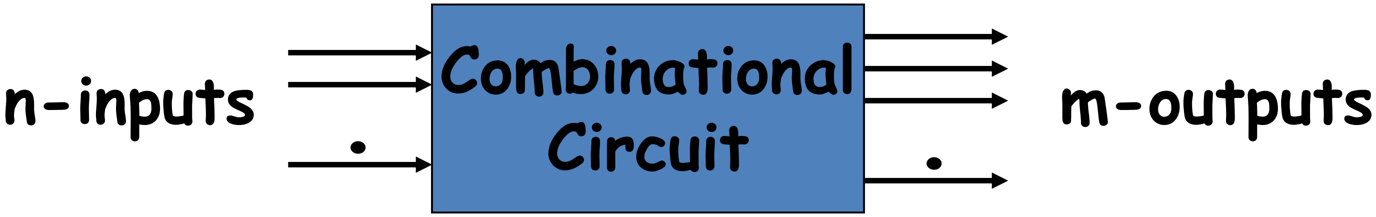
TYPES OF CIRCUITS

- Combinational Circuit
- Sequential Circuit

COMBINATIONAL CIRCUITS

- A combinational circuit consists of logic gates whose outputs, at any time, are determined by combining the values of the inputs.



COMBINATIONAL CIRCUITS : HALF ADDER

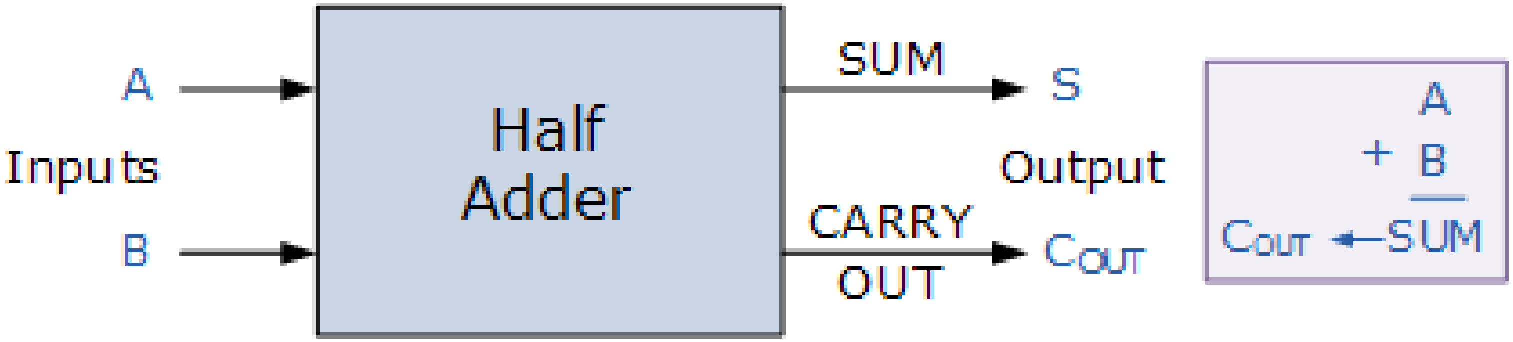
- **Circuit that performs 2 bit addition.**

For the **SUM** bit

SUM = A XOR B = $A \oplus B$

For the **CARRY** bit

CARRY = A AND B = A.B



COMBINATIONAL CIRCUITS : FULL ADDER

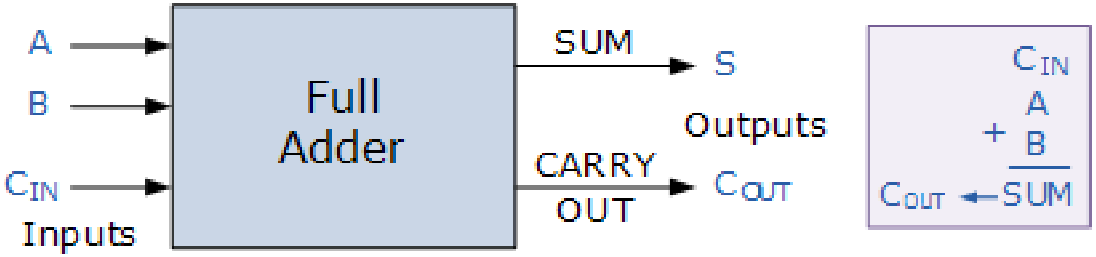
- It performs 3 bit addition. The same two single bit data inputs A and B as before plus an additional Carry-in (C-in) input to receive the carry from a previous stage.

