz	Beq	Bne	J	MemRead	MemWrite	MemtoReg
bgtz	x	0	x	0= BusB	ADD	1	0	0	0	0	0	x

c. jal

Instruction		Meaning	Format	
jal	label	$31=PC+4, jump	$op^6 = 3$	imm^{26}

This instruction is similar to the jump instruction (J) with the difference that register $31 should be loaded with the incremented PC value. Thus, we need to add a MUX at the input of RW input to the register file to select the value 31 when executing this instruction. We also need to add a MUX at the input of BusW in the register file to select the incremented PC value to be loaded instead of the value coming from the output of the data memory MUX. In addition, we need to make changes to the NextPC block to perform the same operation needed by the J instruction for Jal instruction. These changes are shown below.





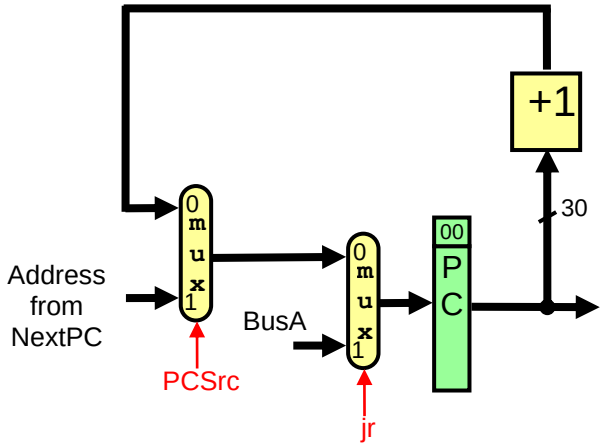
The values of the control signals to control the execution of this instruction are given below:

Op	RegDst	RegWrite	ExtOp	ALUSrc	ALUOp	Jal	Beq	Bne	J	MemRead	MemWrite	MemtoReg
jal	x	1	x	x	x	1	0	0	0	0	0	x

d. jr

Instruction		Meaning	Format					
jr	rs	PC=rs	$op^6 = 0$	rs^5	0	0	0	8

For this instruction, the changes required in the datapath to implement it is to load the PC from BusA which is driven by the RS field. Thus we need to add a MUX to select the target address to be loaded in the PC either from the output of the MUX choosing between the address from NextPC block and incremented PC or from BusA. The required changes are shown below:



The control signals for this instruction are:

Op	RegDst	RegWrite	ExtOp	ALUSrc	ALUOp	jr	Beq	Bne	J	MemRead	MemWrite	MemtoReg
jr	X	0	x	x	x	1	0	0	0	0	0	x

(iii) Assume that the propagation delays for the major components used in the datapath are as follows:

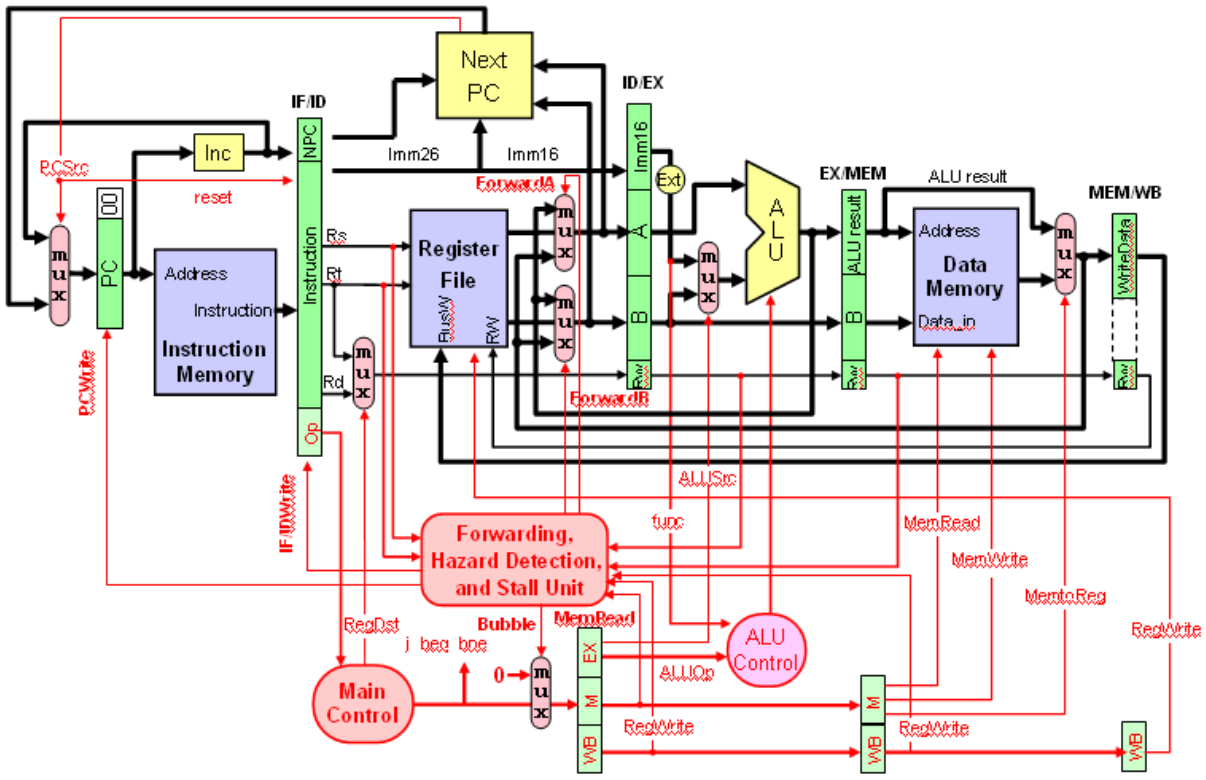
- Instruction and data memories: 100 ps
- ALU and adders: 40 ps
- Register file access (read or write): 10 ps
- Main control: 15 ps
- ALU control: 15 ps

