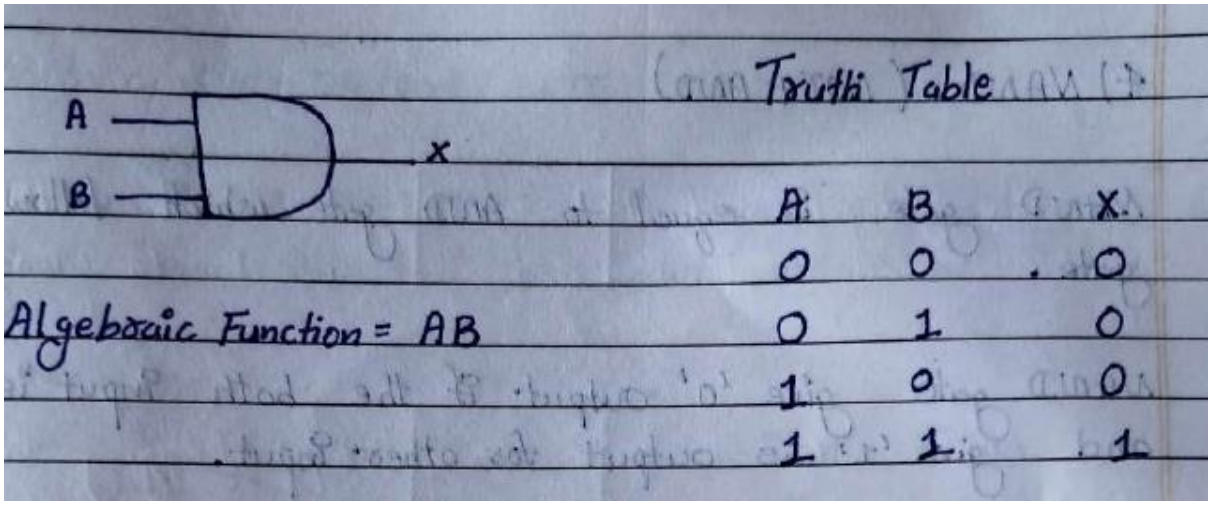
ALU Arithmetic & Logic Unit

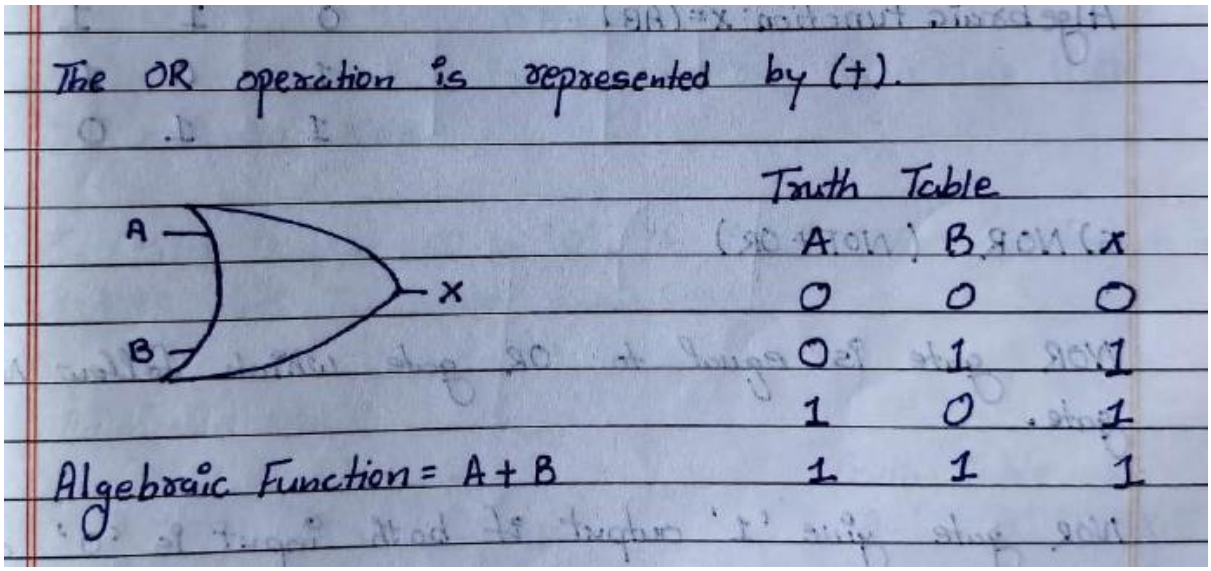
CU Control Unit

