



CS2115-2122A - exam

Computer Organization (City University of Hong Kong)



Scan to open on Studocu

CITY UNIVERSITY OF HONG KONG

Course code & title : CS2115 Computer Organization

Session : Semester A 2021/22

Time allowed : Two hours

This paper has 9 pages (including this cover page).

1. This paper consists of 5 sections.
 2. Answer ALL questions in ALL sections.
-

*This is a **closed-book** examination.*

No materials or aids are allowed during the whole examination. If any unauthorized materials or aids are found on a student during the examination, the student will be subject to disciplinary action.

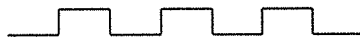
NOT TO BE
TAKEN AWAY

Section A (20 points). 2 points for each question. Only one correct answer for each question.

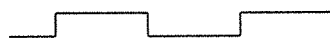
1. When a program is running, the processor directly fetches instructions from ____.
A. data bus
B. operating system
C. hard disk
D. main memory
2. The ____ in the computer is responsible for decoding the instruction.
A. control unit
B. arithmetic logic unit
C. memory controller
D. inputs/outputs controller

3. Suppose the timing diagrams of the input D and clock C of a D-flipflop triggered by falling edges are as follows,

D 

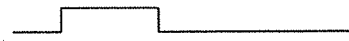
C 

then the timing diagram of its output Q is ____.

A. 

B. 

C. 

D. 

4. Which one of the following types of memory has the highest access speed?
A. Main memory
B. Level-1 cache memory
C. Level-2 cache memory
D. Level-3 cache memory
5. Which of the statements about the difference between CISC and RISC is correct?
A. RISC has higher clock frequency than CISC, so games and multimedia applications on RISC processors run faster.
B. In RISC, the operands are stored in registers.
C. CISC processors are pipelined, while RISC processors are not pipelined.
D. RISC has complex and variable-length instructions, while CISC uses simple instructions with fixed format.
E. Instructions of RISC can take several clock cycles, while CISC has single-cycle instructions.

6. In the microcomputer system, the I/O devices are connected to the system bus of the motherboard through the ____.
A. device controller
B. USB interface
C. DMA controller
D. network interface
7. ____ is used to fill the speed gap between processor and main memory.
A. Register
B. Cache
C. Flash memory
D. Stack
8. An interrupt that can be temporarily ignored is ____.
A. vectored interrupt
B. non-maskable interrupt
C. maskable interrupt
D. high priority interrupt
9. In DMA transfers, the required signals and addresses are issued by the ____.
A. processor
B. device drivers
C. DMA controllers
D. main memory
10. SIMD represents an organization that ____.
A. is capable of processing several programs at the same time
B. contains a control unit, a processor unit and a memory unit
C. includes many processing units under the supervision of a common control unit
D. none of the above

Section B (18 points). 2 points for each question.

1. $(1001\ 0010)_2 = (\text{_____})_{16}$
2. $(D2)_{16} + (3A)_{16} = (\text{_____})_{16}$
3. $(EAD2)_{16} = (\text{_____})_7$
4. The 8-bit 2's complement representation of the result of $(-6)_8 * (5)_{10} =$ _____.
5. Suppose a 32-bit floating-point number has one sign bit, 8 biased exponent bits and 23 significand bits, the floating-point representation of decimal number -3.375 is _____.
6. With the same floating-point number format as the last question,
1 10000010 011111000000000000000000
is the floating-point representation of decimal number $(\text{_____})_{10}$.

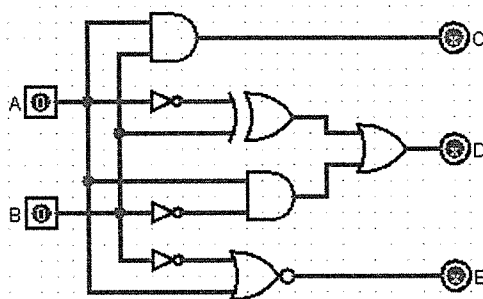
Section C (30 points).

1. **(6 points)** Suppose two CPU *A* and *B* have exactly the same circuits, but different clock frequencies. Answer the following questions:
 - a) If the clock frequency of CPU *A* is 8MHz, how long is one clock cycle of CPU *A*?
 - b) If the CPU *A* can execute 0.4 million instructions per second on average, how long is the average time to execute one instruction?
 - c) What is the average number of clock cycle to execute one instruction for CPU *A*?
 - d) The frequency of CPU *B* is 12MHz. What is the average number of instructions that can be executed per second?


