ICO End Sem. Examination Solution

1. (a) Consider a 16-bit processor in which the following contents appears in main memory, starting at location 200:

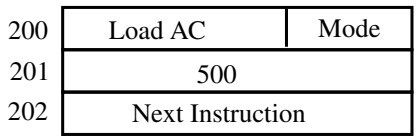

Figure 1: Three locations of main memory.

The first part of the first instruction indicates that this instruction loads a value into an accumulator. The mode field specifies addressing mode and, if appropriate a source register; assume that when used, the source register is R_1, which has a value of 400. There is also a base register that contains the value 100. The value 500 in location 201 may part of the address calculation. Assume that location 399 contains the value 999, location 400 contain a value 1000, and so on. Determine the effective address and operand to be loaded for the following address modes: (6=1+1+1+1+1+1)

Ans. Base register, used as displacement has value = 100. R_1 has value 400. memory location 399 has value 999, 400 has value 1000, ..., x has value $x + 600$.

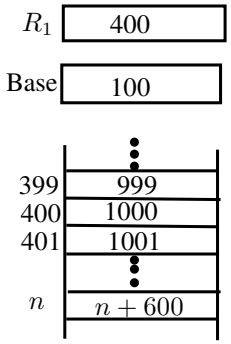

Figure 2: EA and Operand Computation.

i. Direct: EA=500, operand=1100
ii. Indirect: EA=1100, operand=1700
iii. Immediate: EA=201, operand=500
iv. Displacement: EA=900, Operand=1500
v. Register: EA=Register R_1, operand=400
vi. Register Indirect: EA=400, operand=1000

(b) Design a variable-length opcode to allow all of the following to be encoded in a 32-bit instruction: (4=2+1+1)

- Instructions with two 13-bit addresses and one 3-bit register number

Ans. adr1 = 13-bit, adr2 = 13-bit, register 3-bit (8 no. of registers), remaining 32-13-13-3= 3 bit for opcode. opcodes can be:
move: (string from source adr1 to dest adr2), register used as displacement register
add: source operand to dest. operand, and register as displacement register
Similarly, there can be total $2^3 = 8$ instructions.

- Instructions with one 13-bit address and one 3-bit register number

Ans. register plus adr1 is one address, and other operand is immediate of 13-bits.

- Instructions with no address or registers

Ans. A mode can specify other instructions formats, along with immediate address.

2. (a) A CPU has a clock of 3 GHz. It has a special instruction for block move, which move a string of bytes from one memory area to another. The fetch and decode of this instruction takes 6 clock cycles. After that it takes 10 clock cycles to transfer each byte from one memory location to another. (5=2.5,1.5.1)

i. Determine the length of instruction cycle for a string of 32 bytes.

Ans. Fetch and decode = 6 cycles.
clock cycles for 32-bytes transfer = 320
Total clock cycles per instruction = 326, length in times= $326 * 1/3 * 10^{-9}$ sec.= 0.1086 microsec.

ii. What is the worst case delay in clock cycles for acknowledging an interrupt and DMA ?

Ans. Worst case Interrupt acknowledge delay = 326 clock cycles
Worst case DMA acknowledge delay = 10 clock cycles.

iii. What is the best case delay in clock cycles for acknowledging an interrupt and DMA ?

Ans. Best case delay in both = 0 secs.

(b) A non-pipelined CPU has a clock speed of 2 GHz, and average number of CPU clocks cycles per instruction eaeaare four. Imagine that this processor has been upgraded by 6-stage pipeline. Let there are internal pipeline delays, so the clock rate of the processor needs to be reduced to 1.5 GHz. (5=2.5,2.5)

i. What is the speed up achieved for a typical program ?

Ans. Due to introduction pipeline, the same CPU speeds up by 6 times, but due to delays, it slows down from 2 GHz to effective clock of 1.5 GHz. Hence, the net speed up is:

$$= 6 \times \frac{1.5Ghz}{2Ghz}$$
$$= 4.5 \; times \; speedup$$

ii. What is MIPS (million instructions per sec.) rate the processor?

Ans. There are 6 instructions in parallel, effectively, due to 6-stage pipeline. Thus, if one clock cycles were to execute one instruction, the speed could be 1.5GHz $\times$ 6 = 9 GHz. But, each instruction takes 4-cycles, hence speed = 9/4 = 2.25 GHz. In MIPs it is = 2250 MIPs.

3. (a) Suggest a bus arbitration algorithm to grant the control of bus to any of 8 DMA controllers ($C_0 : C_7$) connected to various high speed devices. The algorithm should have the feature so first four controllers ($C_0 : C_3$) have double the priority compared to the last four ($C_4 : C_8$). Describe the method and steps for algorithm. (5)

Ans.1. There can be may approaches. 1. Static fixed priority algorithm. The algorithm should keep the priority as: $c0, c_1, c_2, c_3 = 0.333; c_4, c_5, c_6, c_7 = 0.667$. The

Ans.2 TDM/Round-Robin: The algorithm assigns the bus control in the order of round-robin chain as shown in the figure 3.

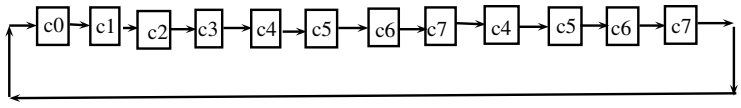

Figure 3: Assignment of bus in Robin-Robin fashin to DMA Controllers.