Logic Gates

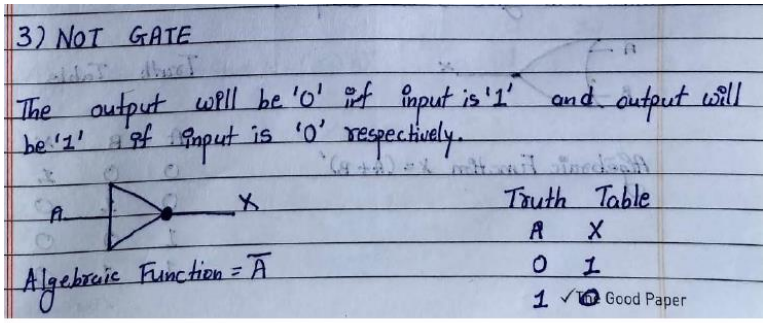
AND (.)



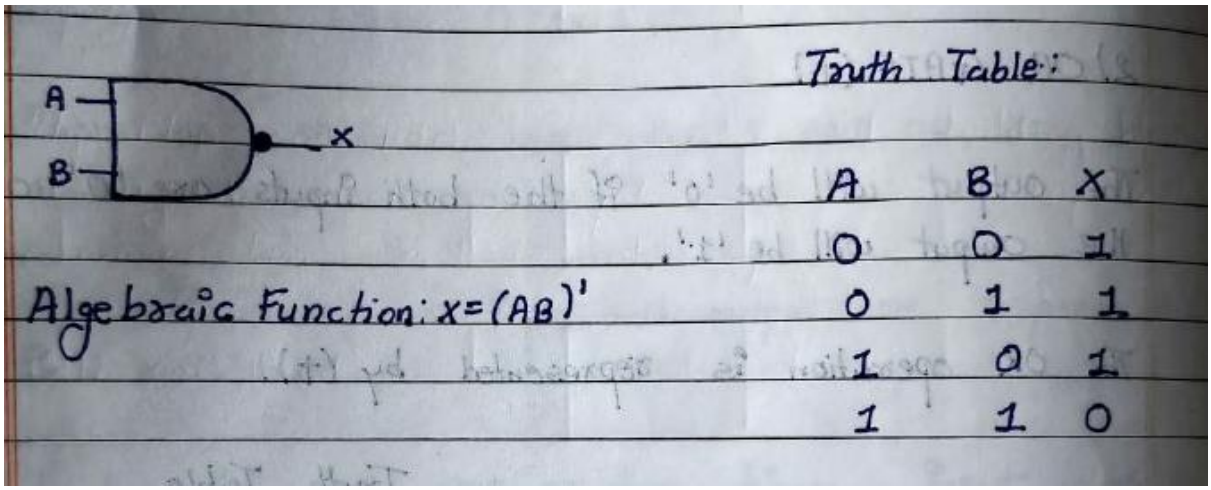
OR(+)



