**7.4, page# 232;**

What are there two different registers (MAR and MDR) associated with memory?

*Solution:*

MAR in processor is actually used to store the address which is taken by PC in processor from the memory. PC in processor actually takes the address from the memory when the processor requires some data for processing. Address which is stored in MAR will be sent to memory along with control signal to memory. Then memory after receiving the address it sends the data present in that address to processor. Then the register MDR in processor is used to store the data which has been sent from the memory. Then data stored in MDR will be sent to IR where it will be decoded such that the processing of data occurs after that in processor.

**7.11, page# 232;**

Suppose that the instruction format for a modified Little Man Computer requires two consecutive locations for each instruction. The high-order digits of the instruction are located in the first mail slot, followed by the low –order digits. The IR is large enough to hold the entire instruction and can be addressed as IR [high] and [IR] low to load it. You may assume that the op code part of the instruction uses IR [high] and that the address is found in IR [low]. Write the fetch-execute cycle for an ADD instruction on this machine.

*Solution:*

* MAR is assigned PC and IR [high] value is assigned to   MDR.
* Increment value of PC by 1.
* Again, MAR is assigned PC and A is assigned MDR.
* At the end, increment value of PC by 1.

The fetch-execute cycle for an ADD instruction on this machine.

PC →MAR

MDR → IR [high]

PC + 1 → PC

PC →MAR

MDR → IR [low]

IR [low] → MAR

MDR + A → A

PC + 1 → PC

**7.12, page# 233;**

The Little Prince Computer (LPC) is a mutant variation on the LMC. The LPC is so named because the differences are a royal plan). The LPC has one additional instruction. The extra instruction requires two consecutive words:

0XX

0YY

This instruction, known as move, moves data directly from location XX to location YY without affecting the value in the accumulator. To execute this instruction, the Little Prince would need to store the XX data temporarily. He can do this by writing the value on a piece of paper and holding it until he retrieves the second address. The equivalent in a real CPU might be called the intermediate address register, or IAR. Write the fetch-execute cycle for the LPC MOVE instruction.

*Solution:*

There are five basic steps involved in the process;

a. Fetch instruction

b. Retrieve data from memory location XX

c. Save the retrieved data in IAR

d. Fetch the next location to get address YY

e. Store the data from IAR to address YY

PC→ MAR

Step 1- MDR → IR

Step 2- IR [add] → MAR

Step 3- MDR → IAR

PC + 1 → PC

PC → MAR

Step4- MDR → IR

IR [add] → MAR

Step 5- IAR → MDR

PC + 1 → PC

**7.13, page# 233.**

Generally, the distance that a programmer wants to move from the current instruction location on a BRANCH ON CONDITION is fairly small. This suggests that it might be appropriate to design the BRANCH instruction is such a way that the new location is calculated relative to the current instruction location. For example, we could design a different LMC instruction 8CX. The C digit would specify the condition on which to branch, and X would be a single-digit relative address. Using 10’s complement, this would allow a branch of -5 to +4 locations from the current address. If we were currently executing this instruction at location 24,803 would cause a branch on negative to location 27. Write a fetch-execute cycle for this exercise, and you may also assume that the complementary addition is handled correctly. The single-digit address X, is still found in IR [address]

*Solution:*

PC→ MAR

MDR→ IR

IR[addr] →PC

BRANCH on Condition

PC →MAR

MDR →IR

If condition false: PC + 1 →PC

If condition true: IR[addr] →PC