Q.2 a. Explain with the help of block diagram about the basic functional units used in Computer System.

Answer:

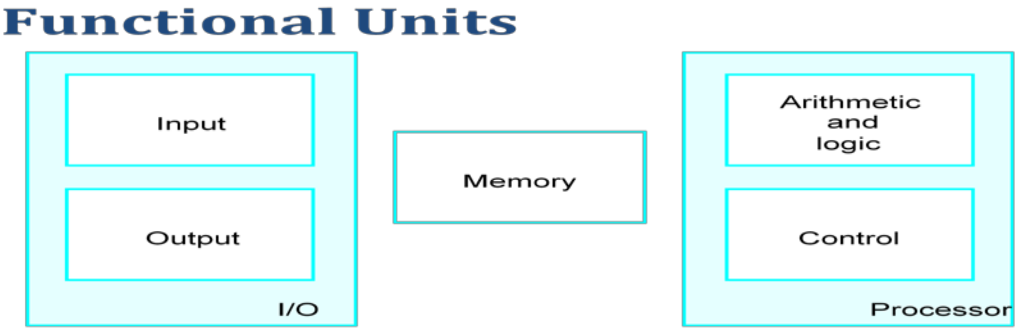


Figure 1. Basic functional units of a computer.

b. Explain the connection between the processor and the memory with the help of a diagram.

Answer:

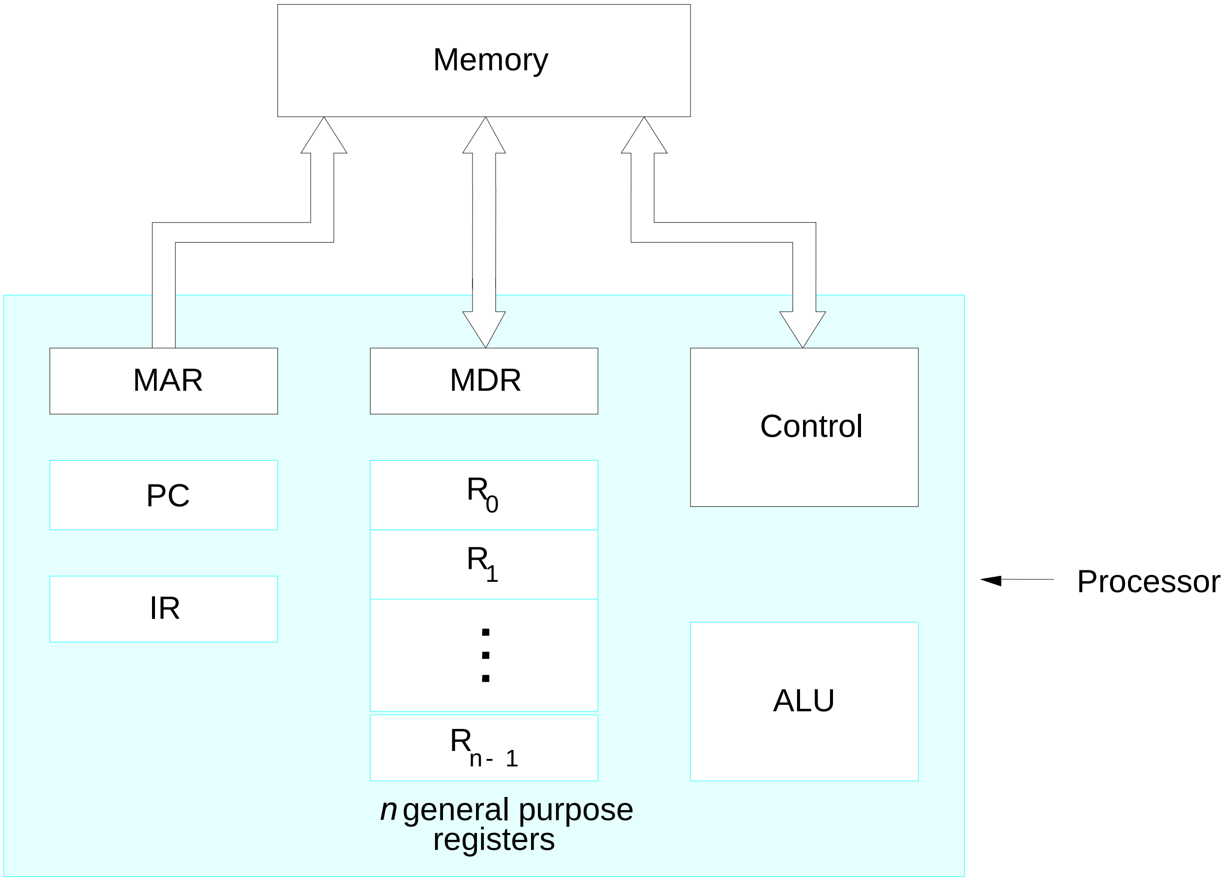


Figure 2. Connections between the processor and the memory.