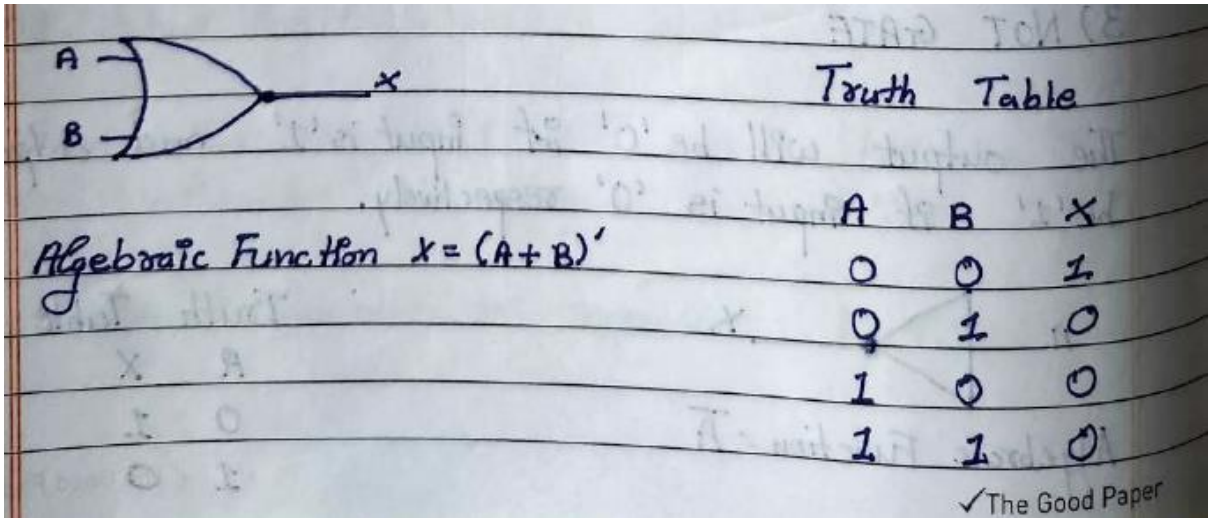
NOT(0-→1 , 1→ 0)



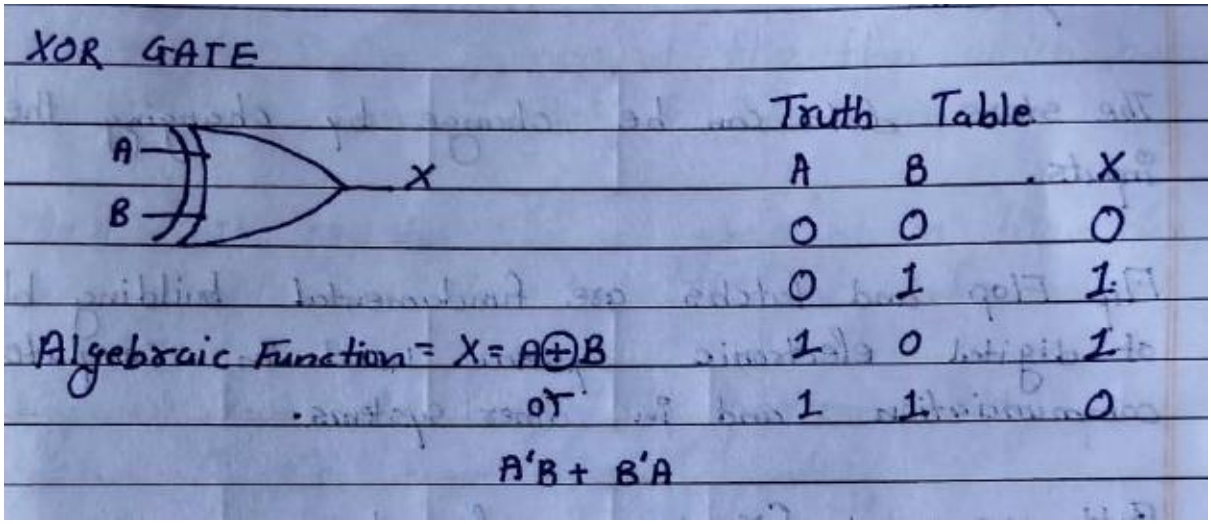
NAND(NOT+AND)



