UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FINAL EXAMINATION
SEMESTER II
SESSION 2012/2013

COURSE NAME	: COMPUTER ARCHITECTURE AND ORGANIZATION / COMPUTER ARCHITECTURE
COURSE CODE	: BEC 30303 / BEC 4113 / BEX 45503
PROGRAMME	: BEB / BEC / BED / BEE / BEH / BEU
EXAMINATION DATE	: JUNE 2013
DURATION	: 3 HOURS
INSTRUCTION	: ANSWER **ALL** QUESTIONS

THIS QUESTION PAPER CONSISTS OF **TEN (10)** PAGES

Q1 (a) Convert the following expressions from reverse Polish notation to Infix notation.

(i) 3 5 7 + 2 1 − × 1 + +
(ii) ABCDE + F / + G − H / × +

(2 marks)

(b) Convert the following expressions from Infix notation to reverse Polish notation.

(i) (5 × (4 + 3) × 2 − 6)
(ii) (A−B) × (((C − D × E) / F) / G) × H

(2 marks)

(c) Produce a program to evaluate the given arithmetic statement using a stack organized computer with zero-address instructions.

$$X = \frac{A}{(B \times C)} + D$$

(5 marks)

(d) Produce an assembly-language program for the following fragment C code:

```
int n, p;
for (n = 1; n <= 10;  n++){
        p = p + 1;
}
```

Assume that the variable *n* and *p* are assigned to registers R01 and R02, respectively.

(13 marks)

Q2 (a) With the suitable diagram, sketch the typical Direct Memory Access (DMA) module.

(6 marks)

(b) Briefly explain how DMA transfer data using the block-mode method.

(4 marks)

(c) Based on Figure 2.1, synthesize the interrupt operation when user program is interrupted after the instruction at location N.

(10 marks)

Q3 (a) Figure 3.1 shows a single bus organization of the data path inside a processor. Analyze the sequence of control steps required for the bus structure as shown for each of the following instructions:

(i) Add the (immediate) number NUM to register R1
(ii) Add the contents of memory location NUM to register R1
(iii) Add the contents of the memory location whose address is at memory location NUM to register R1

Assume that each instruction consists of two words. The first word specifies the operation and the addressing mode, and the second word contains the number NUM.

(12 marks)

(b) Assume that propagation delays along the bus and through the ALU of Figure 3.1 are 0.3 and 2 ns, respectively. The setup time for the registers is 0.2 ns, and the hold time is 0 ns. Produce the minimum clock period needed?

(4 mark)

Q4 (a) Pipelining is a technique of decomposing a sequential process into suboperations, with each subprocess being executed in a special dedicated segment that operates concurrently with all other segments. Illustrate a space time diagram for a six segment pipeline showing the time it takes to process eight tasks.

(4 marks)

(b) Calculate the number of clock cycles that it takes to process 200 tasks in Q4(a).

(2 marks)

(c) A non-pipeline system takes 50 ns to process a task. The same task can be processed in a six segment pipeline with a clock cycle of 10 ns.

Analyze:

(i) The speedup ratio of the pipeline for 100 tasks.
(ii) The maximum speed up that can be achieved.

(4 marks)

(d) Consider a statement as followed:

IF A > B THEN action 1, ELSE action 2

Create a sequence of instructions using

(i) branch instruction only
(ii) conditional instructional

(4 marks)

(e) By using the suitable aid of diagram, create a simple two stage pipeline to compare execution times for the two approaches in Q4(d).

(4 marks)

Q5 (a) Processor and DMA controllers both need to initiate data transfers on the bus and access main memory. The device that is allowed to initiate transfers on the bus at any given time is called the bus master. When the current bus master relinquishes its status as the bus master, another device can acquire this status. The process by which the next device to become the bus master is selected and bus mastership is transferred to it is called bus arbitration.

(i) With the suitable aid of diagram, choose the type of bus arbitration that can fulfilled the following requirement design: Hardware must be balance with bus priority, flexible and fairness factors.