Ignore the delays in the multiplexers, PC access, extension logic, and wires. What is the cycle time for the single-cycle datapath given above?

Cycle Time = IM + max(Main Control+ALU Control, Register Reading) +
ALU + DM + Register Writing
= 100 ps + 30 ps + 40 ps + 100 + 10 ps = 280 ps

[14 Points]

(Q2) Consider the pipelined MIPS processor design given below:



(i) Show the control signals that will be used for forwarding along with their conditions. In case both forwarding conditions from the ALU and Memory Mux are met, which one should be allowed to forward?

Control Signal	Explanation
ForwardA = 00	First ALU operand comes from the register file
ForwardA = 01	Forwarded from the previous ALU result
ForwardA = 10	Forwarded from data memory or 2nd previous ALU result
ForwardB = 00	Second ALU operand comes from the register file
ForwardB = 01	Forwarded from the previous ALU result
ForwardB = 10	Forwarded from data memory or 2nd previous ALU result

Forwarding Conditions:

if (IF/ID.Rs == ID/EX.Rw and ID/EX.Rw ≠ 0 and ID/EX.RegWrite)
ForwardA = 01
elseif (IF/ID.Rs == EX/MEM.Rw and EX/MEM.Rw ≠ 0 and EX/MEM.RegWrite)
ForwardA = 10
else ForwardA = 00

if (IF/ID.Rt == ID/EX.Rw and ID/EX.Rw ≠ 0 and ID/EX.RegWrite)
ForwardB = 01
elseif (IF/ID.Rt == EX/MEM.Rw and EX/MEM.Rw ≠ 0and EX/MEM.RegWrite)
ForwardB = 10
else ForwardB = 00

In case both forwarding conditions from the ALU and Memory Mux are met, forwarding should be taken from the ALU as it is the latest instruction to execute before the current instruction.

Condition for Stalling the pipeline:

1. Stalling the Pipeline due to Load Instruction:

if ((ID/EX.MemRead == 1) and (ID/EX.Rw ≠ 0) and
((ID/EX.Rw == IF/ID.Rs) or (ID/EX.Rw == IF/ID.Rt))) Stall

Stall means that the signals PCWrite=0 and IF/IDWrite=0, which will freeze the content of PC and IF/ID registers and bubble=1 which will introduce a bubble in the ID/EX register by setting the control signals to 0.

2. Stalling the Pipeline due to taken branch Instruction:

Also, when PCSrc=1, reset=1 and the content of IF/ID register will be reset to 0 to make the fetched instruction a NOP.