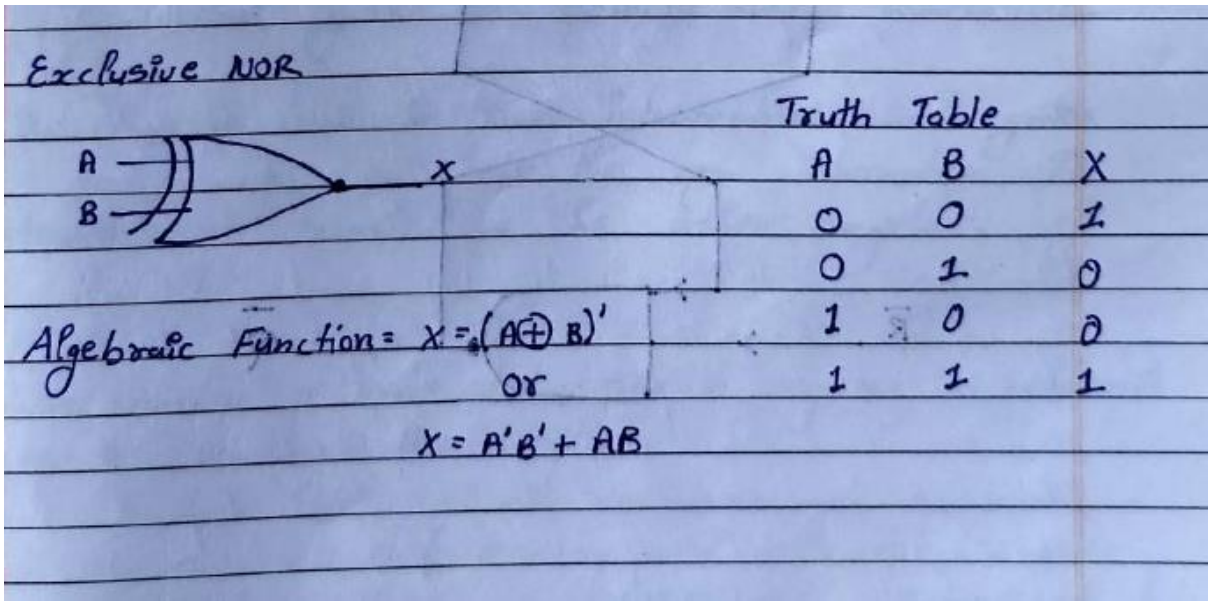
NOR(NOT+OR)



Exclusive OR(or XOR)



Exclusive NOR



Micro-operation(Operation that can be performed on the data stored in register.)

Types of Micro-operation:

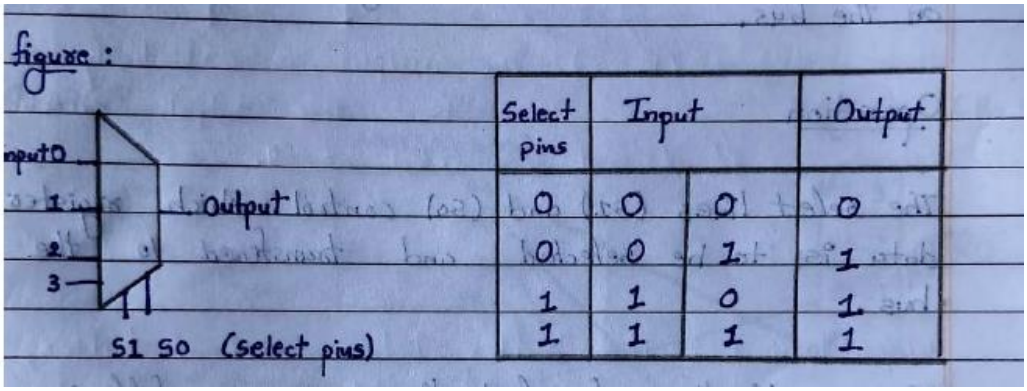
Arithmetic

Logic

Shift

Multiplexer: 2^n inputs has “N” selective lines.

It’s also known as MUX.