(4 marks)

(ii) Explain the bus arbitration process in Q2d(i).

(4 marks)

(b) Register R5 is used in a program to point to the top of a stack. Write a sequence of instructions using the Index, Autoincrement, and Autodecrement addressing modes to perform each of the following tasks:

(i) Pop the top two items off the stack, add them, and then push the result onto the stack.
(ii) Copy the fifth item from the top into register R3.
(iii) Remove the top ten items from the stack.

(9 marks)

(c) With aid of suitable diagram, evaluate the stack operations for 3 * 4 + 5 * 6.

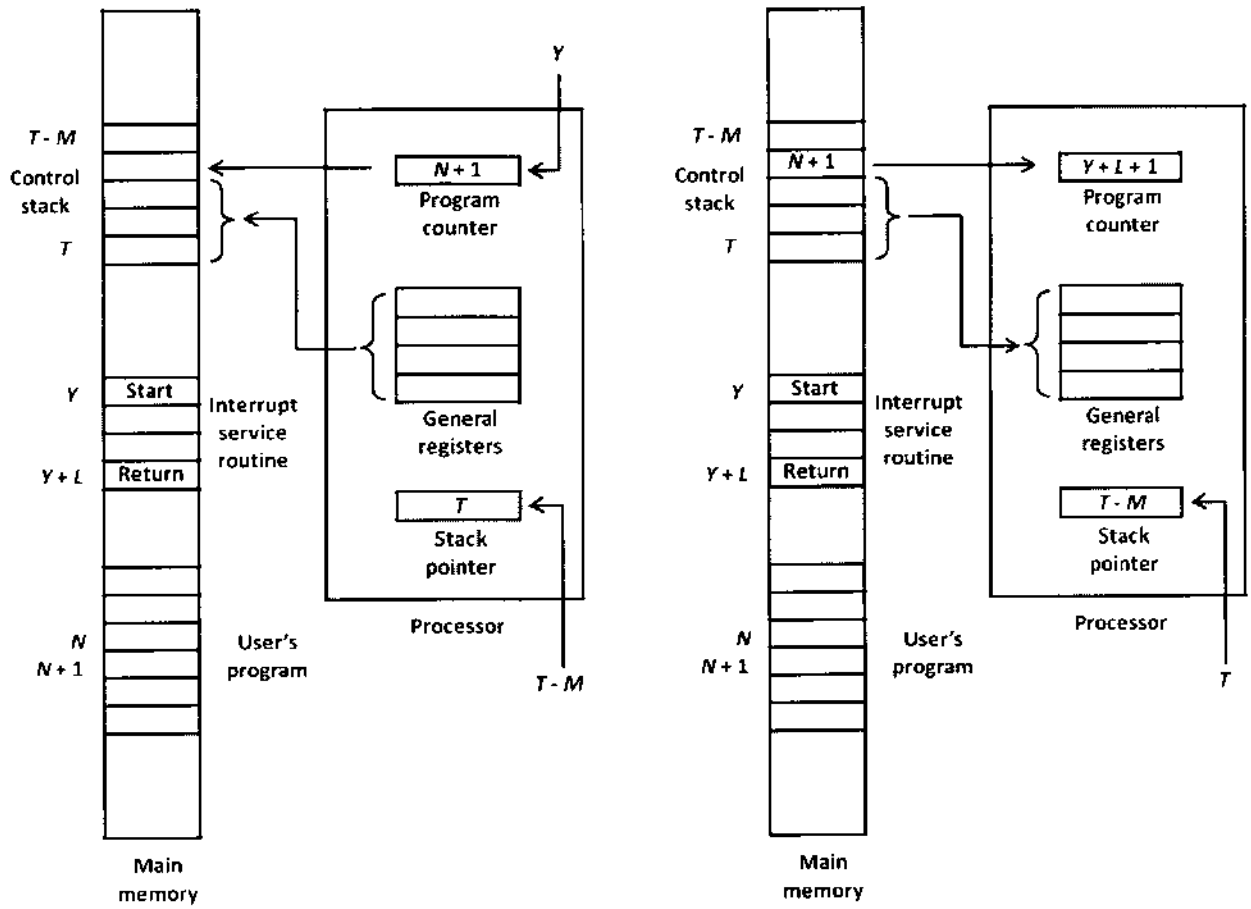
(7 marks)

