

## ASSIGNMENT - I

### COMPUTER ORGANIZATION & ARCHITECTURE

1. What is the Memory Unit & Control Unit?

Ans. • Memory Unit :- The function of Memory Unit is to store programs & data. There are two classes of storage, called :-

(i). Primary

(ii). Secondary

Primary Memory is a fast memory that operates at electronic speeds.

Examples of Primary memory is RAM (Random Access Memory) & ROM (Read only Memory). The memory contains a large number of semiconductor storage cells, each capable of storing one bit of information.

Secondary Memory is used when large amounts of data & many programs have to be stored, particularly for information that is accessed infrequently. A wide selection of secondary storage devices is available, including magnetic disks & tapes & optical disks (CD-ROMS).

- Control Unit :- The memory, arithmetic & logic, & I/P & O/P units store & process information & perform I/P & O/P operations. The operation of these units must be co-ordinated in some way. This is task of the control unit.

The control unit is effectively the nerve centre that sends control signals to other units & senses their states.

2. Explain the following terms :-

Ans. (a). Addressing Modes :- The basic method for addressing memory operands is to generate the effective address, EA, of the operand by adding a signed offset to the contents of a base register, Rn, which is specified in the instruction. The magnitude of the offset is either an immediate value, contained in the low-order 12 bits of the instruction, or it is the contents of a third register, Rm.

For example, the load instruction :-

LDR Rd, [Rn, #offset].

(b). Memory Locations & Address :- A memory address is a unique identifier used by a device or CPU for data tracking. This binary address is defined by an ordered & finite sequence allowing the CPU to track the location of each memory byte.

Hardware devices & CPUs track stored data by accessing memory addresses via data buses.

(c). Hit Rate & Miss Penalty :- Hit Rate is defined as the ratio of number of hits to that of total number of attempts in percentage.

$$\text{Hit Rate} = \frac{\text{No. of hits}}{\text{Total No. of Attempts}} \times 100\%.$$

- Miss Penalty is defined as the extra time required to fetch a block into a level of the memory hierarchy from the lowest level is called miss penalty.

$$\text{Miss Penalty } L_1 = \text{Hit time } L_2 + \text{Miss Ratio } L_2 \times \text{Miss Penalty } L_2.$$

∴ where,  $L_1$  &  $L_2$  are the levels of cache.

(d) ROM :- Read only Memory (ROM) is a type of non-volatile memory used in computers & other electronic devices. Data stored in ROM can't be electronically modified after the manufacture of the memory device.

It's normal operation involves only reading of stored data, so memory of this type is called Read only Memory.

3. What is the advantage & disadvantage of using a variable length instruction format?

Ans. • Advantage :- It provides large repository of opcode with variable length.

• Disadvantage :- Processor complexity depends on instruction length. Variable length instruction format does not remove desirability of instruction length which is integrally related to word length.

4. LRU is almost universally used as the cache Replacement policy. Why?

Ans. LRU cache stands for Least Recently Used Cache, which evict least recently

Used entry. As cache purpose is to provide fast & efficient way of retrieving data.

It need to meet certain requirement.

Some of the requirement are, fixed size.

Cache needs to have some bounds to limit memory usages!

5. Write the program that can evaluate the expression :  $(A * B) + (C * D)$ .

In a single accumulator processor. Assume that the processor has load, store, multiply & add instruction, & that all values fit in the accumulator.

Ans: LOAD A

MUL B

STORE E

LOAD C

MUL D

ADD E

6. Registers R1 & R2 of a computer contain the decimal values 1200 & 4600. What is the effective address of the memory operand in each of the following instruction?

(a). Load 20(R1), R5 .

Ans:  $1200 + 20 = 1220$ .  $R_5 = 1220$ .

(b) Move #3000, R5.

Ans.  $R5 = 3000$ .

(c) Stosue R5, 30 (R1, R2).

Ans. This means,  $R5 = 30 + R_1 + R_2$

$$R5 = 30 + 1200 + 4600$$

$$R5 = 5830.$$

(d) Add -(R2), R5.

Ans.  $R5 = R2 - 1$ .

$$R5 = 4600 - 1 = 4599.$$

(e) Subtract (R1)+, R5.

Ans. ~~REG~~  $\therefore EA = 1200$ .