Tri state buffer

A tri-state buffer is a type of digital electronic component that can exist in one of three states: high (1), low (0), or high impedance (Z). The high impedance state effectively disconnects the buffer from the circuit, allowing multiple outputs to be connected to a single line without interference. Here's a more detailed explanation:

Tri-State Buffer

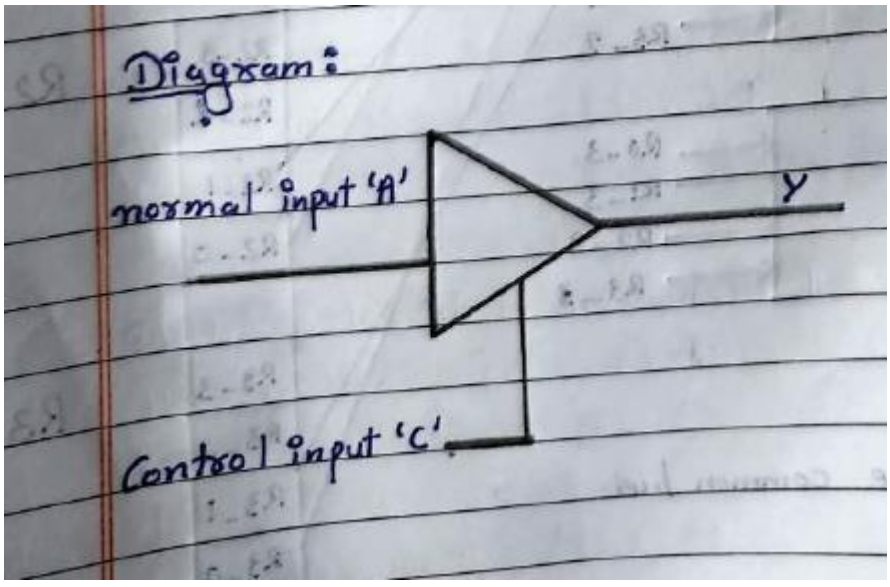
States:

1. **High (1):** The buffer outputs a high voltage level, representing a binary 1.
2. **Low (0):** The buffer outputs a low voltage level, representing a binary 0.
3. **High Impedance (Z):** The buffer is in a high impedance state, essentially disconnected from the circuit, allowing other components to drive the line.

Control Line:

A tri-state buffer has an additional control line (enable line) that determines its output state:

- **Enabled (Control Line Active):** The buffer outputs either high (1) or low (0) based on the input.
- **Disabled (Control Line Inactive):** The buffer goes into high impedance (Z) state, effectively disconnecting from the circuit.



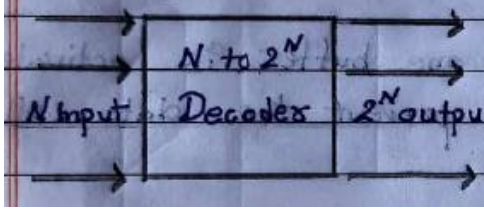
Decoder:

The circuit which change the binary information into 2^N output known as decoder.