2. **(4 points)** Given the following truth table, please write down the logic expression of the two outputs D and E in the **sum-of-products** form (no simplification is needed).


Inputs			Outputs	
A	B	C	D	E
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	1
1	0	0	0	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	0


3. **(6 points)** Please draw the truth table of the following circuit, where A and B are inputs; C, D and E are outputs.





Note:

 is NOT gate

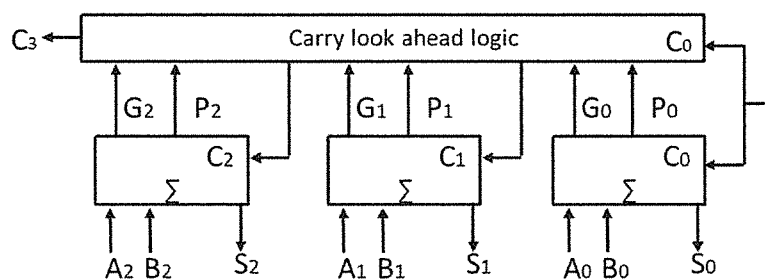
 is AND gate

 is OR gate

 is XOR gate

 is NOR Gate

4. (4 points) Assume that a program executes on a multi-core computer. 60% of its workload is executed in parallel on 20 cores, 20% of its workload is executed in parallel on 10 cores, and the remaining workload is executed one core. What is the speedup comparing with executing the entire program on one core? Write down the procedure of your calculation.
5. (6 points) The following figure shows the circuits of a 3-bit carry look ahead adder. The inputs are two 3-bit numbers, $A_2A_1A_0$ and $B_2B_1B_0$, as well as a carry-in bit C_0 . If $A_2A_1A_0=101$, $B_2B_1B_0=011$ and $C_0=1$, then what are the values of $G_0, G_1, G_2, P_0, P_1, P_2, C_3, C_2, C_1, S_2, S_1, S_0$?



6. (4 points) Suppose the processor has a direct-mapped cache, and the physical addresses of its memory is 28-bit long. The memory is byte-addressable (i.e., one byte is stored in each memory address). The direct-mapped cache has 8 cache lines, and the size of each cache line is 64 bytes. What are the number of bits for the tag, index and offset, respectively?

Section D (10 points).

1. (10 points) Write an assembly language program using the following listed instructions to sort four numbers x, y and z in decreasing order, where x, y and z are input by users. Please print each value of the sorted sequence in decreasing order (each number is print in a separate line). For example, if the inputs are

3

4

2

then the output should be

4

3

2

Instructions that may be used:

Name	Format	Functionality
<i>li</i>	<i>li \$r1, x</i>	load immediate number <i>x</i> to one register (<i>\$r1</i>)
<i>add</i>	<i>add \$r1, \$r2, \$r3</i>	add two register numbers (<i>\$r2</i> and <i>\$r3</i>) and store the result in one register (<i>\$r1</i>)
<i>sub</i>	<i>sub \$r1, \$r2, \$r3</i>	sub two register numbers (<i>\$r2</i> and <i>\$r3</i>) and store the result in one register (<i>\$r1</i>)
<i>move</i>	<i>move \$r1, \$r2</i>	move the value of one register (<i>\$r2</i>) to another register (<i>\$r1</i>)
<i>beq</i>	<i>beq \$r1, \$r2, Label</i>	if the first register number (<i>\$r1</i>) equals the second register number (<i>\$r2</i>), then jump to <i>Label</i>
<i>blez</i>	<i>blez \$r1, Label</i>	if the register number (<i>\$r1</i>) is less than or equal to zero, then jump to <i>Label</i>
<i>bnez</i>	<i>bnez \$r1, Label</i>	if the register number (<i>\$r1</i>) is not equal to zero, then jump to <i>Label</i>
<i>bgtz</i>	<i>Bgtz \$r1, Label</i>	if the register number (<i>\$r1</i>) is greater than zero, then jump to <i>Label</i>
<i>jal</i>	<i>jal ProcedureAddr</i>	jump to the starting address <i>ProcedureAddr</i> of the procedure, at the same time, store the return address into <i>\$ra</i> , and puts the parameter values in <i>\$a0-\$a3</i>
<i>j</i>	<i>j Label</i>	directly jump to the given label without needing to satisfy any condition
<i>jr</i>	<i>jr \$ra</i>	the jump register instruction, jump back to the caller when the procedure finishes, at the same time, the return values are stored in <i>\$v0-\$v1</i>
<i>syscall</i>	<i>syscall</i>	the system call code is 1 for printing an integer and is 11 for printing a character. For both cases, the argument (i.e., data to print) should be stored in <i>\$a0</i>
<i>halt</i>	<i>halt</i>	all registers (including PC, the 8 general purpose registers and all other registers) in the processor are disabled (so the processor halts)

The processor supports the instructions in the following table.