For the **SUM** (S) bit

$$SUM = (A \oplus B) \oplus Cin$$

For the **CARRY-OUT** (Cout) bit

$$CARRY\text{-}OUT = A.B + Cin(A \oplus B)$$

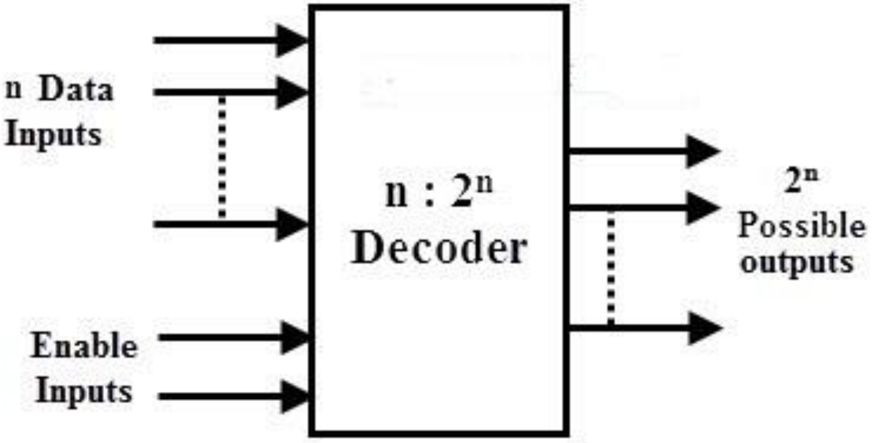


COMBINATIONAL CIRCUITS : BINARY SUBTRACTER

- The Binary Subtractor is another type of combinational arithmetic circuit that is the opposite of the Binary Adder.
- Binary Subtractor is a decision making circuit that subtracts two binary numbers from each other.

COMBINATIONAL CIRCUITS : DECODER

- **Accepts a value and decodes it**
  - **Output corresponds to value of *n* inputs**
- **Consists of:**
  - Inputs (n)
  - Outputs (2^n , numbered from 0 → 2^n - 1)
  - Selectors / Enable (active high or active low)