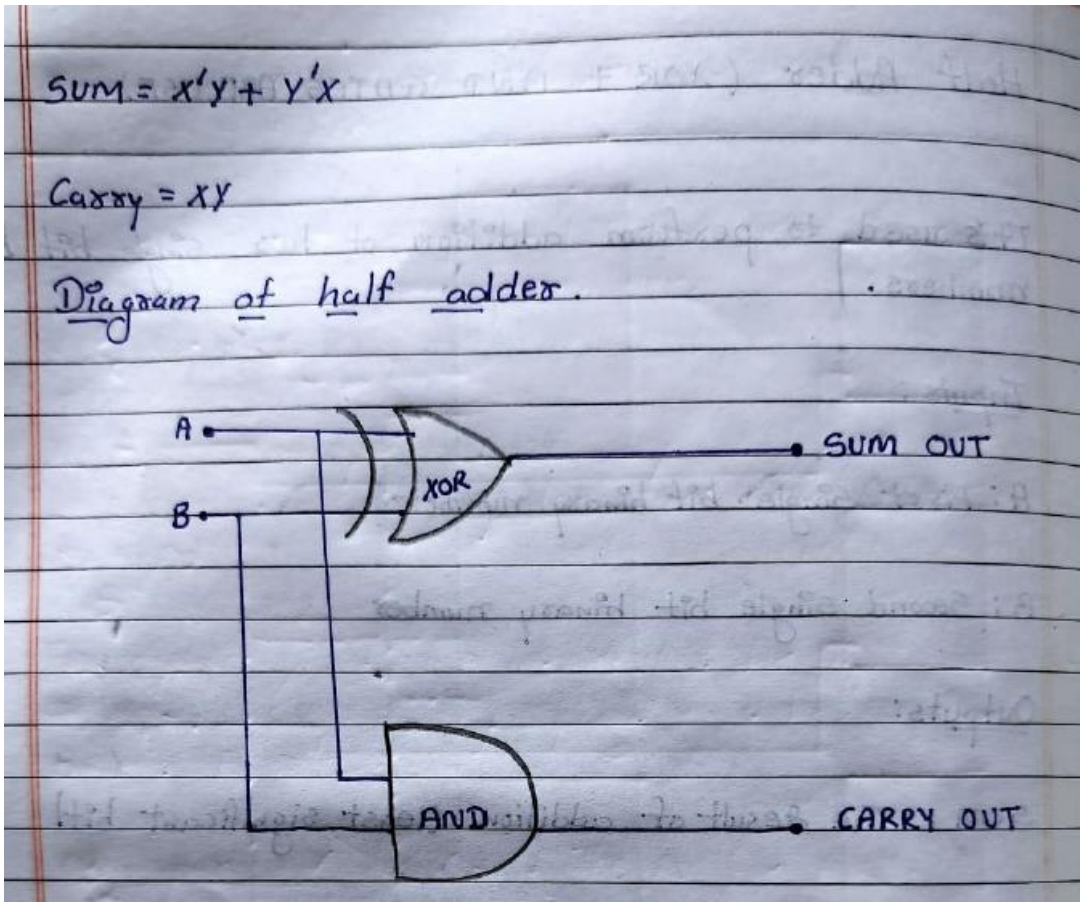
The binary information is passed in the form of N input lines.

In simple word the decoder is reverse of Encoder.