Table 2. The illustration of 5 instructions for the 32-bit single-cycle MIPS processor.

Type	Instruction	Format	Functionality
R-type	add	add \$rd, \$rs, \$rt	add two register numbers (\$rs and \$rt) and store the result in one register (\$rd)
	sub	sub \$rd, \$rs, \$rt	sub two register numbers (\$rs and \$rt) and store the result in one register (\$rd)
I-type	lw	lw \$rt, xxx(\$rs)	load a word from memory to register \$rt, the address of memory content to be loaded is computed by xxx+\$rs, where xxx is an immediate number.
	sw	sw \$rt, xxx(\$rs)	store a word from register \$rt to memory, the memory address of data to be stored is computed by xxx+\$rs, where xxx is an immediate number.
	beq	beq \$rs, \$rt, x	branch instruction, if \$rs=\$rt, jump to address x
J-type	jump	jump xxx	jump instruction, directly jump to address xxx without needing to satisfy any condition

1. (12 points) Please complete the following table 2 with the corresponding control signals of the different type of instructions. The content with X indicates “don’t care”. Please draw the output part of this table on your answer sheet and fill the blanks (you don’t need to write the output values that are already given to you in the table, but **only** need to answer those are blank).

Table 3. The control function of the processor.

Input or output	Opcode	R-type	I-type			J-type
			lw	sw	beq	
Input	Op5	0	1	1	0	0
	Op4	0	0	0	0	0
	Op3	0	0	1	0	0
	Op2	0	0	0	1	0
	Op1	0	1	1	0	1
	Op0	0	1	1	0	0
Output	Jump	0	0	0		
	Branch		0	0		X
	ALUSrc	0		1		X
	ALUOp1	1	0	0	0	X
	ALUOp0	0	0	0	1	X
	RegDst	1		X	X	X
	RegWrite	1		0		X
	MemRead		1	0	0	X
	MemWrite	0		1	0	X
	MemtoReg	0		X	X	X

2. (5 points) Suppose we will execute the following assembly program on this processor

```
sub $3,$4,$5
lw $1,20($2)
sw $9,20($10)
beq $11,$12,20
jump 10
```

Assume the “shamt” and “funct” fields for instruction “sub” are 00000 and 000001. Please write the machine codes of the above instructions (in hexadecimal numbers).

3. (5 points) Now, suppose we change the processor to a 5-stage pipeline structure. The 5 stages are:

- IF: instruction fetch
- ID: instruction decode and register file read
- EX: execution or address calculation
- MEM: data memory access
- WB: write data back to register

Moreover, the processor supports *forwarding* to directly send the result of the ALU to the register file and directly send the memory data to the register file within one cycle. Now we will execute the MIPS assembly program on this processor:

```
lw $5, 10($6)
add $7, $8, $9
sub $10, $5, $7
lw $11, 10($10)
sw $11,10($12)
```

Please complete the following table showing how the above instructions will proceed through pipeline (draw the table on the answer sheet and fill in your answer).

Instruction	cycle 1	cycle 2	cycle 3	cycle 4	cycle 5	cycle 6	cycle 7	cycle 8	cycle 9	cycle 10
lw \$5, 10(\$6)										
add \$7, \$8, \$9										
sub \$10, \$5, \$7										
lw \$11, 10(\$10)										
sw \$11,10(\$12)										

Note: Please use an **arrow** to show the forwarding between the stage that provides the data and the stage that receives the data. The following shows an example with two instructions and forwarding occurred at the 4th clock cycle:

Instruction	cycle 1	cycle 2	cycle 3	cycle 4	cycle 5	cycle 6	cycle 7
lw \$1, 10(\$2)	IF	ID	EX	MEM	WB		
add \$3 \$1, \$1			IF	ID	EX	MEM	WB