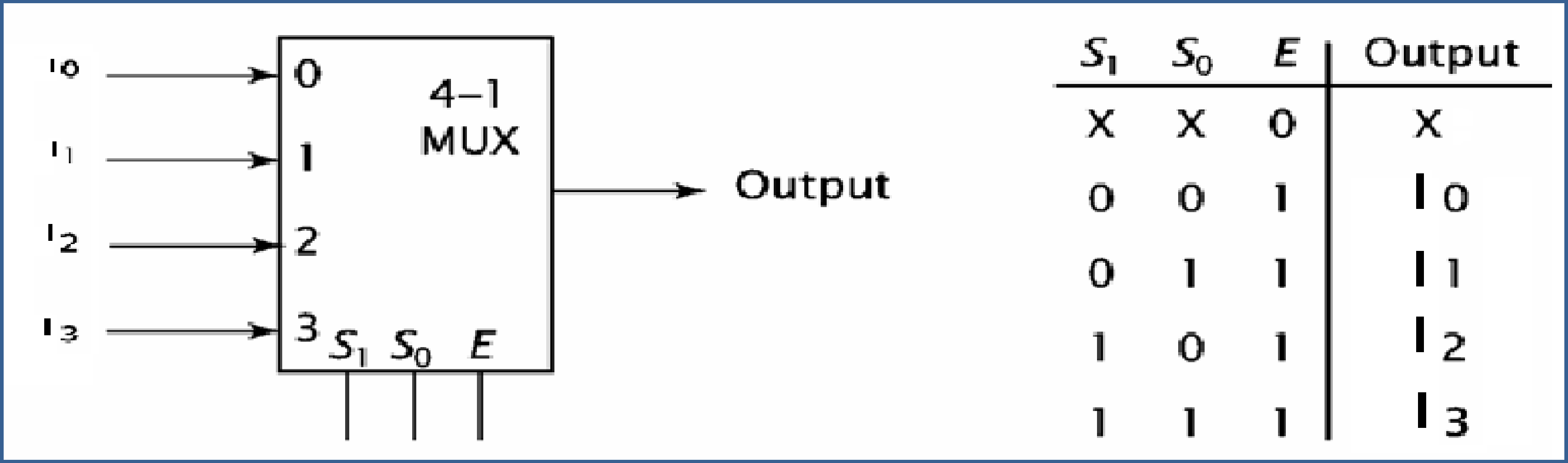


COMBINATIONAL CIRCUITS : ENCODER

- Perform the inverse operation of a decoder
  - 2^n (or less) input lines and n output lines

COMBINATIONAL CIRCUITS : MULTIPLEXER

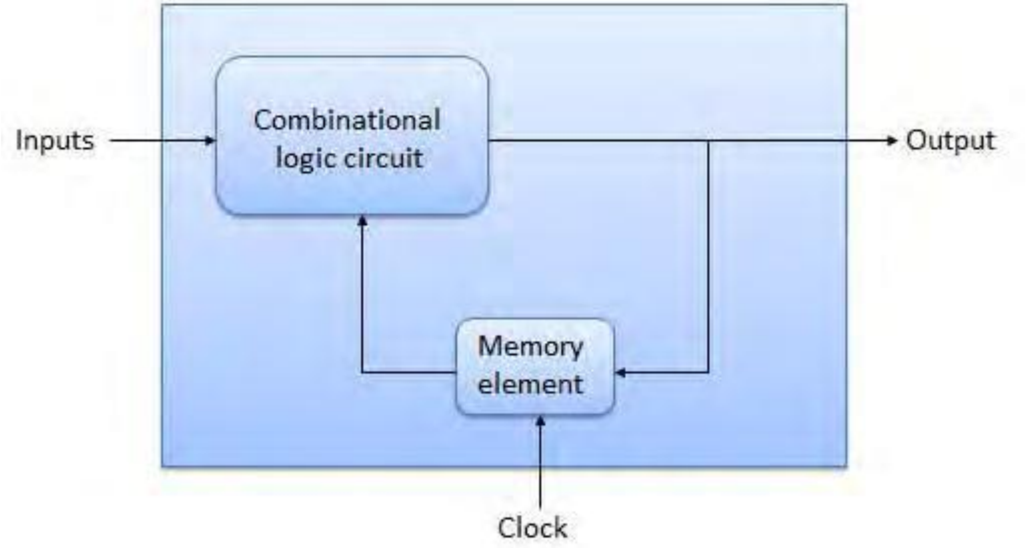
- A multiplexer can use addressing bits to select one of several input bits to be the output.
  - A selector chooses a single data input and passes it to the MUX output
  - It has one output selected at a time.



S_1	S_0	E	Output
X	X	0	X
0	0	1	I_0
0	1	1	I_1
1	0	1	I_2
1	1	1	I_3

SEQUENTIAL CIRCUITS

- Sequential: it consists of memory elements in addition com to logic gates. Their output is function of input and the state of the memory elements. And this is function

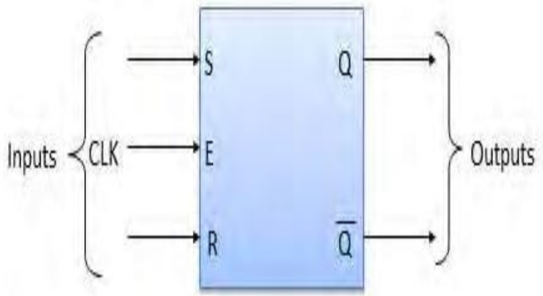


SEQUENTIAL CIRCUITS : FLIP-FLOPS

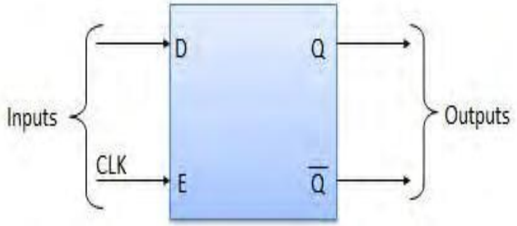
- A flip-flop or latch is a circuit that has two stable states and used to store state information. The circuit can have one or more control inputs and will have one or two outputs.
- It is the basic storage element in sequential logic.
- It is constructed from 2 nand and 2 nor gates.

