

End Semester Examination 2013-14  
 CSE, III Yr. (I Sem),  
 30002: Computer Organization

GROUP -A

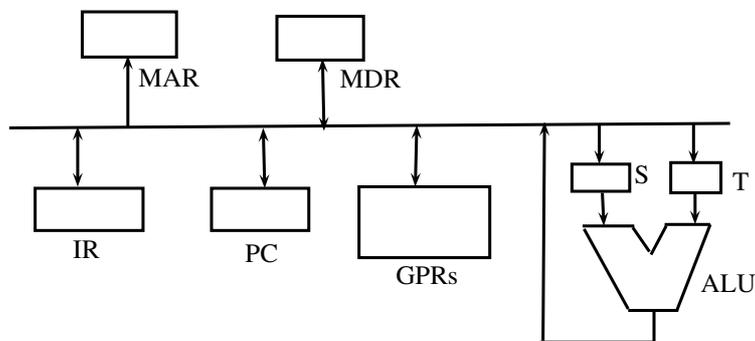
Instructions:

1. Write the question paper group (**A, B, C, D**), on front page top of answer book, as per what is mentioned in the question paper.
2. Attempt all questions.
3. Write the correct choice a, b, c, or d in answer book, for multiple choice questions (MCQs).
4. Wrong answer for an MCQ carries  $-\frac{1}{4}$  of its full weightage.
5. Except MCQs, give the detailed answer/solution for each question.

Max. Marks=40

Time= 3 Hrs.

1. (a) The ALU, S, T, the bus and all the registers in the data path are of identical size. All operations including incrementation of the PC and the GPRs (general purpose registers) are to be carried out in the ALU. Two clock cycles are needed for memory read operation – the first one for loading address in the MAR and the next one for loading data from the memory bus into the MDR. (next two questions)



- i. The instruction “add R0, R1” has the register transfer interpretation  $R0 \leftarrow R0 + R1$ . The minimum number of clock cycles needed for execution cycle of this instruction is: (2)

(a) 2 (b) 3 (c) 4 (d) 5

**Answer: (b) 3.** One cycle to load R0 in accumulator register S, one cycle to add, one cycle to store value in register R0.

- ii. The instruction “call Rn, sub” is a two word instruction. Assuming that PC is incremented during the fetch cycle of the first word of the instruction, its register transfer interpretation is

$$Rn \leftarrow PC + 1;$$

$$PC \leftarrow M[PC];$$

The minimum number of CPU clock cycles needed during the execution cycle of this instruction is: (2)

- (a) 2 (b) 3 (c) 4 (d) 5

**Answer: (b) 3.** One cycle to increment PC, one cycle to load PC into MAR, one cycle to fetch memory content and load into PC.

- (b) i. A CPU has 24-bit instructions. A program starts at address 300 (in decimal). Which one of the following is a legal program counter (all values in answers are in decimal)? Assume that each memory location is of one byte size. (2)

- (a) 400 (b) 500 (c) 600 (d) 700

**Answer: (c) 600.** Since each instruction is 3 byte (24 bit) long and memory locations are one byte each, the legal PC values are address, where  $address\ mode\ 3 == 0$ .

- ii. Which one of the following is true for a CPU having a single interrupt request line and a single interrupt grant line? (1)

- (a) Neither vectored interrupt nor multiple interrupting devices are possible  
(b) Vectored interrupts are not possible but multiple interrupting devices are possible.  
(c) Vectored interrupts and multiple interrupting devices are both possible  
(d) Vectored interrupt is possible but multiple interrupting devices are not possible

**Answer (b).** Because the vectored interrupts requires separate interrupts.

- iii. For the elements of sets {A,B,C,D} and {1,2,3,4} (1)

- (A) DMA I/O (1) High speed RAM  
(B) Cache (2) Disk  
(C) Interrupt I/O (3) Printer  
(D) Condition code Register (4) ALU

following is the correct matching:

- (a) A-4 B-3 C-1 D-2 (b) A-2 B-1 C-3 D-4  
(c) A-4 B-3 C-2 D-1 (d) A-2 B-3 C-4 D-1

**Answer (b)**

- iv. Purpose of start-bit in RS232C serial communication protocol is: (1)

- (a) to synchronize receiver for receiving every byte  
(b) to synchronize receiver for receiving a sequence of bytes  
(c) a parity bit  
(d) to synchronize receiver for receiving the last byte

**Answer (a)**

- v. In serial communication employing 8-data bits, a parity bit and 2 stop bits, the minimum baud rate required to sustain a transfer rate of 300 characters per seconds is: (1)

- (a) 2400 baud (b) 19200 baud  
(c) 4800 baud (d) 1200 baud

Total bits per characters =  $8+2+1=11$ , bits for 300 characters/sec =  $300*11 = 3300$  (baud). So minimum speed out of the given speeds is **Answer: (c) 4800 baud**