[15 Points]

(Q3) Consider the code given below:

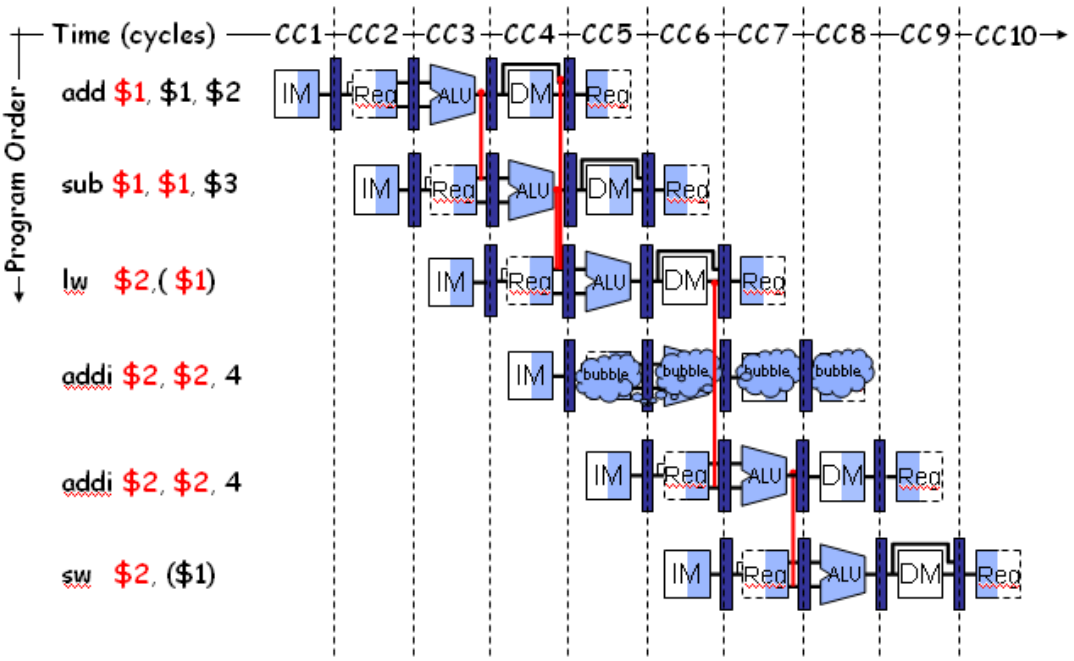
```
add  $1, $1, $2
sub  $1, $1, $3
lw   $2, ($1)
addi $2, $2, 4
sw   $2, ($1)
```

(i) Identify all the **RAW** data dependencies in the above code. Which dependencies are data hazards that will be resolved by forwarding? Which dependencies are data hazards that will cause a stall?

RAW dependencies:

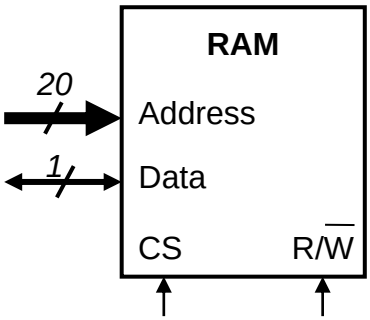
add $1, $1, $2	**and**	**sub $1, $1, $3**	**(forwarding)**
add $1, $1, $2	**and**	**lw $2, ($1)**	**(forwarding)**
sub $1, $1, $3	**and**	**lw $2, ($1)**	**(forwarding)**
lw $2, ($1)	**and**	**addi $2, $2, 4**	**(stall 1 cycle & forwarding)**
addi $2, $2, 4	**and**	**sw $2, ($1)**	**(forwarding)**

(ii) Using a multiple-clock-cycle graphical representation, show the instruction execution across the pipeline including forwarding paths and stalled cycles if any. How many clock cycles will be needed to execute the instructions?