Q.3 a. Give the difference between Two-address instruction and Three-address instruction, with example and give their execution process.

Answer:

Two-address instruction

Instruction form: **Operation Source, Destination**

- For example, **Add A, B**: performs the operation B[A]+[B].
- When the sum is calculated, the result is sent to the memory and stored in location B
- As another example, **Move B, C**: performs the operation C[B], leaving the contents of location B unchanged

Three-address instruction

- Instruction form: **Operation Source1, Source2, Destination**
- For example, **Add A, B, C**: adds A and B, and the result is sent to the memory and stored in location C
- If k bits are needed to specify the memory address of each operand, the encoded form of the above instruction must contain 3k bits for addressing purposes in addition to the bits needed to denote the Add operation.

b. How Addressing modes of operands are encoded? Explain it in detail.

Answer:

Addressing modes of operands are encoded depends on

* the range of addressing modes

* the degree of independence between opcodes and modes

For **small** number of addressing modes or opcode/addressing mode combinations, the addressing mode can be encoded in **opcode.**

For a **larger** number of combinations, typically a separate **address specifier** is needed for each operand.

The architect has to balance several competing forces when encoding the instruction set:

* The desire to have **as many** registers and addressing modes **as possible**.

* The **impact of the size** of the register and addressing mode fields on the average instruction size and hence on the average program size.

* A desire to have instructions encode into lengths that are **easy to handle** in the implementation (multiples of bytes, fixed-length) with possible sacrificing in average code size.

Q.4 a. What are different modes of operation used in DMA?

Answer:

Modes of operation
Burst mode

An entire block of data is transferred in one contiguous sequence. Once the DMA controller is granted access to the system bus by the CPU, it transfers all bytes of data in the data block before releasing control of the system buses back to the CPU. This mode is useful for loading program or data files into memory, but renders the CPU inactive for relatively long periods of time. The mode is also called **Block Transfer Mode**.

Cycle stealing mode

The *cycle stealing mode* is used in systems in which the CPU should not be disabled for the length of time needed for burst transfer modes. In the cycle stealing mode, the DMA controller obtains access to the system bus the same way as in burst mode, using **BR (Bus Request) and BG (Bus Grant) signals**, which are the two signals controlling the interface between the CPU and the DMA controller. However, in cycle stealing mode, after one byte of data transfer, the control of the system bus is deasserted to the CPU via BG. It is then continually requested again via BR, transferring one byte of data per request, until the entire block of data has been transferred. By continually obtaining and releasing the control of the system bus, the DMA controller essentially interleaves instruction and data transfers. The CPU processes an instruction, then the DMA controller transfers one data value, and so on. On the one hand, the data block is not transferred as quickly in cycle stealing mode as in burst mode, but on the other hand the CPU is not idled for as long as in burst mode. Cycle stealing mode is useful for controllers that monitor data in real time.

Transparent mode

The *transparent mode* takes the most time to transfer a block of data, yet it is also the most efficient mode in terms of overall system performance. The DMA controller only transfers data when the CPU is performing operations that do not use the system buses. It is the primary advantage of the transparent mode that the CPU never stops executing its programs and the DMA transfer is free in terms of time. The disadvantage of the transparent mode is that the hardware needs to determine when the CPU is not using the system buses, which can be complex and relatively expensive.

b. With neat block diagram explain the working of DMA controller.

Answer:
D*irect Memory Access,* a technique for transferring data from main memory to a device without passing it through the CPU. Computers that have DMA channels can transfer data to and from devices much more quickly than computers without a DMA channel can. This is useful for making quick backups and for *real-time* applications.
DMA Controller, a device that can control data transfers between an I/O subsystem and a memory subsystem in the same manner that a processor can control such transfers.

c. Discuss synchronous and asynchronous buses.

Answer:
Synchronous Buses
With a synchronous protocol, data transfers occur in relation to successive edges of the system clock. Inherent in this type of protocol is the assumption that data will arrive within a certain time window (if it does not, then the data is lost).

Asynchronous Buses
Asynchronous bus transfers bear no particular timing relation to the system clock; transfers can take place at any time. Additional handshake lines are required in order to guarantee data transfers between master and slave. Synchronous bus transfers, by way of contrast, only depend on the system clock (the protocol being built into the system).

Q.5 a. How internal organization of memory chips is defined?

Answer:

- Each memory cell can hold one bit of information.
- Memory cells are organized in the form of an array.
- One row is one memory word.
- All cells of a row are connected to a common line, known as the “word line”.
- Word line is connected to the address decoder.
- Sense/write circuits are connected to the data input/output lines of the memory chip.

b. Give difference between static and dynamic RAMs.