SEQUENTIAL CIRCUITS : FLIP-FLOPS

RS flip flop: It is an RS flip flop with an invered in the R input.

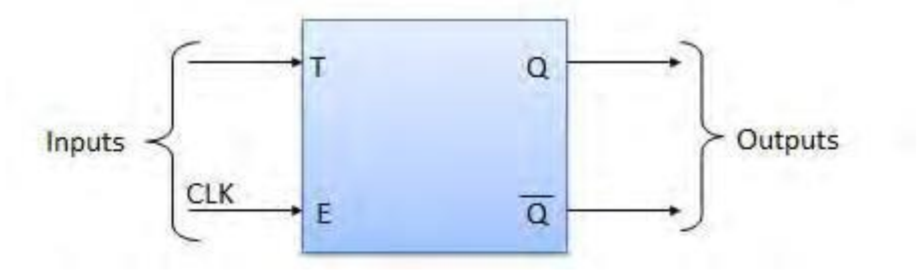


D flip flop: It is an RS flip flop with an inverted in he R input.



SEQUENTIAL CIRCUITS : FLIP-FLOPS

- JK flip flop: It is refinement of RS flip flop. It is same as SR flip flop but contain feedback from the output of second to input of first.
- T flip flop: It is an JK flip flop with an inverted in the K input.



RACE AROUND CONDITION

Race around condition is said to occur when flip-flop changes its state from unknown state to known state and vice versa.

Solution

1) Edge triggering
2) Master Slave

SEQUENTIAL CIRCUITS : REGISTERS

- Registers: It is a group of binary storage cell (flip flop) suitable for holding binary information.

COMPUTER ARCHITECTURE - DEFINITION

Computer Architecture = ISA + MO

- Instruction Set Architecture
  - Logical View
- Machine Organization
  - Physical View

DESIGNING COMPUTERS

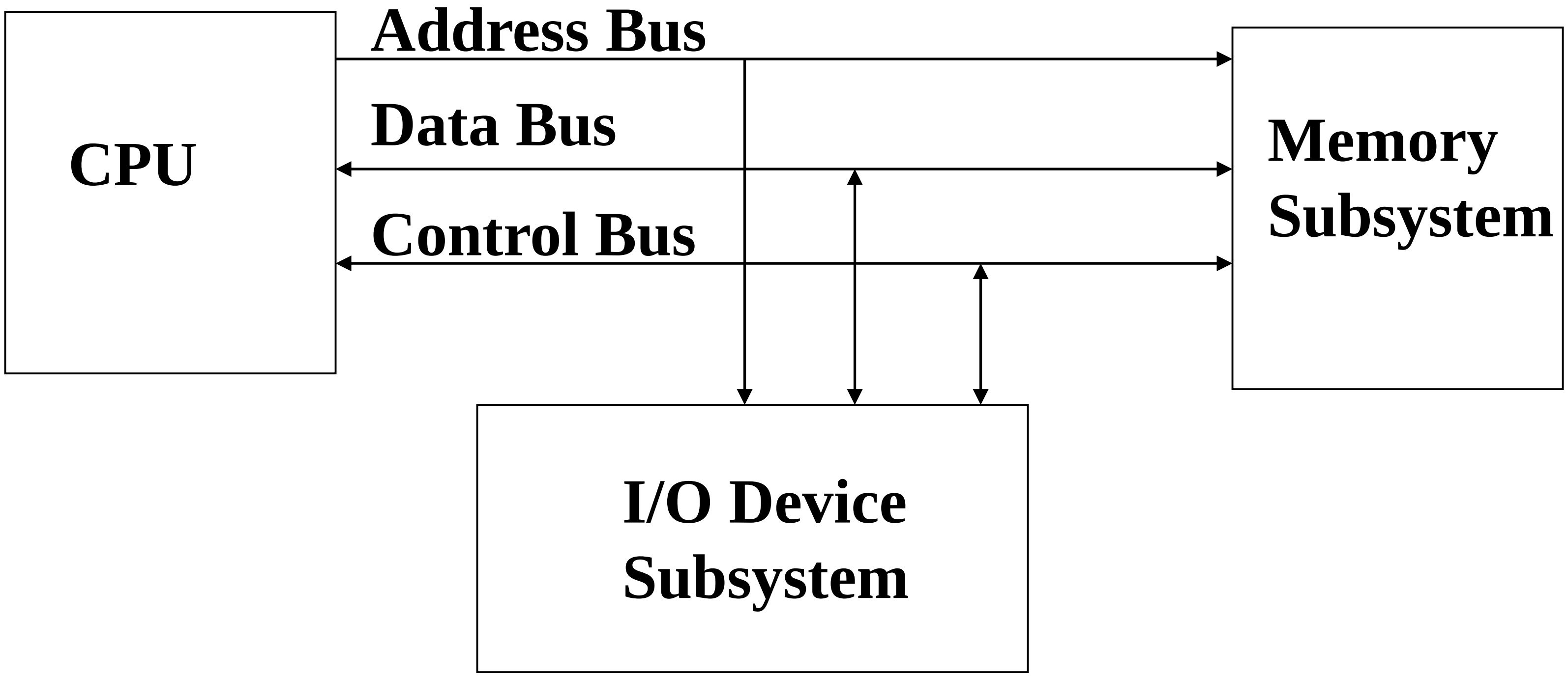
- All computers more or less based on the same basic design, the Von Neumann Architecture!