It is Post increment addressing.

$$R5 = 1200.$$

7. Register R5 is used in a program to point to the top of stack. Write a sequence of instruction using the index, AUTO inc. & AUTO dec., addressing modes to perform the following task:

(a) Pop the two items off the stack, add them, & then push the result onto the stack.

Ans: MOVE - (R5) +, R0 MOVE - (R5, R1)

ADD - (R5) +, R0 MOVE - (R5), R2

MOVE R0, -(R5) ADD R3, (R1), (R2)

MOVE R3, (R5) +

(b): Copy the fifth item from the top into Register R3.

Ans: MOVE [R5], R3 MOVE - (R5), #10  
STORE R3, (R5).

8. The subroutine call instruction of the computer saves the return address in a processor register called the link register RL. What would you do to allow subroutine nesting? Would your scheme allow the subroutine to call itself?

Ans: A common programming practice, called subroutine nesting, is to have one subroutine call another. In this case, the return address of the second call is also stored in the link register, destroying the previous contents. Hence, it is essential to save the contents of the link register in some other location before calling another subroutine. Otherwise, the return address of the first routine will be lost.

9. If a memory system consists of a single external cache with an access time of 20ns & a hit rate of 0.92, & a main memory with an access time of 60ns, then what is effective memory access time of this system?

Ans. Effective Access time =  $t_{cache} + (1 - \text{hit}_{cache}) \times t_{RAM}$ .

$$t_{eff} = 20 + (1 - 0.92) \times 60$$

$$t_{eff} = 28.248 \text{ ns}$$

10. We now add virtual memory to the system described in ques. 9. The TLB is implemented internal to the processor chip & takes two ns to do a translation on a TLB hit. The TLB hit ratio is 98%, the segment table hit ratio is 100% and the page table hit ratio is 50%. What is the effective memory access time of the system with virtual memory?

Ans.  $t_{TLB} = 2 \text{ ns}$ , Main memory access time = 60ns,  
External cache access time = 20ns.

$$t_{eff} = t_{TLB} + (1 - \text{hit}_{TLB}) \times (t_{seg} + t_{page}) + t_{cache} \\ + (1 - \text{hit}_{cache}) \times t_{RAM}$$

$$t_{eff} = 2 + 0.02(20 + 20 + 0.5 \times 60) \text{ ns} + 20 + 0.08 \times 60 \\ = 28.2 \text{ ns}$$

11. Assume that Registers R2, R3 & R4 store values 100, 200 & 400 resp. & R1 is a general purpose Register. Also assume that  $\text{Mem}[96] = 20$ ,  $\text{Mem}[100] = 300$ ,  $\text{Mem}[104] = 30$ ,  $\text{Mem}[300] = 500$ ,  $\text{Mem}[400] = 100$  &  $\text{Mem}[600] = 38$ . What value will be stored in Registers R1 & in R2 in each of the following cases:

a. ADD R1, R2, R4.

ANS:  $R1 = 500$ ,  $R2 = 200$ .

b. ADD R1, R2, #4.

ANS:  $R1 = 500$ ,  $R2 = 415$ .

c. ADD R1, R2, (R4).

ANS:  $R1 = 465$ ,  $R2 = 415$ .

d. ADD R1, R2, 300 (R2).

ANS:  $R1 = 465$ ,  $R2 = 400$ .

e. ADD R1, R2, (R2+R3).

ANS:  $R1 = 500$ ,  $R2 = 400$ .

f. ADD R1, R2, @ (100).

ANS:  $R1 = 500$ ,  $R2 = 600$ .

g. ADD R1, R2, @ (R2).

ANS:  $R1 = 625$ ,  $R2 = 600$ .

h. ADD R1, R3, (R2) + (assume d=4).

Ans. R1 = 500, R2 = 604.

i. ADD R1, R3, -(R2) (assume d=4).

Ans. R1 = 600, R2 = 600.

j. ADD R1, R2, 100(R2)[R2] (assume d=4).

Ans. R1 = 1000, R2 = 600.

12. Briefly Explain instruction format.

Ans. • Instruction format: An Instruction format defines the layout of the bits of an instruction, in terms of its constituents, parts.

An instruction format must include an opcode & implicitly or explicitly zero or more operands. Each explicit operand is referenced using one of the addressing mode that is available for that machine.

