Benha Faculty of Engineering
3rd Year Electronics
Examiner: Dr. Hatem ZAKARIA

Electrical Engineering Department
Final Exam: 15 January 2015
Time allowed: 3 Hours

Computer Organization (E1327)

Answer All Questions	No. of Questions: 5	No. of Pages: 3

Question (1) **(22 Marks)**

a. State if the following statements are (✓) or (×) and justify your answer

1. Pipelining decreases CPU instruction throughput but reduce the execution time of each individual instruction. (×)

   It does not reduce the individual instruction execution time.

2. Variable length instructions make the pipelining much easier. (×)
   Fixed length instructions make the pipelining easier.

3. Cache memory is better implemented using DRAM for its superior speed. (×)

   It is better implemented using SRAM.

4. If Computer A has a higher MIPS rating than computer B, then A is faster than B. (×)

   It is possible to have higher MIPS rating and worse execution time.

5. Data hazard means conflict resources (×)

   Structural hazard means conflict resources

6. On a read, the value returned by the cache depends on which blocks are in the cache. (×)

   It depends on the last value written to the same memory location

7. Allowing ALU and branch instructions to take fewer stages and complete earlier than other instructions does not improve the performance of a pipeline. (✓)

   Pipeline performance is related to throughput, not the latency of instructions

8. Increasing the depth of pipelining by splitting stages always improves performance. (×)

   Not always, at some point pipeline register delays become significant, and more bubbles or stall cycles must be introduced if the pipeline depth is increased

9. Utilizing faster processor results in an increase in the performance regardless of the memory speed. (×)

   The memory speed is a real limitation for overall CPU performance.

10. The higher the memory bandwidth, the larger the cache block size should be. (✓)
    If the memory bandwidth is high then a larger block size can be transferred in same amount of time

b. Illustrates the different instruction formats for the MIPS R3000 CPU explaining the purpose of each field of the instruction word?

Name	Fields						Comments
Field size	6 bits	5 bits	5 bits	5 bits	5 bits	6 bits	All MIPS instructions are 32 bits long
R-format	op	rs	rt	rd	shamt	funct	Arithmetic instruction format
I-format	op	rs	rt	address/immediate			Transfer, branch, imm. format
J-format	op	target address					Jump instruction format

Here is the meaning of each name of the fields in MIPS instructions:

- *op:* Basic operation of the instruction, traditionally called the opcode.
- *rs:* The first register source operand.
- *rt:* The second register source operand.
- *rd:* The register destination operand. It gets the result of the operation.
- *shamt:* Shift amount. (Section 2.6 explains shift instructions and this term; it will not be used until then, and hence the field contains zero in this section.)
- *funct:* Function. This field, often called the *function code,* selects the specific variant of the operation in the op field.

Question (2) **(18 Marks)**

a. Consider three different processors P1, P2, and P3 executing the same instruction set. P1 has a 3 GHz clock rate and a CPI of 1.5. P2 has a 2.5 GHz clock rate and a CPI of 1.0. P3 has a 4.0 GHz clock rate and has a CPI of 2.2.

1. Which processor has the highest performance expressed in instructions per second?

   Answer:

   performance of P1 (instructions/sec) $= 3 \times 10^9/1.5 = 2 \times 10^9$

   performance of P2 (instructions/sec) $= 2.5 \times 10^9/1.0 = 2.5 \times 10^9$

   performance of P3 (instructions/sec) $= 4 \times 10^9/2.2 = 1.8 \times 10^9$

2. If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.

   Answer:

   cycles(P1) $= 10 \times 3 \times 10^9 = 30 \times 10^9$ s

   cycles(P2) $= 10 \times 2.5 \times 10^9 = 25 \times 10^9$ s

   cycles(P3) $= 10 \times 4 \times 10^9 = 40 \times 10^9$ s

3. We are trying to reduce the execution time by 30% but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?

Answer:

No. instructions(P1) $= 30 \times 10^9/1.5 = 20 \times 10^9$

No. instructions(P2) $= 25 \times 10^9/1 = 25 \times 10^9$

No. instructions(P3) $= 40 \times 10^9/2.2 = 18.18 \times 10^9$

$CPI_{new} = CPI_{old} \times 1.2$, then CPI(P1) = 1.8, CPI(P2) = 1.2, CPI(P3) = 2.6

f = No. instr. × CPI/time, then

f(P1) $= 20 \times 10^9 \times 1.8/7 = 5.14$ GHz

f(P2) $= 25 \times 10^9 \times 1.2/7 = 4.28$ GHz

f(P1) $= 18.18 \times 10^9 \times 2.6/7 = 6.75$ GHz

b. Translate the following C code to MIPS. Assume that the variables f, g, h, i, and j are assigned to registers $s0, $s1, $s2, $s3, and $s4, respectively. Assume that the base address of the arrays A and B are in registers $s6 and $s7, respectively. Assume that the elements of the arrays A and B are 4-byte words: B[8] = A[i] + A[j].

Answer:

```
sll     $t0,    $s3,    2       #       $t0 = 4*i, offset of A[i]
sll     $t1,    $s4,    2       #       $t1 = 4*j, offset of A[j]
add     $t0,    $t0,    $s6     #       addressof A[i]
add     $t1,    $t1,    $s6     #       addressof A[j]
lw      $t0,    0($t0)          #       $t0 = A[i]
lw      $t1,    0($t1)          #       $t1 = A[j]
add     $t0,    $t1,    $t0     #       $t0 =A[i] + A[j]
addi    $t1,    $s7,    32      #       addressof B[8]
sw      $t0,    0($t1)          #       B[8]=A[i]+A[j]
```

c. Assume the following register contents: $t0 = 0xAAAAAAAA, $t1 = 0x12345678. For the register values shown above, what is the value of $t0, $t1 and $t2 for the following sequence of instructions?

```
sll $t2, $t0, 44
or $t2, $t2, $t1
```

Answer:

$t0 → 0xAAAAAAAA $t1 → 0x12345678 $t2 → 0xBABEFEF8

Question (3) **(18 Marks)**

a. What are pipeline hazards? Enumerate and briefly present the three types of hazards?

Answer:

Hazards: situations that would cause incorrect execution, if next instruction were launched during its designated clock cycle. Hazards complicate pipeline control and limit performance

1. *Structural hazards*: Caused by resource contention. Using same resource by two instructions during the same cycle
2. *Data hazards*: An instruction may compute a result needed by next instruction. Hardware can detect dependencies between instructions
3. *Control hazards*: Caused by instructions that change control flow (branches/jumps). Delays in changing the flow of control

b. Explain why WAR hazard cannot appear in the MIPS-5 stage pipelined architecture?

Answer:

This kind of hazards can't happen in MIPS 5-stages pipelined processor for the following reasons:

- All instructions take 5-stages, and
- Reads are always in stage number 2, and Writes are always in stage number 5.

c. Referring to the following sequence of instructions:

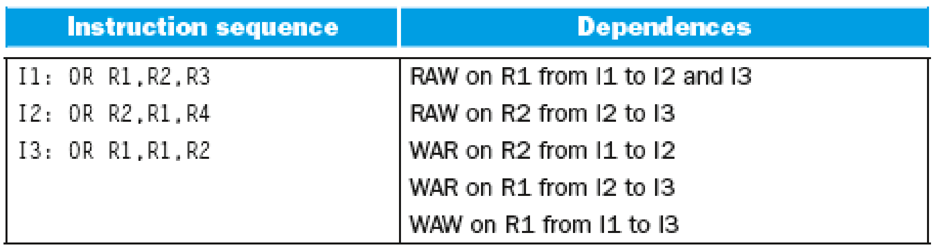
OR R1,R2,R3
OR R2,R1,R4
OR R1,R1,R2

Also, assume the following cycle times for each of the options related to forwarding:

Without Forwarding	With Full Forwarding	With ALU-ALU Forwarding Only
250ps	300ps	290ps

1. Indicate dependences and their type.

Answer:



Instruction sequence	Dependences
I1: OR R1,R2,R3	RAW on R1 from I1 to I2 and I3
I2: OR R2,R1,R4	RAW on R2 from I2 to I3
I3: OR R1,R1,R2	WAR on R2 from I1 to I2
	WAR on R1 from I2 to I3
	WAW on R1 from I1 to I3

2. Assume there is no forwarding in this pipelined processor. Indicate hazards and add *NOP* instructions to eliminate them.