THE VON NEUMANN ARCHITECTURE

- Model for designing and building computers, based on the following characteristics:
  - Program is stored in memory during execution.
  - Program instructions are executed sequentially.

FUNDAMENTAL COMPONENTS OF COMPUTER

- ➢ The CPU (ALU, Control Unit, Registers)
- ➢ The Memory Subsystem (Stored Data)
- ➢ The I/O subsystem (I/O devices)



MEMORY SUBSYSTEM

Memory, also called RAM (Random Access Memory), When the computer is running, both Program and Data (variables) are stored in the memory.

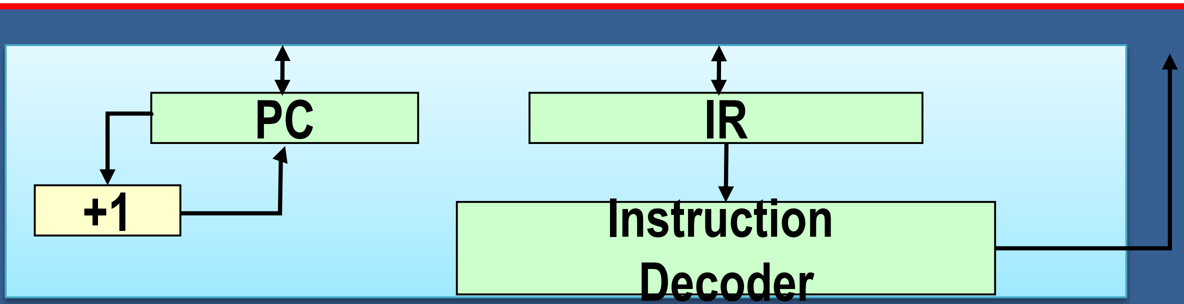
- OPERATIONS ON MEMORY:-
  - Fetch (address):
    - Fetch a copy of the content of memory cell with the specified address.
  - Store (address, value):
    - Store the specified value into the memory cell specified by address.
  - The memory system is interfaced via:
    - Memory Address Register (MAR)
    - Memory Data Register (MDR)
    - Fetch/Store signal

INPUT / OUTPUT SUBSYSTEM

- Handles devices that allow the computer system to:
  - Communicate and interact with the outside world
    - Screen, keyboard, printer, ...
  - Store information (mass-storage)
    - Hard-drives, floppies, CD, tapes, …
- Mass-Storage Device Access Methods:
  - Direct Access Storage Devices (DASDs)
    - Hard-drives, floppy-disks, CD-ROMs, ...
  - Sequential Access Storage Devices (SASDs)
    - Tapes (for example, used as backup devices)

STRUCTURE OF THE CONTROL UNIT

- The function of the control unit is to decode the binary machine word in the IR (Instruction Register) and issue appropriate control signals. These cause the computer to execute its program.
- Types Of Control Unit:-
  - Hardwired
  - Micro-programmed



INSTRUCTION SET DESIGN

- Two different approaches:
  - Reduced Instruction Set Computers (RISC)
    - Instruction set as small and simple as possible.
    - Minimizes amount of circuitry --> faster computers
  - Complex Instruction Set Computers (CISC)
    - More instructions, many very complex
    - Each instruction can do more work, but require more circuitry.