- vi. A hard disk with a transfer rate of 10 Mbytes/second is constantly transferring data to memory using DMA. The processor runs at 600 MHz, and takes 300 and 900 clock cycles to initiate and complete DMA transfer respectively. If the size of the transfer is 20 Kbytes, what is the percentage of processor time consumed for the transfer operation? (3)

- (A) 5.0% (B) 1.0% (c) 0.5% (D) 0.1%

**Answer:(D) 0.1%** Time for transfer of 20kbytes =  $\frac{20 \times 10^3}{10 \times 10^6 \times 2} = 10^{-3}$  sec. This is because for a one word transfer, the address is sent on bus in one clock and

other cycle for memory read. No. of cycles for transfer =  $10^{-3} \times 600 \times 10^6 = 6 \times 10^5$  cycles. Total cycles  $900 + 300 + 6 \times 10^5 \approx 6 \times 10^5$ . The fraction of time of cpu spent =  $\frac{6 \times 10^5}{600 \times 10^6} \times 100 = 0.1\%$

vii. The following code is to run on a pipelined processor with one branch delay slot:

```
I1 : ADD R2 <- R7 + R8
I2 : SUB R4 <- R5 - R6
I3 : ADD R1 <- R2 + R3
I4 : STORE Memory [R4] <- R1
BRANCH to Label if R1 == 0
```

Which of the instructions I1, I2, I3 or I4 can legitimately occupy the delay slot without any other program modification? (2)

- (a) I1 (b) I2 (c) I3 (d) I4

**Answer: (D) I4:** since R1 is only read in I4, and its value will remain as it is while BRANCH is executed and after BRANCH get executed, since in BRANCH R1 is only read.

2. Suppose a stack representation supports, in addition to PUSH and POP, an operation REVERSE, which reverses the order of the elements on the stack.

(a) To implement a queue using the above stack implementation, show how to implement ENQUEUE using a single operation and DEQUEUE using a sequence of three operations. (3)

For Queue, we keep TOS (top of stack) as rear of the queue, and BOS (bottom of stack) as front of the queue. Thus, to add in queue from back, there is only instruction: PUSH. To remove from the front of queue, the three instructions in order are: REVERSE, POP, REVERSE.

(b) The following postfix expression, containing single digit operands and arithmetic operators + and \*, is evaluated using a stack. (3)

5 2 \* 3 4 \* 5 2 \* \* +

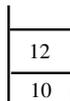
show the contents of stack:

i after evaluation of 5 2 \* 3 4 \*

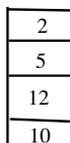
the steps are as follows:

- i. 5 is read from input and pushed onto stack
- ii. 2 is read from input and pushed onto stack
- iii. \* is read and result  $\text{pop} * \text{pop} = 10$  is pushed onto stack
- iv. 3 is read from input and pushed onto stack
- v. 4 is read from input and pushed onto stack
- vi. \* is read and result  $\text{pop} * \text{pop} = 12$  is pushed onto stack

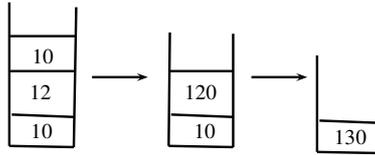
The resultant stack is as shown below:



ii after evaluation of 5 2 \* 3 4 \* 5 2 The resultant stack is:



iii at the end of evaluation. The resultant stack is:



3. A new microprocessor is being designed with a conventional architecture employing single-address instructions and 8-bit words. Due to physical size constraints, only eight distinct 3-bit opcodes are allowed. The use of modifiers or the address field to extend the opcode is forbidden.

(a) What eight instructions would you implement? specify the operations performed by each instruction as well as the location of its operands. (3)

Answer The following eight Instructions are sufficient:

- add,
- load,
- stor,
- and,
- or,
- cma,
- jmp,
- hlt.

Following is explanation:

- i. CMA is complement accumulator, jp is jump. All instructions, except cma, have two operands, one is explicit and other is implicit, which is accumulator. cma has no operand, and takes accumulator as one implicit operand.
  - ii. The remaining five bits ( $d_0 \dots d_4$ ) are for operand.  $d_4 = 0$  indicates that operand is in any one of the 16 registers, whose number is represented by  $d_3 \dots d_0$ .
  - iii. when  $d_4 = 1$ , it indicates that operand is in memory, whose address is indicated by the register  $d_3 \dots d_0$ .
  - iv. Each of these registers  $d_0 \dots d_{15}$  is 16 bit, hence total memory of 64 k can be addressed.
  - v. The register  $r_0$  is accumulator register (8 bits) and status register to hold status flag for zero/non-zero. jmp is unconditional jump instruction.
- (b) Demonstrate that your instruction set is functionally complete in some reasonable sense; or if it is not, describe an operation cannot be programmed using your instruction set. (3)

Answer It is possible to do any arithmetic using add, cma. For example, for subtraction, we do 2's complement addition. 2's complement is cma, followed with add 1. This 1 can be kept in a register. The remaining justification is as follows:

- i. The jump can be made conditional subject to the content of a register, say r1. If  $r1=0$ , then jumps else not. For unconditional jump, keep  $r1=0$ .
  - ii. loop and branching are thus implemented, which are necessary for control.
  - iii. all the logical operations are possible by and, or, not.
  - iv. memory store, and load from it, are possible by load and store instructions.
  - v. The RR, RM, MR, MM based instructions are possible. The indirect addressing is not provided. But that is not compulsory, as many HLLs have no provision.
  - vi. You cannot do the rotate accumulator.
4. It is required to design a hardwired controller to handle the fetch cycle of a single address of an instruction. The operand should be delivered in the fetch cycle itself. Assume that lower 8-bits of an instruction constitute the operand field.

- (a) Draw the logic schematic of the hardwired controller including the data paths. (4)

Answer. The answer is similar to the contents of slide 9, page 8. The instruction op-code part goes into IR, and operand part goes to any register or accumulator. The control signals are generated as given in slide 9 page 10. The data bus is 16 bits.

- (b) Give the micro-operations to realize the above instruction's fetch cycle. (3)

Answer. The answer can be found in slide set 9, page 8.

5. An instruction pipeline has five stages, each stage takes 2 nanoseconds and all instructions use all five stages. Branch instructions are not overlapped, i.e., the instruction after the branch is not fetched till the branch instruction is completed. Under ideal conditions,

- (a) calculate the average instruction execution time assuming that 20% of all instructions executed are branch instructions. Ignore the fact that some branch instructions are may be conditional. (3)

Answer: The five stages are say: FI, DI, FO, EI, WO. After first five instructions, every 2 nsec, one instruction is executed, provided that there is no branch instruction. A branch instruction will take only 4 states, as there is no WO. So consuming  $4 * 2 = 8$  nsec. Average time =  $1 * 2 + 0.2 * 8 = 3.6$  nsec. Note: Because the branch instructions are not overlapped.

**Explanation:**

$$\begin{aligned} Avg, time &= \frac{0.2 \times (5 \times 2) \times n + 0.8 \times ((5 \times 2) + (n - 1) \times 2)}{n} \\ &= 2 + \frac{0.8(10 + (n - 1)2)}{n} \\ &= 2 + \frac{8}{n} + \frac{2 \times 0.8 \times n}{n} - \frac{1.6}{n} \\ \lim_{n \rightarrow \infty} &= 2 + 1.6 = 3.6nsec \end{aligned}$$

- (b) If a branch instruction is a conditional branch instruction, the branch need not be taken. If branch is not taken, the instruction following to that instruction can be overlapped. When 80% of all branch instructions are conditional branch instructions, and 50% of the conditional branch instructions are such that the branch is taken, calculate the average instruction execution time. (3)

Answer 1:  $T_{avg} = (1 + \text{stall frequency} * \text{stall penalty}) * \text{clock time}$

20% are branch instructions out of which 20% unconditional and 80% conditional 50% of conditional are taken, we have to consider only taken branch conditions:

so  $T_{avg} = (1 + 0.2(0.2 + 0.8 * 0.5) * (5 - 1)) * 2 = 2.96$  ns (considering 80% as fraction of original 20%). The detailed explanation is as follows: Out of 20%, 80% are conditional branch instructions. Thus, conditional branch instructions are  $= 0.8 * 0.2n = 0.16n$ . The non-conditional instructions are  $0.04n$ . For 50% of conditional branch instructions, there is branch,  $= \frac{0.16n}{2} = 0.08n$ . Thus total time is =

$$\begin{aligned} &\frac{2(5 + 0.8n - 1) + (0.08n \times 10) + (0.04n \times 10) + (0.08n - 1 + 5) \times 2}{n} \\ &= \frac{2(4 + 0.8n) + 0.8n + 0.4n + (0.08n + 4) * 2}{n} \\ &= \lim_{n \rightarrow \infty} = 2.96nsec \end{aligned}$$

Answer 2:  $(1 + 0.8 * (0.5 * (5 - 1))) * 2 = 5.2$ ; here 0.5 corresponds to 50% branch instructions where branch has taken. (considering 80% as fraction of 100)

Considering 80% total branch instructions:

$$\begin{aligned} \text{Avg.time} &= \frac{0.8(0.5(10 * n) + 0.5(10 + (n - 1)2)) + 0.2(10 + (n - 1)2)}{n} \\ &= \frac{0.8(5n + (5 + (n - 1))) + 0.2(8 + 2n)}{n} \\ &= \frac{0.8(5n + (4 + n)) + 1.6 + 0.4n}{n} \\ &= 4 + \frac{3.2}{n} + 0.8 + \frac{1.6}{n} + 0.4 \\ &= \lim_{n \rightarrow \infty} = 4 + 0.8 + 0.4 = 5.2nsec. \end{aligned}$$