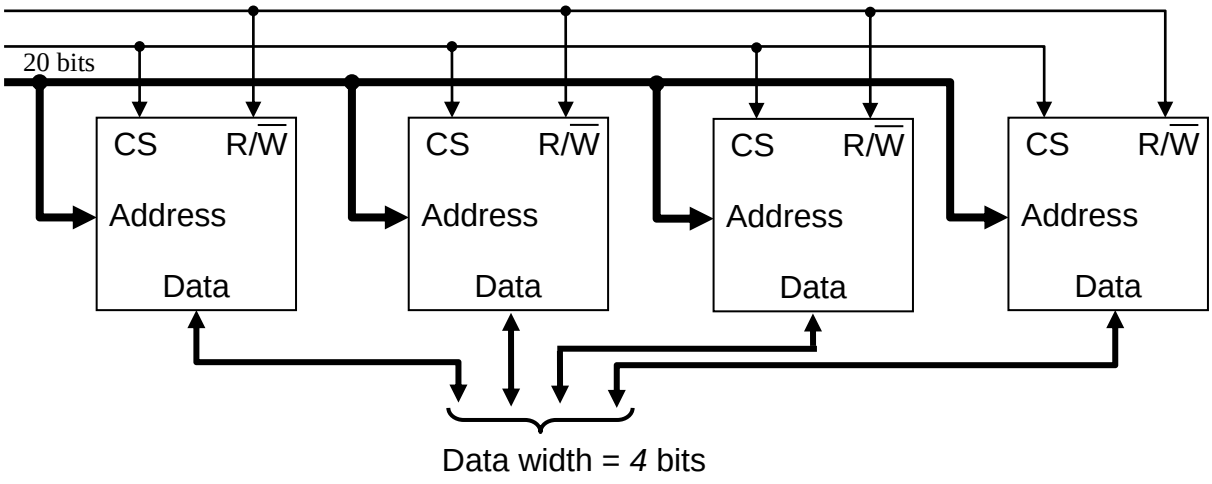


[10 Points]

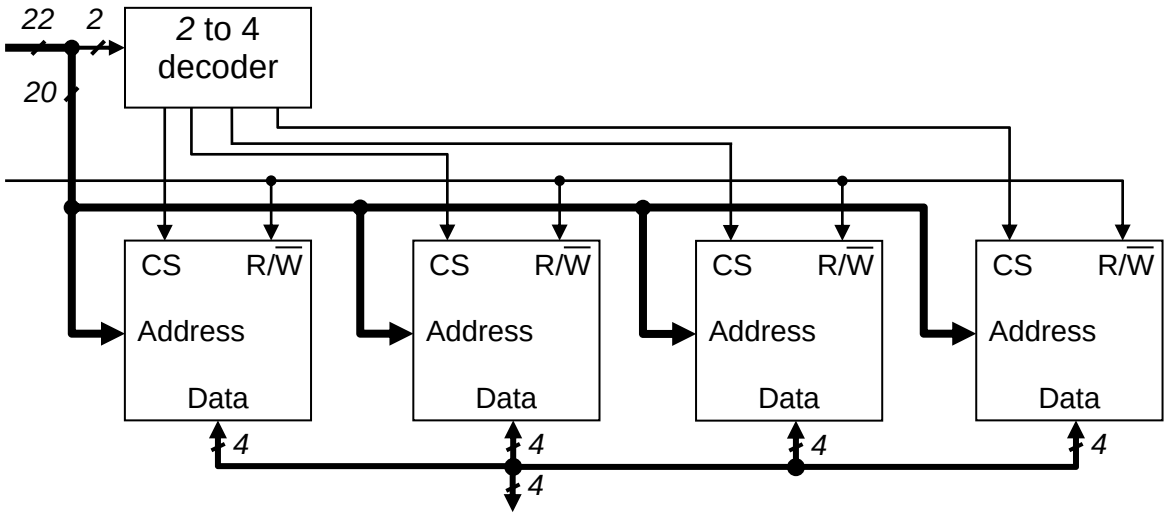
(Q4) Given a 1M x 1 memory block as shown below. Use this block to implement a 4M x 4 memory block.



First, we design a block of RAM of size 1M x 4 as follows.



Then, we design a block of RAM of size 4M x 4 using the above block as follows.



[15 Points]

(Q5) Assume that you have a 32-bit address and a cache with **4K byte data size** (i.e. not including tag and valid bits).