INSTRUCTION CYCLES

- Procedure the CPU goes through to process an instruction.
- 1. Fetch - get instruction
- 2. Decode - interperate the instruction
- 3. Execute - run the instruction.

INTERRUPT

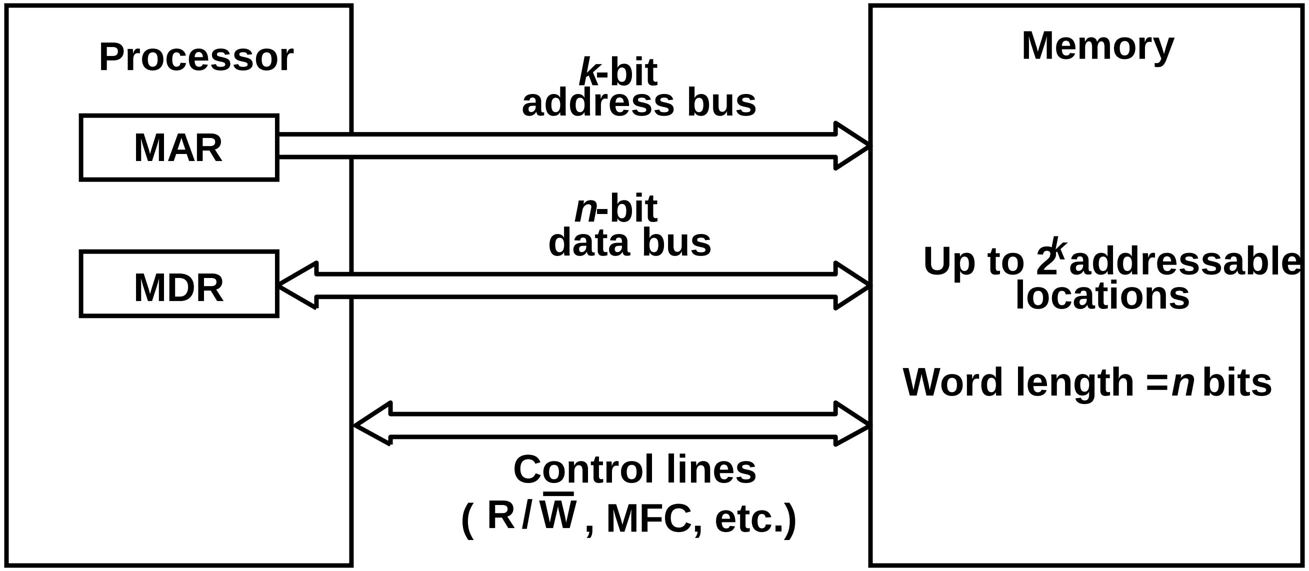
- An interrupt is a signal from a device attached to a computer or from a program within the computer that causes the main program to be suspended temporarily.
- An operating system usually has some code that is called an interrupt handler .
- The interrupt handler prioritizes the interrupts and saves them in a queue if more than one is waiting to be handled. The operating system has another little program, sometimes called a scheduler.