13. An 8-bit Register R contains the binary value 10011100. What is the Register value after an Asithmetic shift Right? Starting from the initial no. 10011100, determine the Register value after an Asithmetic shift left, state whether there is an overflow?

Ans.  $R = 10011100$ .

As arithmetic shift Right: 110011100.

As arithmetic shift left: 00111000.

Overflow, because a negative no. changed to positive.

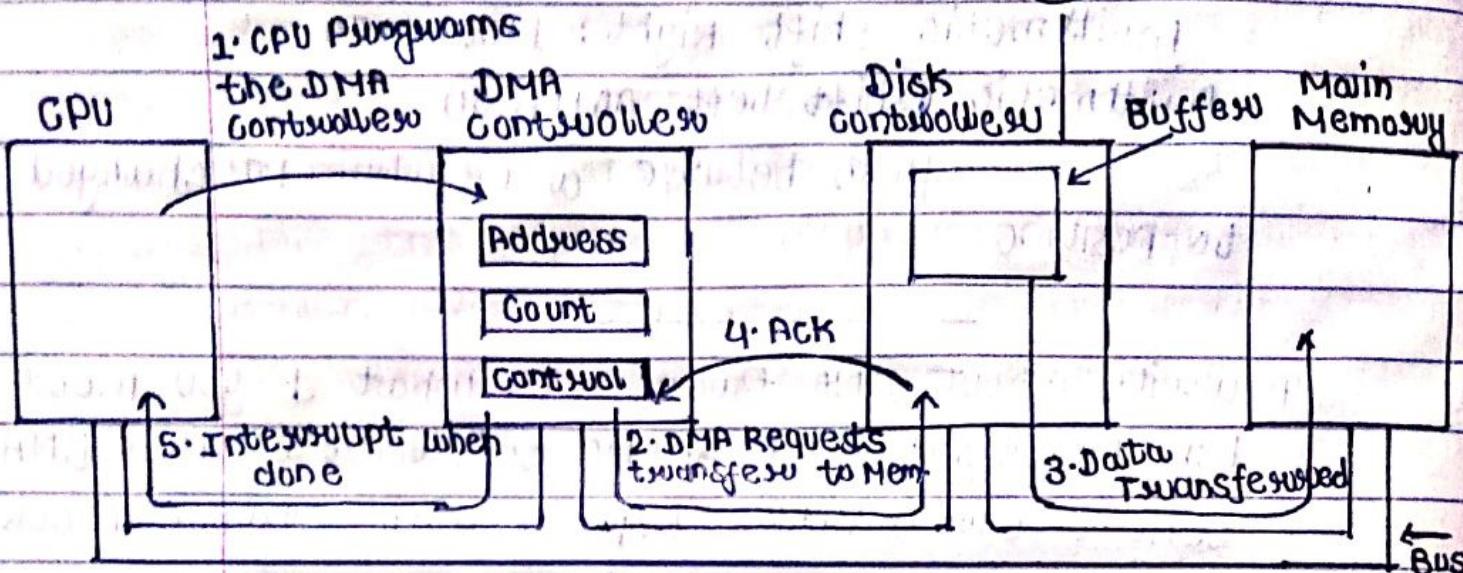
14. Write about DMA transfer. What do you mean by initialization of DMA controller? How DMA controller works? Explain with suitable block diagram.

Ans. Direct memory access (DMA) is a method that allows an input / output (I/O) device to send or receive data directly to or from the main memory operations. The process is managed by a chip known as a DMA controller (DMAC).

DMA is means of having a peripheral device control a processor's memory bus directly. DMA permits the peripheral, such as a UART, to transfer data directly to or from memory without having each byte (or word) handled by the processor.

The DMA increases more efficient use of interrupts, increases data throughput & potentially reduces hardware costs by eliminating the need for peripheral specific

fifo buffers.



### • Operation of a DMA transfer.

15. When a DMA module takes control of bus & while it retains control of bus, what does the processor do?

Ans: When the DMA controller takes over the bus, the processor responds with a HLDA signal & remains on hold until the DMA controller executes its task of transferring data from/to peripheral devices to/from main memory. When the data transfer complete, DMA gives out HRQ signal thus returning the control of bus back to - Processor... Throughout the process of data transfer, processor remains idle.