Answer:

- Static RAMs (SRAMs):
  - Consist of circuits that are capable of retaining their state as long as the power is applied.
  - Volatile memories, because their contents are lost when power is interrupted.
  - Access times of static RAMs are in the range of few nanoseconds.
  - However, the cost is usually high.

- Dynamic RAMs (DRAMs):
  - Do not retain their state indefinitely.
  - Contents must be periodically refreshed.
  - Contents may be refreshed while accessing them for reading.

Q.6 a. What are the functions of an interface circuit?

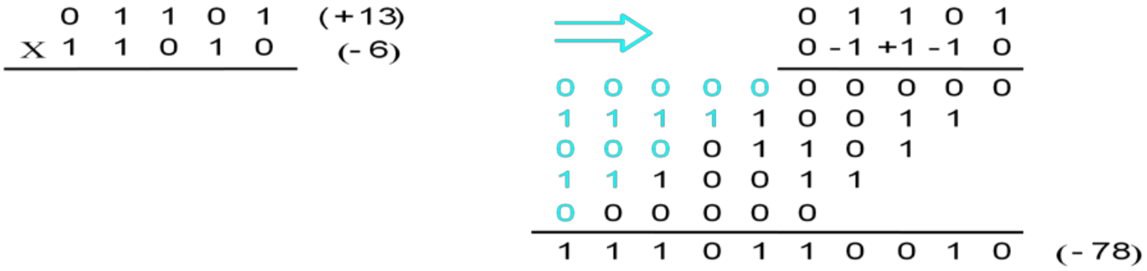
Answer: Page no 248 of Text Book

b. Draw the diagram of a serial interface & explain its working.

Answer: Page no 257 of Text Book

Q7 a. What is Booth's algorithm? Explain it with the help of an example.

Answer:
Booth's multiplication algorithm is a multiplication algorithm that multiplies two signed binary numbers in two's complement notation.



```
    0 1 1 0 1   (+13)                    0  1  1  0  1
  x 1 1 0 1 0   (-6)         ⟹          0 -1 +1 -1  0
  -----------                   -----------------------
                           0 0 0 0 0 0  0  0  0  0
                           1 1 1 1 1 0  0  1  1
                           0 0 0 0 1 1  0  1
                           1 1 1 0 0 1  1
                           0 0 0 0 0 0
                         ------------------------
                           1 1 1 0 1 1  0  0  1  0   (-78)
```

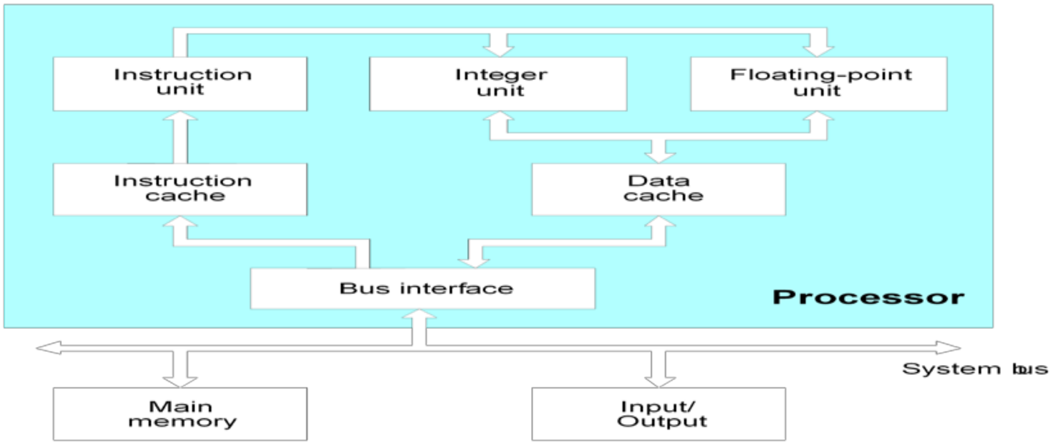
Booth multiplication with a negative multiplier.

b. Explain the multiplication & division of two floating point numbers.

Answer: Page no 376 to 393 of Text Book

Q8. a. Draw the block diagram of the complete processor and explain its working.

Answer:
Block Diagram of complete processor is given below:



b. What are microinstructions? Give an example of partial format for field-encoded microinstructions.

Answer:

- A straightforward way to structure microinstructions is to assign one bit position to each control signal.
- However, this is very inefficient.
- The length can be reduced: most signals are not needed simultaneously, and many signals are mutually exclusive.

  All mutually exclusive signals are placed in the same group in binary coding.

Text Book

Computer Organization, Carl Hamacher, Zvonko Vranesic, Safwat zaky, 5th Edition, TMH, 2002