Answer:

In the basic five-stage pipeline WAR and WAW dependences do not cause any hazards. Without forwarding, any RAW dependence between an instruction and the next two instructions (if register read happens in the second half of the clock cycle and the register write happens in the first half). The code that eliminates these hazards by inserting NOP instructions is:

Instruction sequence	
`OR R1,R2,R3`	
`NOP`	Delay I2 to avoid RAW hazard on R1 from I1
`NOP`	
`OR R2,R1,R4`	
`NOP`	Delay I3 to avoid RAW hazard on R2 from I2
`NOP`	
`OR R1,R1,R2`	

3. Assume there is full forwarding. Indicate hazards and add NOP instructions to eliminate them.

Answer:
With full forwarding, an ALU instruction can forward a value to EX stage of the next instruction without a hazard. However, a load cannot forward to the EX stage of the next instruction (by can to the instruction after that). The code that eliminates these hazards by inserting NOP instructions is:

Instruction sequence	
`OR R1,R2,R3`	
`OR R2,R1,R4`	No RAW hazard on R1 from I1 (forwarded)
`OR R1,R1,R2`	No RAW hazard on R2 from I2 (forwarded)

4. What is the total execution time of this instruction sequence without forwarding and with full forwarding? What is the speedup achieved by adding full forwarding to a pipeline that had no forwarding?

Answer:

The total execution time is the clock cycle time times the number of cycles. Without any stalls, a three-instruction sequence executes in 7 cycles (5 to complete the first instruction, then one per instruction). The execution without forwarding must add a stall for every NOP we had in 2, and execution forwarding must add a stall cycle for every NOP we had in 3. Overall, we get:

No forwarding	With forwarding	Speedup due to forwarding
(7 + 4)*180 ps = 1980 ps	7*240 ps = 1680 ps	1.18

Question (4) **(14 Marks)**

a. What is data forwarding? What is a stall in a pipelined CPU?

Forwarding is making the data available to subsequent instructions as soon as the computation is complete and allowing instructions to receive this data in the beginning of the EX stage instead of retrieving it in ID. Thus, the results of the ALU and MEM register are given as possible source operands to the ALU.

b. Given the following MIPS assembly language code:

I0: ADD $4, $1, $0
I1: SUB $9, $3, $4
I2: ADD $4, $5, $6
I3: LW $2, 100($3)
I4: LW $2, 0($2)
I5: SW $2, 100($4)
I6: AND $2, $2, $1
I7: BEQ $9, $1, Target
I8: AND $9, $9, $1

	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17
I0	IF	ID	EX	MEM	WB													
I1		IF	ID	EX	MEM	WB												
I2			IF	ID	EX	MEM	WB											
I3				IF	ID	EX	MEM	WB										
I4					IF	ID	X	EX	MEM	WB								
I5						IF	X	ID	EX	MEM	WB							
I6							X	IF	ID	EX	MEM	WB						
I7									IF	ID	EX	MEM	WB					
I8										IF	ID	EX	MEM	WB				

The final execution time of the code is 13 cycles.

Question (5) **(18 Marks)**

a. Briefly explain why the computer needs to utilize a memory hierarchy?

Answer:

Computer needs need memory to fit very large programs & data and to work at a speed comparable to that of the microprocessors. The main issue is that memories are much slower than processors and the faster the memory the greater the cost per bit. The solution is to build a composite memory system which combines a small fast memory and a large slow main memory; which behaves (most of the time) like a large fast memory. This is called memory hierarchy.

b. Consider a direct-mapped cache with 128 blocks. The block size is 32 bytes.