16. A given application written in Java runs 15 sec. on a desktop processor. A new Java compiler is released that requires only 0.6 as many instructions as the old compiler. Unfortunately, it increases the CPI by 1.1. How fast can you expect the Application to run using this new compiler?

Ans. A: 15 seconds =  $Ins_A * CPI_A * ClockRate$ .

$$\text{So, clock rate} = \frac{15 \text{ seconds}}{(Ins_A * CPI_A)}$$

B:  $Time_B = (0.6 * Ins_A) * (1.1 * CPI_A) * ClockRate$ .

$$Time_B = (0.6 * Ins_A) * (1.1 * CPI_A) * \frac{15 \text{ seconds}}{(Ins_A * CPI_A)}$$

$$Time_B = 0.6 * 1.1 * 15 \text{ seconds}$$

$$Time_B = 9.9 \text{ seconds}$$

18. What should happen if the processor issue a request that misses in the cache while a block is being written back to main memory from the write buffer?

Ans. In the case of the write-back protocol, the block containing the addressed word is first brought into the cache, & then the desired word in the cache is overwritten with the new information.

20. When a program generates a Reference to a page that does not reside in the Physical main memory, execution of the program is suspended until the requested page is loaded into the main memory. What difficulties might arise when an instruction in one page has an operand in a different page? What capabilities must the processor have to handle this situation?

Ans. A page fault occurs when some instruction accesses a memory operand that is not in the main memory, resulting in an interruption before the execution of this instruction is completed. Hence, when the task resumes, either the execution of the interrupted instruction must continue from the point of interruption, or the instruction must be restarted. The design of a particular processor dictates which of the options should be used.

17. How many total bits are required for a direct-mapped cache with 16 KB of data & 4 word blocks, assuming a 32-bit address?

Ans. Given, data memory size of cache = 16 KB.  
Assuming the memory to be byte-addressable (i.e. 1 word size = 1 B).

No. of Cache lines (block) =  $16\text{ KB} / 4\text{ B}$

$$= 2^{12} \text{ lines}$$

So, no. of bits needed to represent cache lines = 12 bits.

No. of bits needed to represent a word in a line =  $2^2 = 4$

$$\Rightarrow 2 \text{ bits.}$$

So, no. of bits for tag field is

$$= 32 - 12 - 2 = 18 \text{ bits.}$$

Size of tag memory = No. of tag bits \* No. of lines

$$= 18 * 2^{12} \text{ bits.}$$

$$= 72 \text{ K bits.}$$

& size of data memory = 16 KB.

$$= 128 \text{ K bits.}$$

So, total memory needed for cache

$$= (128 \text{ K} + 72 \text{ K}) \text{ bits}$$

$$= 200 \text{ K bits}$$

Ans.

19. "Having a large no. of process Register makes it possible to reduce the number of memory accesses needed to perform complex tasks." Devise a simple computational task to show the validity of this statement for a

Processors that have four Registers composed to another that has only two Registers.

Ans. The Dot Product program is:

$$\text{Dot Product} = \sum_{i=0}^{n-1} A(i) \times B(i)$$

It below is the program for dot product of two vectors.

MOVE # AVEC, R1	R1 points to vector A.
MOVE # BVEC, R2	R2 points to vector B.
MOVE N, R3	R3 serves as a counter.
CLEAR R0	R0 accumulates the dot product.
LOOP	Compute the dot product of next components.
Multiply (R1)+, R4	Add to previous sum.
Add (R4), R0	Add to previous sum.
Decrement R3	Decrement the counter.
Branch >0 LOOP	Loop again if not done.
MOVE R0, DOTPROD.	Store dot product in memory.

Program for computing the dot product of two vectors.

The dot product program uses five Registers. Instead of using R0 to accumulate the sum, the sum can be accumulated directly into DOTPROD. This means that the ~~last~~ Move instruction in the program can be removed, but DOTPROD is read & overwritten on each pass through the loop, significantly increasing memory accesses.

The four Registers R1, R2, R3 & R4 are still needed to make this program efficient, & they are all used in the loop. Suppose that R1 & R2 are retained as Pointers to the A & B vectors; Counter Register R3 & temporary storage Register R4 could be replaced by memory locations in a 2-Register machine; but the number of memory accesses would increase significantly.

---

X