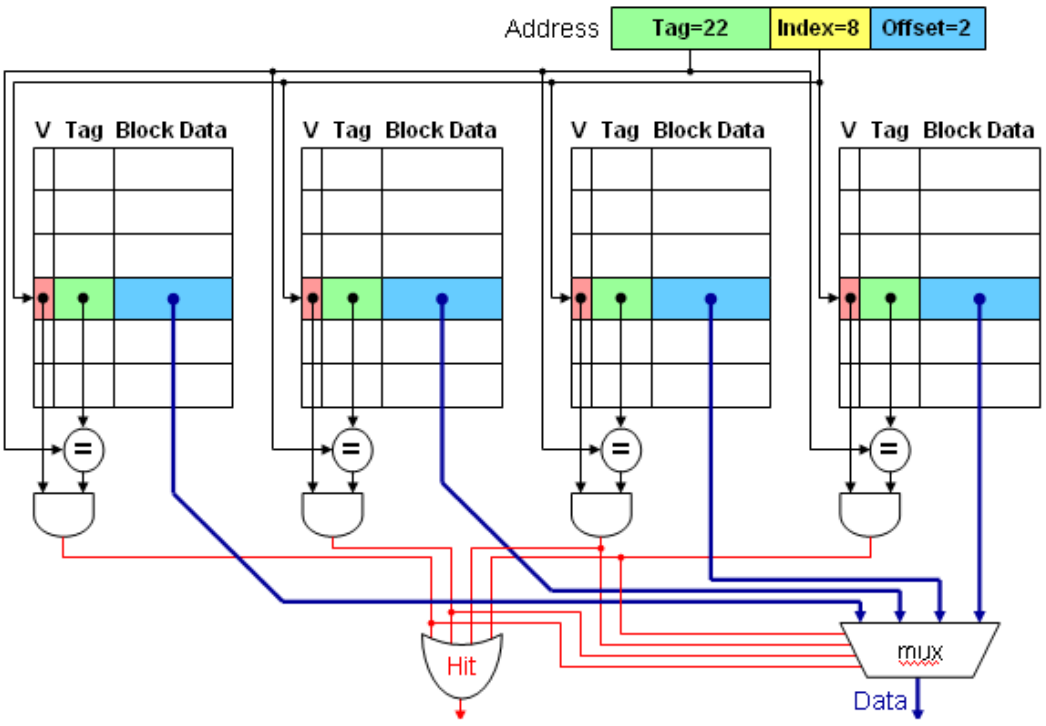
(i) Assuming that the cache is organized as **direct-mapped** with **4-byte block size**, determine the number of bits in the offset, index and tag fields.

Offset = $\log_2$ (block size) = $\log_2$ 4 = 2 bits
Index = $\log_2$ (# locations) = $\log_2$ (4K/4) = $\log_2$ 1K = 10 bits
Tag= 32- 12 = 20 bits

(ii) Assuming that the cache is organized as **four-way set associative** with **4-byte block size**, determine the number of bits in the offset, index and tag fields.

Offset = $\log_2$ (block size) = $\log_2$ 4 = 2 bits
Index = $\log_2$ (# locations) = $\log_2$ (4K/4*4) = $\log_2$ (256) = 8 bits
Tag= 32- 10 = 22 bits

(iii) Show the organization of the **cache organized as four-way set associative** with **4-byte block size**.



[16 Points]

(Q6) A processor runs at 3 GHz and has a CPI=2 for a perfect cache (i.e. without including the stall cycles due to cache misses). Assume that load and store instructions are 25% of the instructions. The processor has an I-cache with a 5% miss rate and a D-cache with 2.5% miss rate. The hit time is 1 clock cycle. Assume that the time required to transfer a block of data from the RAM to the cache, i.e. miss penalty, is 40 ns.

(i) What is the average memory access time for instruction access in clock cycles?

Miss penalty in clock cycles = $40 * 10^{-9} * 3 * 10^{9} = 120$
AMAT = hit time + miss rate * miss penalty = 1 + 0.05 * 120 = 7

(ii) What is the average memory access time for data access in clock cycles?

AMAT = 1 + 0.025 * 120 = 4

(iii) What is the number of stall cycles per instruction and the overall CPI?

Number of stall cycles per instruction = 1*0.05*120+0.25*0.025*120 = 6.75
Overall CPI = 2 + 6.75 = 8.75

(iv) A new technology is proposed that can make the processor run at 4 GHz. The only impact of this technology is that the cache size has to be decreased to keep a hit time of one clock cycle. Assume that the time required to transfer a block of data from the RAM to the cache is reduced to 30 ns. What should be the number of stalls per instruction in the new processor to be faster by a factor of at least 1.2. What should be the instruction miss rate in the new technology if the data miss rate is 4%.

Miss penalty in clock cycles = $30 * 10^{-9} * 4 * 10^{9} = 120$

Speedup = $8.75 * 4 * 10^{9} / CPI * 3 * 10^{9} = 1.2$
CPI = $8.75 * 4 * 10^{9} / 1.2 * 3 * 10^{9} = 9.72$
Thus, number of stalls per instruction should be less than or equal to 9.72-2=7.72.
Number of stall cycles per instruction <=7.72
IMR * 120 + 0.25 * 0.04 * 120<=7.72
IMR * 120+1.2<=7.72
IMR<=6.52/120
IMR<= 5.43%