1. Find the number of tag bits, index bits, and offset bits in a 32-bit address.

   Answer: Offset bits = 5 Index bits = 7 Tag bits = 32 – 12 = 20 bits

2. Find the number of bits required to store all the valid and tag bits in the cache.

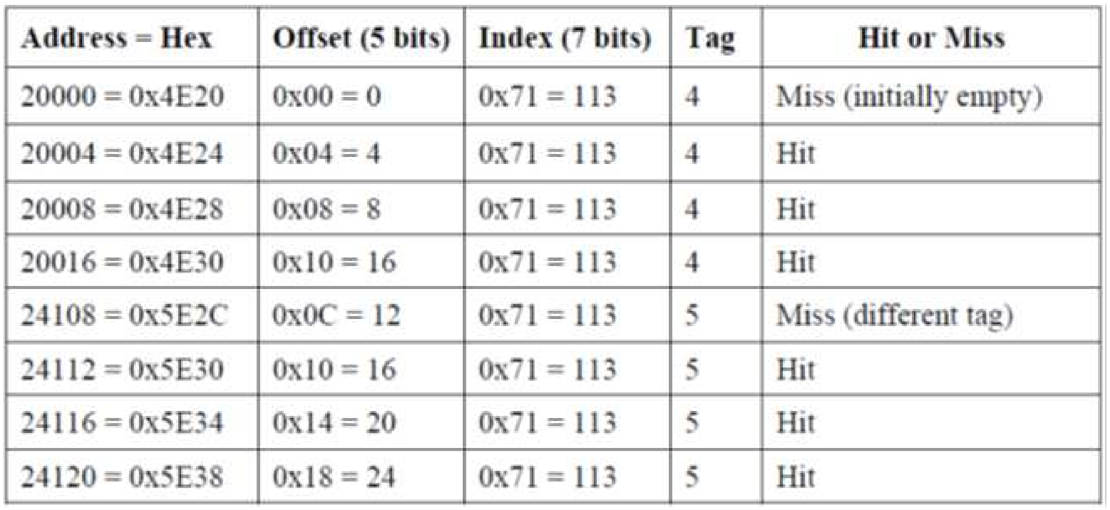
   Answer: Total number of tag and valid bits = 128 * (20 + 1) = 2688 bits

3. Given the following sequence of address references in decimal:

   20000, 20004, 20008, 20016, 24108, 24112, 24116, 24120

   Starting with an empty cache, show the index and tag for each address and indicate whether a hit or a miss occurred when referencing each address.

   Answer:



Address = Hex	Offset (5 bits)	Index (7 bits)	Tag	Hit or Miss
20000 = 0x4E20	0x00 = 0	0x71 = 113	4	Miss (initially empty)
20004 = 0x4E24	0x04 = 4	0x71 = 113	4	Hit
20008 = 0x4E28	0x08 = 8	0x71 = 113	4	Hit
20016 = 0x4E30	0x10 = 16	0x71 = 113	4	Hit
24108 = 0x5E2C	0x0C = 12	0x71 = 113	5	Miss (different tag)
24112 = 0x5E30	0x10 = 16	0x71 = 113	5	Hit
24116 = 0x5E34	0x14 = 20	0x71 = 113	5	Hit
24120 = 0x5E38	0x18 = 24	0x71 = 113	5	Hit

