Quiz 5
Show the working in rough sheet for all questions

Question **1**
Not yet answered
Marked out of 2.00
Flag question
Edit question

Assume we have a single-level, 4 KiloByte direct-mapped L1 cache with 4-byte blocks. Address is 32-bits. How many tag bits exist?

Answer:

Question **2**
Not yet answered
Marked out of 4.00
Flag question
Edit question

Assume that we are running the following instructions on a 5 stage MIPS pipeline.

Assume that we can write and read registers in the same clock cycle.

ADD t4, t2, t3

SW t4, 0(t2)

BEQ t1, t4, target

Which of the following statements are true?

Select one or more:

- a. No data forwarding required to Instruction 3
- b. With forwarding, data needs to be forwarded from MEM of Instruction1 to MEM of Instruction3
- c. With forwarding, data needs to be forwarded from EX of Instruction1 to MEM of Instruction2
- d. With forwarding, data needs to be forwarded from MEM of Instruction1 to EX of Instruction3
- e. No data forwarding required between any instructions
- f. With forwarding, data needs to be forwarded from EX of Instruction1 to EX of Instruction2
- g. With forwarding, data needs to be forwarded from EX of Instruction2 to EX of Instruction3

Show the working in rough sheet

Question 3

Not yet answered

Marked out of 3.00

Flag question

Edit question

Assume that we have a 6 stage processor pipeline:

IF, ID, EX1, EX2, MEM, WB

Forwarding is enabled.

Branch is resolved in EX2

Processor keeps fetching PC+4 always.

We execute the following set of instructions:

ADD t1, t2, t0

BEQ t1, t2, target

LW t2, 0(t0)

SUB t1, t2, t3

ADD t1, t2, t0

..

..

target: SUB t1, t2, t3

At some point in time, the processor finds out that t1= t2.

Does the processor need to flush the pipeline? If yes, how many instructions does it need to flush. If no, write 0 as the answer

Answer:

Show the working in rough sheet

Q4:
(10 marks)

We have a six stage processor pipeline: IF, ID, EX, Mem1, Mem2, WB.

The data fetched from memory is available only after the Mem2 stage (for Loads). Branches are resolved in the EX stage. By default, the next instruction (PC+4) is always fetched.
The processor supports forwarding.
There are no structural hazards
The following instructions are executed on this processor in the same order.
LW R1, 0(R2)
BEQ R1, R2, target
ADD R2, R3, R4
Instr 6
Instr 7
Instr 8
Instr 9
Instr 10
...
...

target: LW R1, 0(R2)
...

Question:
a. Write the pipeline diagram for these instructions assuming that R1= R2 when BEQ executes. Assume that Instructions 6 to 10 are independent of each other with no hazards (5 marks)
b. Indicate which register is forwarded, Indicate the from and to instruction. Indicate the from and to pipeline stage for forwarding (2 marks)
c. If it needs a pipeline flush, how many instructions need to be flushed and which instructions are flushed (3 marks)

Question **5**

Not yet answered

Marked out of 4.00

Flag question

Edit question

The following set of instructions work well on a non-pipelined processor. But, when we create a 5-stage pipeline, register t1 gets the wrong answer. How do you fix this error?

Assume that we CANNOT write and read registers in the same clock cycle. (We cannot write in first half of the cycle, and read in the second half)

ADDI t0 t7 1

ADD t2 t0 t3

SUB t1 t2 t3

Select one or more:

- [ ] a. Insert 2 NOPs (No-operation instruction) after Instruction 1, and after Instruction 2
- [ ] b. Forward from execute to ID stage:Instruction 2 to 3 only
- [ ] c. Insert 1 NOP (No-operation instruction) after Instruction 2 only
- [ ] d. Forward from execute to ID stage: Instruction 1 to 2 only
- [ ] e. Forward from execute to execute stage: Instruction 1 to 2, Instruction 2 to 3
- [ ] f. Insert 3 NOPs (No-operation instruction) after Instruction 1, and after Instruction 2

Question **6**

Not yet answered

Marked out of 3.00

Flag question

Edit question

We have a 5 stage MIPS processor pipeline with forwarding enabled.