INTERRUPT TYPES

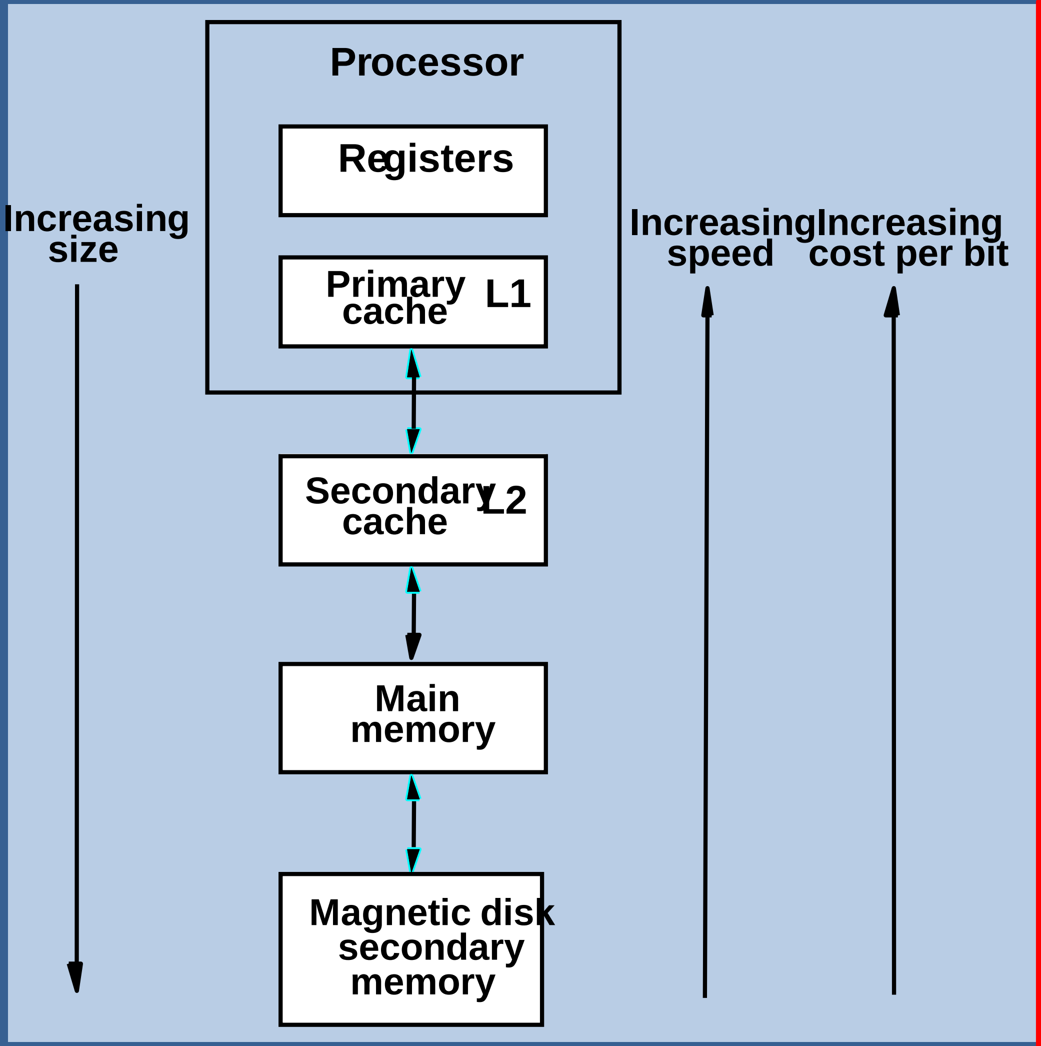
- Hardware Interrupts
  - occurs, for example, when an I/O operation is completed such as reading some data into the computer from a tape drive.
- Software Interrupts
  - occurs when an application program terminates or requests certain services from the operating system