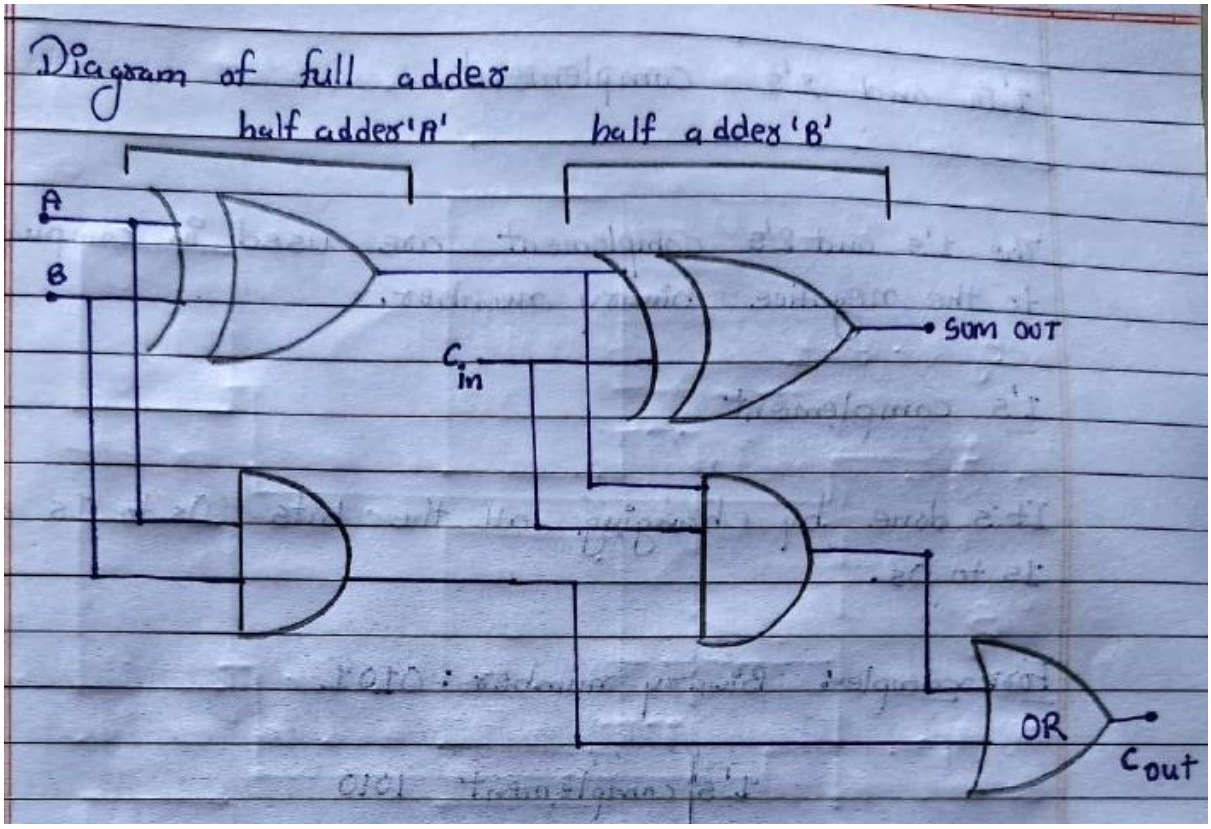
The diagram of decoder is as follow:



Half Adder (XOR+AND GATES ARE USED)

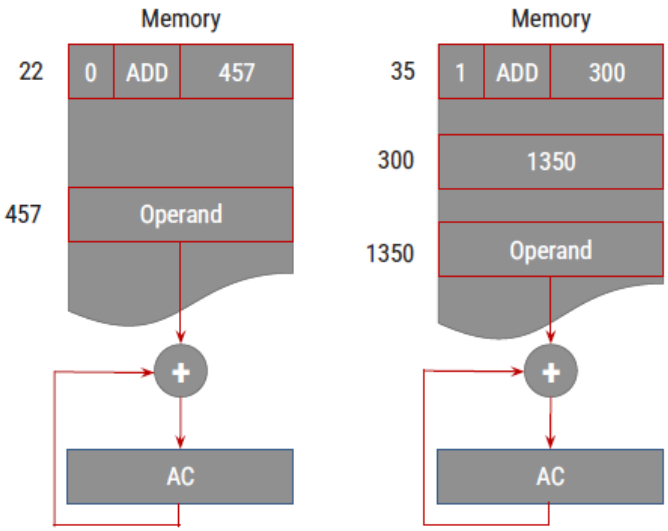


Full Adder (2 Half Adder+ OR gate)



Types of Addressing of Memory

Direct & Indirect Addressing of Memory



- If the second part of an instruction format specifies the address of an operand, the instruction is said to have a **direct address**.
- In **Indirect address**, the bits in the second part of the instruction designate an address of a memory word in which the address of the operand is found.