Fill up the blank such that, the instruction will create one stall between Instruction 1 and 2

Instruction 1: _____

ADD t2, t4, t0

SUB t2, t2, t0

Select one or more:

- [ ] a. ADD t1, t4, t0
- [ ] b. BEQ t4, t0, loop
- [ ] c. LW t0, 0(t4)
- [ ] d. LW t4, 0(t0)
- [ ] e. SW t4, 0(t0)

Question **7**

Not yet answered

Marked out of 3.00

Flag question

Edit question

Assume that we have a 4-stage pipeline: IF, ID, EX/MEM, WB. All stages take 1 cycle to execute.

Assume that we are **NOT** allowed to write/read the registers in the same clock cycle.

Does any hazard exist between the following set of instructions? If yes, identify the type of hazard.

ADDI t0 x0 1

SUB t1 t5 t3

SW t2 0(t5)

BEQ t3, t5, target

..

..

Select one or more:

- [ ] a. Control hazard
- [ ] b. No hazard
- [ ] c. Structural hazard
- [ ] d. Data hazard

Question 8

Not yet answered

Marked out of 3.00

Flag question

Edit question

Address

Tag	Index	Offset

	Tag	Way 0		Tag	Way 1	
0	0x000004	0x412C1232	0x41111234	0x000C04	0x11241239	0x412C1111
1	0x000200	0x002C12AB	0x00111235	0x000240	0x00011238	0x412C1000
2	0xAA4004	0x41AC1200	0x41100237	0xAAA404	0x11541125	0x412C1133
3	0x500001	0x002A3A38	0x001122AA	0x500900	0x00011222	0x412C1000
4	0xA57000	0xAAAC1232	0x41111234	0xA56880	0x11111230	0x412C1BB
5	0x000200	0xCA1C1238	0xCC111239	0x000300	0x03010023	0x412C10A4
31						

We have a 2 way set associative cache. Each way has 2 blocks of data, each 4 bytes of data.

The address is 32 bits wide.

As part of the address, we get an index = 3. The address is a cache hit and it returns data byte 41. What is the 32 bit address in this case? Write it in hexadecimal format (0x.....)

Answer:

Q9: (2 marks)

Address

Tag	Index	Offset

	Tag	Way 0		Tag	Way 1	
0	0x000004	0x412C1232	0x41111234	0x000C04	0x11241239	0x412C1111
1	0x000200	0x002C12AB	0x00111235	0x000240	0x00011238	0x412C1000
2	0xAA4004	0x41AC1200	0x41100237	0xAAA404	0x11541125	0x412C1133
3	0x500001	0x002A3A38	0x001122AA	0x500900	0x00011222	0x412C1000
4	0xA57000	0xAAAC1232	0x41111234	0xA56880	0x11111230	0x412C1BB
5	0x000200	0xCA1C1238	0xCC111239	0x000300	0x03010023	0x412C10A4
31						

The address is 32 bits wide.

We want to convert this 2 way set associative cache into a fully associative cache. How many ways will the fully associative cache contain?

Answer:

Q10: (2 marks)

	Way0	Way1	Way 2	Way3
0	M[0]	M[4]	M[8]	M[10]
7				

Assume the 4 way set associative cache. Tag is not shown in the cache. The Sequence of memory accesses are shown below in the same order. Addresses 0,4,8,10 and so on are in hexadecimal.

M[0]
M[4]
M[8]
M[10]
M[8]
M[8]
M[0]

At this point, if the processor issues a load for memory address 0x18, which is, M[18] or M[11000], which of the following is true?

Let us assume a LRU replacement policy for this cache.

Select one or more:

- [ ] a. M[18] is a cache miss
- [ ] b. M[18] goes into Way 3
- [ ] c. M[18] replaces M[0] and goes into Way 0
- [ ] d. M[18] goes into Way 2
- [ ] e. M[18] replaces M[4] and goes into Way 1
- [ ] f. M[18] goes into a different index other than 0
- [ ] g. M[18] is a cache hit