MEMORY ORGANIZATION

➢ CPU-Main Memory Connection



MEMORY HIERARCHY



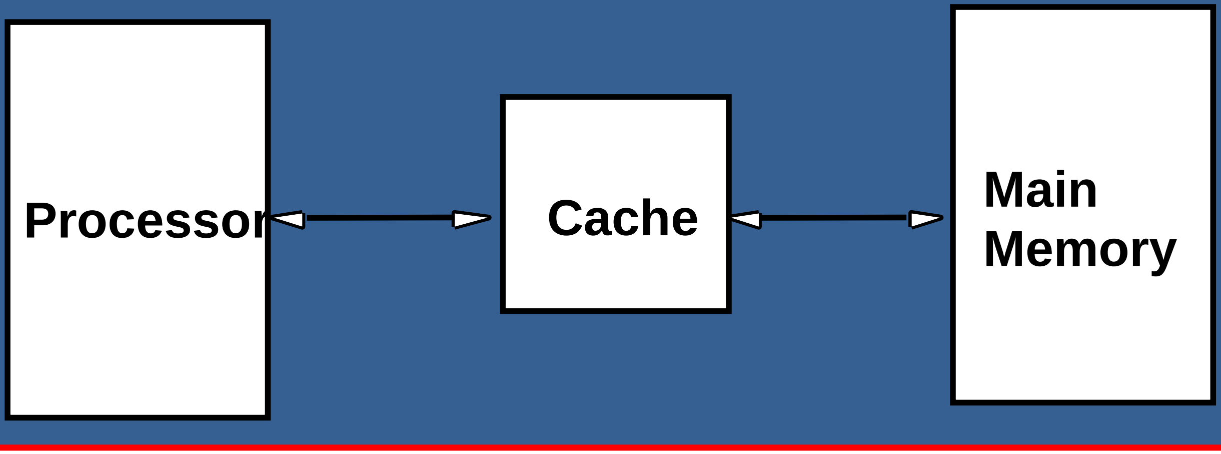
- Fastest access is to the data held in processor registers.
- Relatively small amount of memory that can be implemented on the processor chip. This is processor cache.
- Two levels of cache. Level 1 (L1) cache is on the processor chip. Level 2 (L2) cache is in between main memory and processor.
- Next level is main memory. Much larger, but much slower than cache memory.
- Next level is magnetic disks.

CACHE MEMORIES

- Cache memory is an architectural arrangement which makes the main memory appear faster to the processor than it really is.

- Cache memory is based on the property of computer programs known as "locality of reference".

CACHE MEMORIES



- Processor issues a Read request, a block of words is transferred from the main memory to the cache, one word at a time.
- If the data is in the cache it is called a hit.
- If the data is not present in the cache, then a miss occurs.

MAPPING FUNCTIONS

- Mapping functions determine how memory blocks are placed in the cache.

- Three mapping functions:
  - Direct mapping
  - Associative mapping
  - Set-associative mapping.

ASSOCIATIVE MEMORY

- Memory that is addressed by content rather than by address.
- Content addressable is often used synonymously.

I/O SUBSYSTEM

- Many I/O tasks can be complex and require logic to be applied to the data to convert formats and other similar duties thus resulting CPU more busy. This situation is called 'I/O bound'.
- Solution is DMA
  - feature of computer systems that allows certain hardware subsystems to access main system memory (RAM) independently of the central processing unit (CPU).

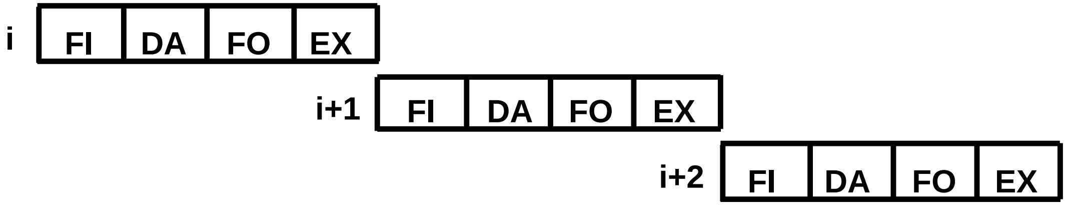
PARALLEL COMPUTERS : CLASSIFICATION

- Flynn's classification
  - Based on the multiplicity of Instruction Streams and Data Streams
  - Instruction Stream
    - Sequence of Instructions read from memory
  - Data Stream
    - Operations performed on the data in the processor

		Number of *Data Streams*	
		Single	**Multiple**
Number of *Instruction Streams*	**Single**	**SISD**	**SIMD**
	Multiple	**MISD**	**MIMD**

PIPELINING

A technique of decomposing a sequential process into sub operations, with each sub process being executed in a partial dedicated segment that operates concurrently with all other segments.

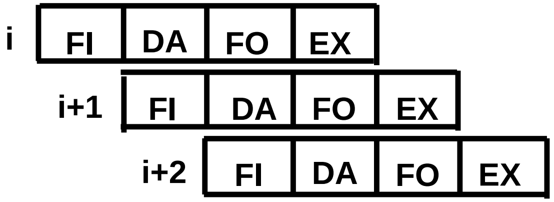


i FI DA FO EX

i+1 FI DA FO EX

i+2 FI DA FO EX

Pipelined



i FI DA FO EX

i+1 FI DA FO EX

i+2 FI DA FO EX

Thank You
shift
{
[
}
]
4
\