- END OF QUESTION -

FINAL EXAMINATION

SEMESTER / SESSION : SEM II / 2012/2013 PROGRAMME: BEB/BEC/BEE/BED/BEH/BEU

COURSE NAME: COMPUTER ARCHITECTURE AND ORGANIZATION / COMPUTER ARCHITECTURE COURSE CODE: BEC30303/BEC4113/BEX45503



(a) Interrupt occurs after instruction at location N (b) Return from interrupt

FIGURE 2.1

FINAL EXAMINATION

SEMESTER / SESSION : SEM II / 2012/2013 PROGRAMME: BEB/BEC/BEE/BED/BEH/BEU

COURSE NAME: COMPUTER ARCHITECTURE AND ORGANIZATION / COMPUTER ARCHITECTURE COURSE CODE: BEC30303/BEC4113

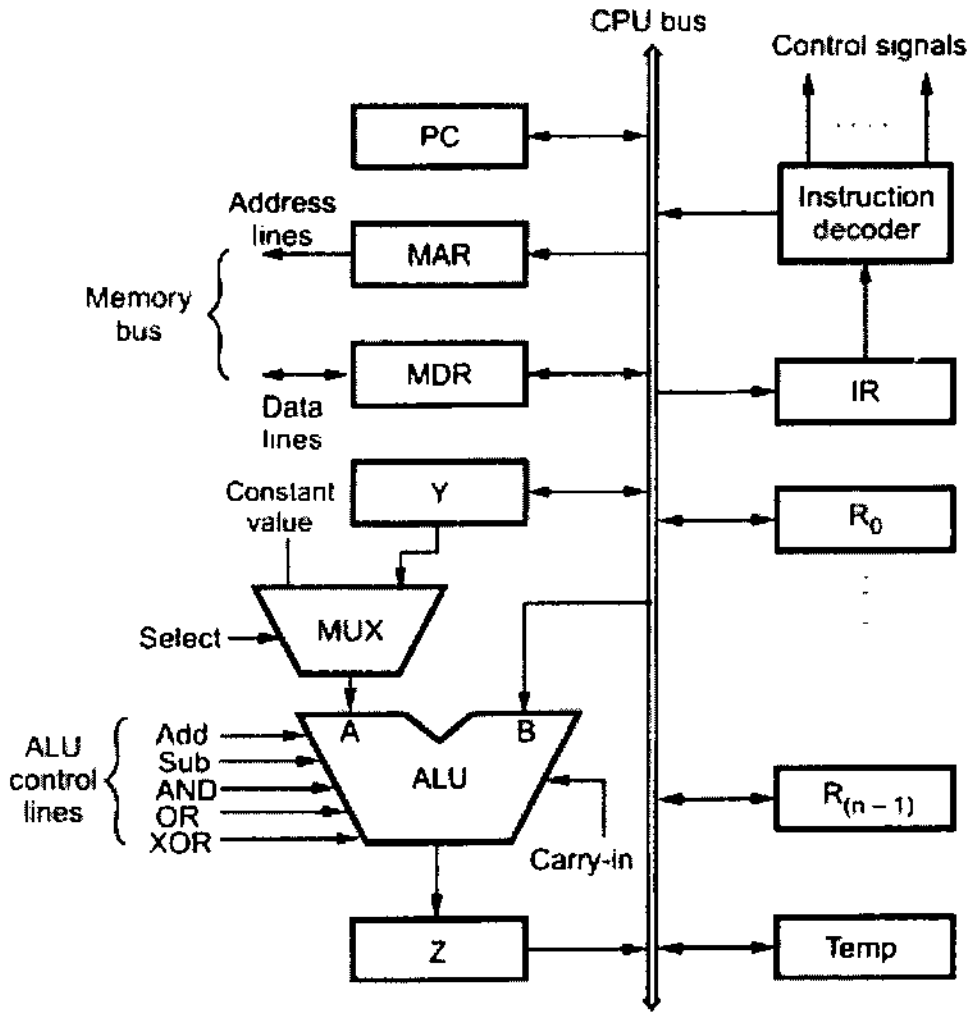


FIGURE 3.1

FINAL EXAMINATION

SEMESTER / SESSION : SEM II / 2012/2013 PROGRAMME: BEB/BEC/BEE/BED/BEH/BEU

COURSE NAME: COMPUTER ARCHITECTURE AND ORGANIZATION / COMPUTER ARCHITECTURE COURSE CODE: BEC30303/BEC4113

Appendix - Instruction Sub-set

Instruction	Description
Data Transfer Instruction	
MOV	Move data to register; move register to register e.g. **MOV #2, R01** moves number 2 into register R01 **MOV R01, R03** moves contents of register R01 into register R03
LDB	Load a byte from memory to register
LDW	Load a word (2 byte) from memory to register
STB	Store a byte from register to memory
STW	Store a word (2 byte) from register to memory
PSH	Push data to top of hardware stack (TOS); push register to TOS e.g. **PSH #6** pushes number 6 on top of the stack **PSH R03** pushes the contents of register R03 on top of the stack
POP	Pop data from top of hardware stack to register. e.g. **POP R05** pops contents of top of stack into register R05 **Note**: if you try to POP from an empty stack you will get the error message "Stack underflow"

FINAL EXAMINATION

SEMESTER / SESSION : SEM II / 2012/2013 PROGRAMME: BEB/BEC/BEE/BED/BEH/BEU

COURSE NAME: COMPUTER ARCHITECTURE AND ORGANIZATION / COMPUTER ARCHITECTURE COURSE CODE: BEC30303/BEC4113

Arithmetic Instruction	
ADD	Add number to register; add register to register e.g. **ADD #3, R02** adds number 3 to contents of register R02 and stores the result in register R02 **ADD R00, R01** adds contents of register R00 to contents of register R01 and stores the result in register R01