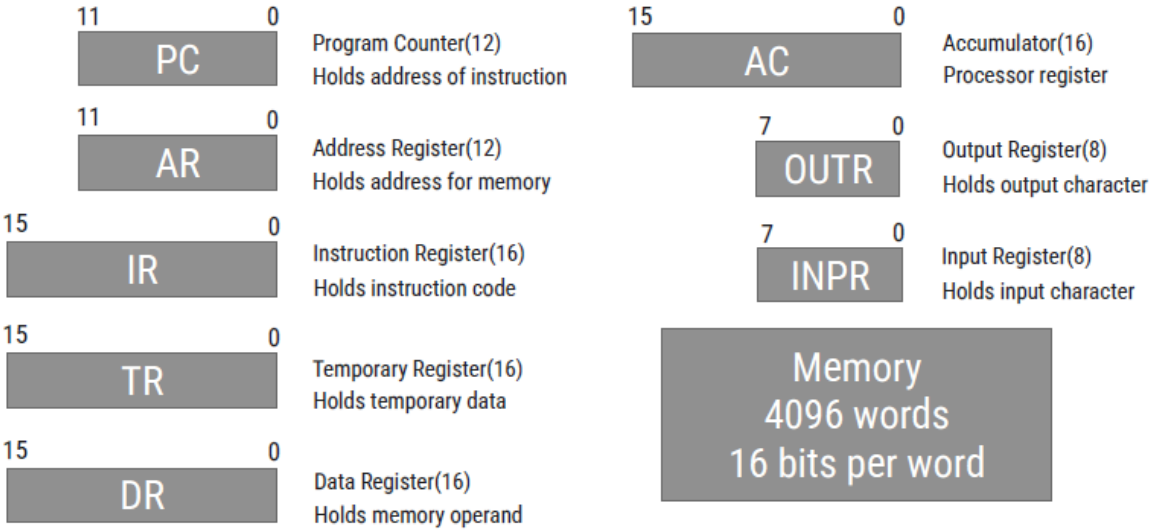
Direct & Indirect Addressing of Memory

- A **direct address** instruction is placed at address 22 in memory.
- The I bit is 0, so the instruction is recognized as a direct address instruction.
- The opcode specifies an ADD instruction, and the address part is the binary equivalent of 457.
- The control finds the operand in memory at address 457 and adds it to the content of AC.

- The instruction in address 35 has a mode bit I = 1, recognized as an **indirect address** instruction.
- The address part is the binary equivalent of 300.
- The control goes to address 300 to find the address of the operand.
- The address of the operand in this case is 1350.
- The operand found in address 1350 is then added to the content of AC.

Computer Registers

Computer Registers



Types of Computer Instruction:

1. Memory Reference Instruction

ADD : add the content of memory to AC

AND: AND the content of memory to AC

LDA: Load memory word to AC

STA: store the content of AC in memory

BUN: Branch Unconditionally

BSA: Branch and Save return address

ISZ: Increment and skip if zero

2. Register Reference Instruction

CLA: Clear AC

CLE: clear E

CMA: complement AC

CME: complement W

CIR: Circulate Right AC and E

CIL: Circulate left AC and E

INC: Increment AC

SPA: skip next instruction if AC is positive

SNA: skip next instruction if AC is negative

SZA: skip next instruction if AC is Zero

SZE: skip next instruction if E is zero

HLT: halt computer

3. Input output instruction

INP: Input character to AC

OUT: Output character to AC

SKI: Skip on input flag

SKO: skip on output flag

ION: interrupt on

IOF: interrupt off

Parts of CPU:

1. Arithmetic Logic Unit (ALU)

Definition: Performs arithmetic and logical operations, such as addition, subtraction, AND, OR, and NOT.

2. Control Unit (CU)

Definition: Directs the operation of the processor by fetching, decoding, and executing instructions from memory.

3. Registers

Definition: Small, high-speed storage locations within the CPU used to temporarily hold data and instructions.

4. Cache Memory

Definition: A small, high-speed memory located close to the CPU to reduce the time needed to access frequently used data and instructions.

5. Bus Interface

Definition: Facilitates communication between the CPU and other components, such as memory and I/O devices.

6. Instruction Decoder

Definition: Interprets binary instructions fetched from memory and translates them into signals that control other parts of the CPU.

7. Floating Point Unit (FPU)

Definition: Handles complex arithmetic operations involving floating-point numbers.

8. Memory Management Unit (MMU)

Definition: Manages memory access and translation of virtual memory addresses to physical addresses.