CHAPTER 5

Computer Organization

(Solutions to Review Questions and Problems)

Review Questions

Q5-1. The three subsystems are the central processing unit (CPU), the main memory, and the input/output.

Q5-3. The ALU performs arithmetic and logical operations.

Q5-5. The main memory stores data and programs when the program is being executed.

Q5-7. The cache memory provides the CPU with fast access to part of data stored in main memory.

Q5-9. The surface of a magnetic disk is divided into circular rings called tracks. Each track is divided into sections called sectors. The width of a magnetic tape is divided into 9 tracks. The length of the tape may be divided into blocks.

Q5-11. An SCSI (small computer system interface) controller is a parallel interface that provides a daisy chain connection between devices and the buses. The FireWire interface is a high speed serial interface that transfers data in packets. It can use a daisy chain or tree configuration. USB is a serial controller that connects both low and high-speed devices to the computer bus. Multiple devices can be connected to a USB controller.

Q5-13. In the programmed I/O method, the CPU waits for the I/O device. A lot of CPU time is wasted by checking for the status of an I/O operation. In the interrupt-driven I/O method, the I/O device informs the CPU of its status via an interrupt. In direct memory access (DMA), the CPU sends its I/O requests to the DMA controller which manages the entire transaction.

Q5-15. Pipelining allows different types of phases belonging to different cycles to be done simultaneously. Pipelining can increase the throughput of the computer.

Problems

P5-1. We have 64 MB /(4 bytes per word) = 16 Mega words = $16 \times 2^{20} = 2^4 \times 2^{20} = 2^{24}$ words. Therefore, we need 24 bits to access memory words.

P5-3. We need 4 bits to determine the instruction ($2^4 = 16$). We need 4 bits to address a register ($2^4 = 16$). We need 10 bits to address a word in memory ($2^{10} = 1024$). The size of the instruction is therefore (4 + 4 + 10) or 18 bits.

P5-5. The instruction register must be at least 18 bits long (See solution to Exercise 43).

P5-7. The data bus must be wide enough to carry the contents of one word in the memory. Therefore, it must be 18 bits (See Solution to Exercise 43).

P5-9. The control bus should handle all instructions. The minimum size of the control bus is therefore 4 bits ($\log_2 16$) (See Solution to Exercise 43).

P5-11. The address bus uses 10 lines which means that it can address $2^{10} = 1024$ words. Since the memory is made of 1000 words and the system uses shared (memory-mapped I/O) addressing, 1024 − 1000 = 24 words are available for I/O controllers. If each controller has 4 registers, then 24/4 = 6 controllers can be accessed in this system.

P5-13. Table 5.1 shows the instruction codes, the first column is not part of the code; it contains instruction addresses for reference. We type A on the keyboard. The program reads and stores it as we press the ENTER key.

Table 5.1 *Solution to P5-13*

$(00)_{16}$	**$(1FFE)_{16}$**	**// $R_F \leftarrow M_{FE}$, Input A from keyboard to R_F**
$(01)_{16}$	**$(240F)_{16}$**	**// $M_{40} \leftarrow R_F$, Store A in M_{40}**
$(02)_{16}$	**$(1040)_{16}$**	**// $M_{40} \leftarrow R_0$, Load A from M_{40} to R_0**
$(03)_{16}$	**$(A000)_{16}$**	**// $R_0 \leftarrow R_0 + 1$, Increment A**
$(04)_{16}$	**$(A000)_{16}$**	**// $R_0 \leftarrow R_0 + 1$, Increment A**
$(05)_{16}$	$(A000)_{16}$	**// $R_0 \leftarrow R_0 + 1$, Increment A**
$(06)_{16}$	$(2410)_{16}$	**// $M_{41} \leftarrow R_0$, Store The result in M_{41}**
$(07)_{16}$	$(1F41)_{16}$	**// $R_F \leftarrow M_{41}$, Load the result to R_F**
$(08)_{16}$	$(2FFF)_{16}$	**// $M_{FF} \leftarrow R_F$, Send the result to the monitor**
$(09)_{16}$	$(0000)_{16}$	**// Halt**

P5-15. Table 5.2 shows the instructions. The first column is not part of the code; it contains the instruction addresses for reference. First, we type 0 and *n* (*n* has a minimum value of 2) from the key board. The program reads and stores them in registers R_0 and R_1 as we press the ENTER key. We then type the first number and press ENTER. The program stores the first number in register R_2. The program then decrements R_1 twice. We type the second number which is stored in register R_3. The program adds the content of R_2 and R_3 and stores the result in register R_2. The program then compares the value of R_1 with R_0, If they are the same, it displays the result on the monitor and halts; otherwise, it jumps back to the second decrement statement and continues.

Table 5.2 *Solution to P5-15*

$(00)_{16}$	**$(10FE)_{16}$**	**// $R_F \leftarrow M_{FE}$, Input 0 from keyboard to R_0**
$(01)_{16}$	**$(11FE)_{16}$**	**// $R_F \leftarrow M_{FE}$, Input *n* from keyboard to R_1**
$(02)_{16}$	**$(12FE)_{16}$**	**// $R_F \leftarrow M_{FE}$, Input the first number to R_2**
$(03)_{16}$	**$(B100)_{16}$**	**// $R_1 \leftarrow R_1 - 1$ Decrement R_1**
$(04)_{16}$	**$(B100)_{16}$**	**// $R_1 \leftarrow R_1 - 1$ Decrement R_1**
$(05)_{16}$	**$(13FE)_{16}$**	**// $R_F \leftarrow M_{FE}$, Input the next number to R_3**
$(06)_{16}$	**$(3223)_{16}$**	**// $R_2 \leftarrow R_2 + R_3$ Add R_3 to R_2 and store in R_2**
$(07)_{16}$	**$(D104)_{16}$**	**// If R0 $\neq$ R1 the PC = 04, otherwise continue**
$(08)_{16}$	**$(2FF2)_{16}$**	**// $M_{FF} \leftarrow R_2$, Send the result to the monitor**
$(09)_{16}$	**$(0000)_{16}$**	**// Halt**