(b) Assume that it is required to be able occasionally to delay a CPU's response to to a DMA request to the end of the current instruction cycle. Design the necessary add-on logic to implement this type of delayed DMA request. There should be provision to turn on (enable) and turn off (disable) the DMA delay. State clearly all the assumptions underlying your design. (5)

Ans. The DMA break-point signal (i.e., any machine cycle over) coming from the control logic and 'instruction execution over' signal coming from the control logic are anded together to introduce the delay in DMA to occur. The switch's operation will enable/disable this operation. Any suitable diagram for represent this operation.

4. (a) A computer consists of a CPU and an IO device D connected to the main memory M via a one-word shared bus. The CPU can execute a maximum of 10^5 instructions/sec. An average instruction requires five machine cycles, three of which use the memory bus. A memory read or write operation uses one machine cycle. Suppose that the CPU is continuously executing a "background" program that requires 95% of its instruction execution rate but no IO instructions. Now the device is to be used to transfer very large blocks data to and from M. (5=2.5,2.5)

i. If programmed IO is used and each one-word IO transfer requires the CPU to execute two instructions, estimate the maximum IO transfer rate r_{max} possible through D.

Ans. The CPU can devote only 5% of its time to IO, so maximum IO instruction rate is $= 0.05 \times 10^5 =$ 5000 instructions/sec. Since each IO requires two instructions, the IO transfer rate is half, i.e., $r_{max} = 2500$ words per sec.

ii. Estimate r_{max} if DMA transfer is used.

Ans. The number of machine cycles available for DMA control due to 5% time $= 0.05 \times 5(cycles/per\ instr.) \times 10^5 =$ 25,000 machine cycles/sec.

The number of machine cycles available for DMA control due to 2 machine cycles per instruction, out of 95% of the instructions $= 0.95 \times 2 \times 10^5 = 190,000$ machine cycles/sec.

Total is 190.000+25,000= 215,000 words per sec.

(b) For programmed IO, the processor gets stuck in a wait loop doing status checking of an IO device. To increase the efficiency, the IO software could be written so that the processor periodically checks the status of the device. If the device is not ready, the processor can jump to other tasks. After some timed interval, the processor comes back to check status again. (5=2.5,2.5)

i. Consider the above scheme to transfer data, one character at a time to a printer that operates at 10 characters per second (cps). What will happen if its status is scanned every 250 ms?

Ans. With 10 char/sec, time for 1 char print is 100 msec. Now, if speed of checking is with 250 m. sec. interval, one char can be printed at maximum 250 m.sec. interval, thus maximum speed achievable is 4 char/sec.

ii. Next consider a keyboard with a single character buffer. On average, characters are entered at a rate of 8 cps. However, the time interval between two consecutive key depressions can be as short as 70 ms. At what frequency should the keyboard be scanned by the I/O program?

Ans. Average Time for 1 char entry = 1/8 sec. = 125 msec. But, when depressed with speed, it can be 70 msec. So the minimum frequency of scanning of the keyboard is 1/0.07 = 14.3 times per sec.

5. (a) By analysing the program below, write the equation that was implemented on the stack architecture. Assume A, B, C, D, E, F, G, H are the memory addresses to hold a, b, c, d, e, f, g, h, respectively. (3)

```
PUSH B
PUSH C
PUSH D
MUL
PUSH E
```

```
MUL
PUSH F
PUSH G
MUL
PUSH H
ADD
SUB
ADD
POP A
```

Ans. $a = cde - (fg + h) + b$

(b) Consider the Following Machine instructions:

- ADD R_0, (R_5) ; register indirect addressing

Ans.

$$t_1 : MAR \leftarrow (IR(adr)); i.e., MAR \leftarrow (R_5)$$
$$t_2 : MBR \leftarrow memory$$
$$t_3 : MAR \leftarrow (MBR)$$
$$t_4 : MBR \leftarrow memory$$
$$t_5 : y \leftarrow MBR$$
$$t_6 : z \leftarrow y + R_0$$
$$t_7 : acc \leftarrow z$$

- BNZ ; branch on non zero to address in register R_{15}

Ans.

$$t_1 : if(flag \neq 0)$$
$$t_2 : MAR \leftarrow (R_5)$$
$$t_3 : MBR \leftarrow memory$$
$$t_4 : PC \leftarrow (MBR)$$

Draw a diagram representing the minimum hard-wired CPU's control logic for a machine to sequence the execution of these instructions. Also, write the micro-operations sequences for the above two instructions. (7=3,4)

The data movement diagram is in the similar lines as shown in the fig. below.

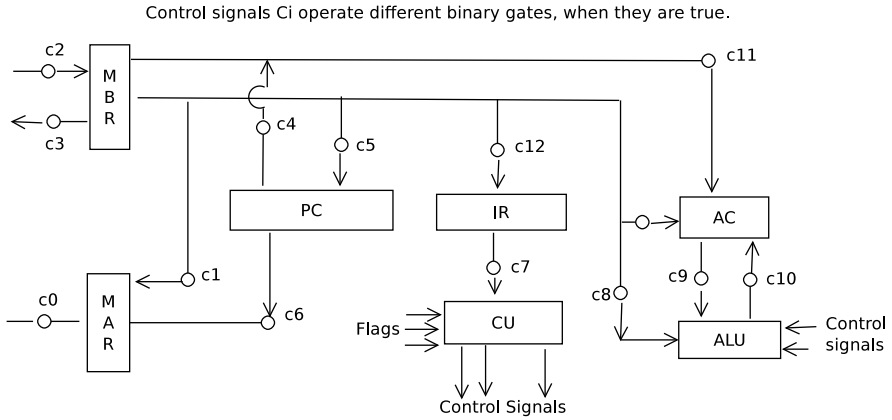

Figure 4: Control unit structure.