SUB	Subtract number from register; subtract register from register
MUL	Multiply number with register; multiply register with register
DIV	Divide number with register; divide register with register
Control Transfer Instructions	
JMP	Jump to instruction address unconditionally e.g. **JMP 100** unconditionally jumps to address location 100
JLT	Jump to instruction address if less than (after last comparison)
JGT	Jump to instruction address if greater than (after last comparison)
JEQ	Jump to instruction address if equal (after last comparison instruction) e.g. **JEQ 200** jumps to address location 200 if the previous comparison instruction result indicates that two numbers are equal, i.e. the Z status flag is set(the Z box will be checked in this case)

FINAL EXAMINATION

SEMESTER / SESSION : SEM II / 2012/2013 PROGRAMME: BEB/BEC/BEE/BED/BEH/BEU

COURSE NAME: COMPUTER ARCHITECTURE AND ORGANIZATION / COMPUTER ARCHITECTURE COURSE CODE: BEC30303/BEC4113

JNE	Jump to instruction address if not equal (after last comparison)
MSF	Mark Stack Frame instruction is used in conjunction with the CAL instruction. e.g. **MSF** reserve a space for the return address on program stack **CAL 1456** save the return address in the reserved space and jump to subroutine in address location 1456
CAL	Jump to subroutine address
RET	Return from subroutine
SWI	Software Interrupt (used to request OS help)
HLT	Halt simulation
Comparison Instruction	
CMP	Compare number with register; compare register with register e.g. CMP #5, R02 compare number 5 with the contents of register R02 CMP R01, R03 compare the contents of register R01 and R03 Note: If R01 = R03 then the status flag Z will be set, i.e. the Z box is checked If R03 > R01 then none of the status flags will be set, i.e. none of the status flag boxes are checked IF R01 > R03 then the status flag N will be set, i.e. the N status box is checked