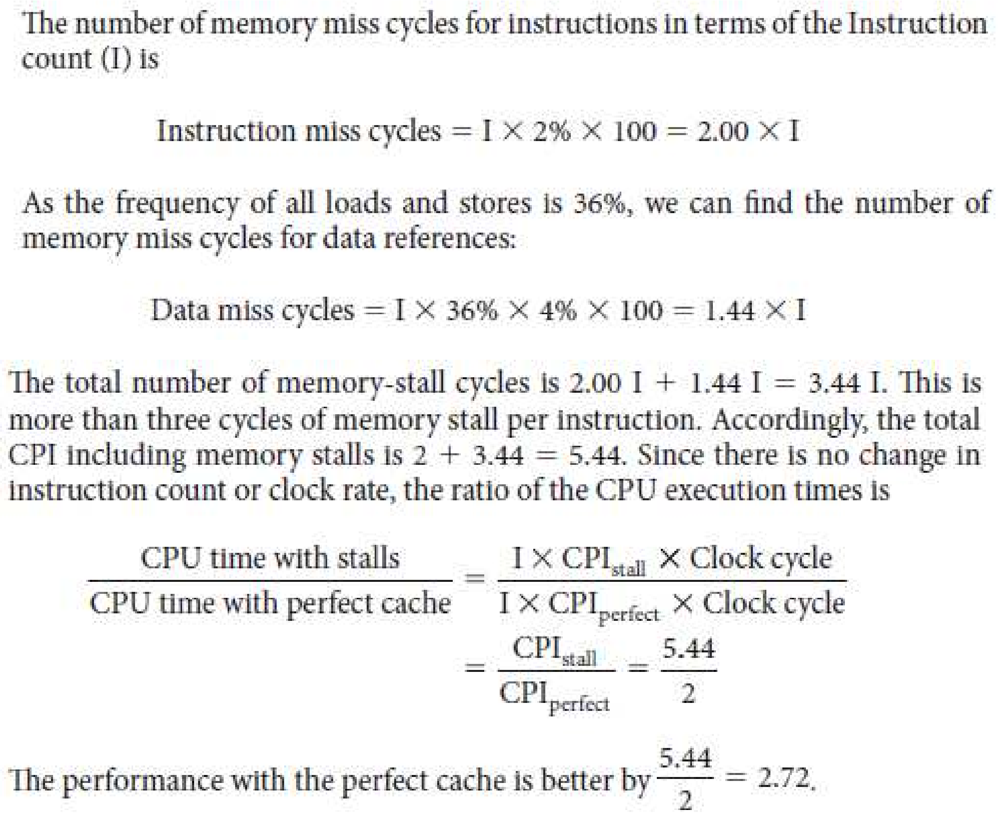
4. What is the rule of the valid bit in a direct mapped cache?

Answer:

A field in the tables of a memory hierarchy that indicates that the associated block in the hierarchy contains valid data.

c. Assume the miss rate of an instruction cache is 2% and the miss rate of the data cache is 4%. If a processor has a CPI of 2 without any memory stalls and the miss penalty is 100 cycles for all misses, determine how much faster a processor would run with a perfect cache that never missed. Assume the frequency of all loads and stores is 36%.

<u>Answer</u>:



The number of memory miss cycles for instructions in terms of the Instruction count (I) is

$$\text{Instruction miss cycles} = I \times 2\% \times 100 = 2.00 \times I$$

As the frequency of all loads and stores is 36%, we can find the number of memory miss cycles for data references:

$$\text{Data miss cycles} = I \times 36\% \times 4\% \times 100 = 1.44 \times I$$

The total number of memory-stall cycles is 2.00 I + 1.44 I = 3.44 I. This is more than three cycles of memory stall per instruction. Accordingly, the total CPI including memory stalls is 2 + 3.44 = 5.44. Since there is no change in instruction count or clock rate, the ratio of the CPU execution times is

$$\frac{\text{CPU time with stalls}}{\text{CPU time with perfect cache}} = \frac{I \times \text{CPI}_{\text{stall}} \times \text{Clock cycle}}{I \times \text{CPI}_{\text{perfect}} \times \text{Clock cycle}}$$
$$= \frac{\text{CPI}_{\text{stall}}}{\text{CPI}_{\text{perfect}}} = \frac{5.44}{2}$$

The performance with the perfect cache is better by $\frac{5.44}{2} = 2.72$.

(Good Luck)