COMBINATIONAL & ARITHMETIC CIRCUITS

Presented by Nabanita Das

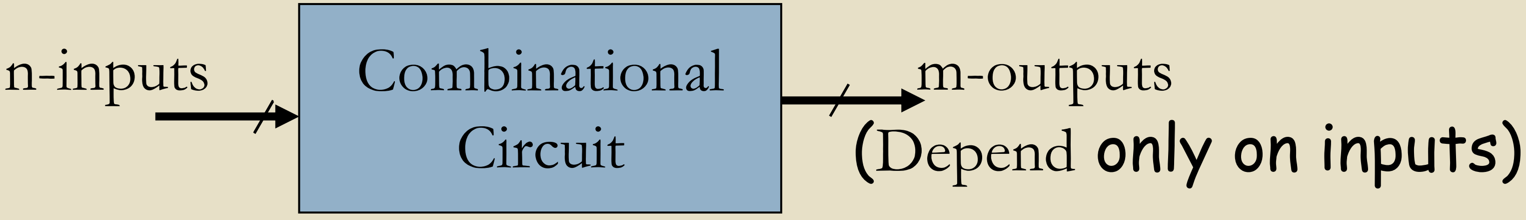
Combinational Circuits

- A combinational circuit consists of logic gates whose outputs, at <u>any time</u>, are determined by combining the values of the inputs.
- For n input variables, there are 2^n possible binary input combinations.
- For each binary combination of the input variables, there is one possible output.

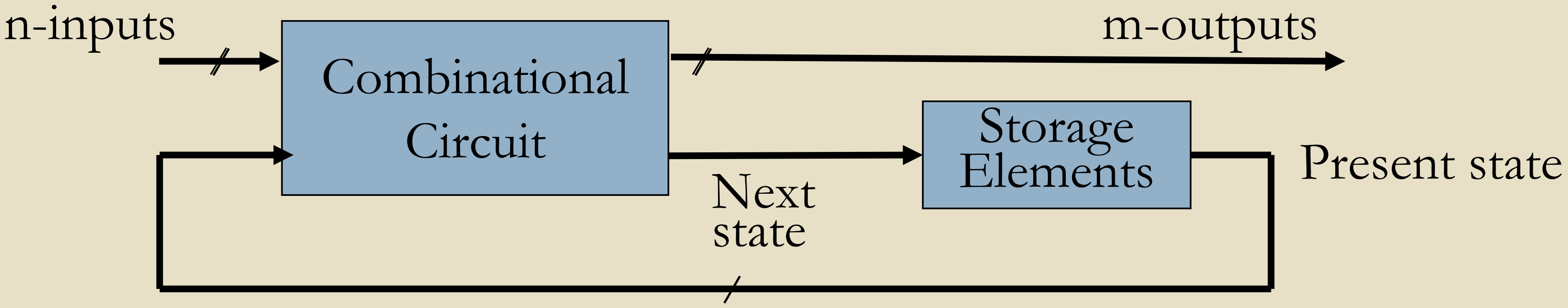
Combinational vs. Sequential Circuits

- Combinational circuits are memory-less. Thus, the output value depends ONLY on the current input values.
- Sequential circuits consist of combinational logic as well as memory elements (used to store certain circuit states). Outputs depend on BOTH current input values and output of previous input values (kept in the storage elements).

Combinational vs. Sequential Circuits



Combinational Circuit



Sequential Circuit

Digital Combinational Logic/Arithmetic Circuits

- Arithmetic
  - Adder/Subtractor
- Converters
  - Decoder/Encoder/Comparator
  - Multiplexer/ Demultiplexer
- Parity Circuits
  - Generators
  - Checkers

Adder

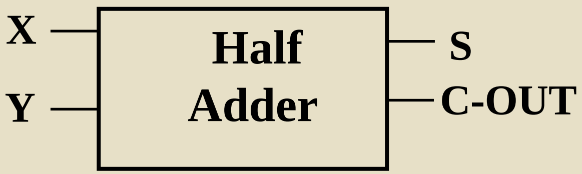
- An adder is a digital logic circuit in electronics that implements addition of numbers. In many computers and other types of processors, adders are used to calculate addresses, similar operations and also in other parts of the processors. These can be built for many numerical representations like binary coded decimal. An adder is a digital circuit that performs addition of numbers.
- Adders are classified into two types:
  - half adder
  - full adder.

Half Adder

The half adder accepts two binary digits on its inputs and produce two binary digits outputs, a sum bit and a carry bit. Here, adding two single-bit binary values, X, Y produces a sum S bit and a carry out C-out bit.

Half Adder Truth Table:

Inputs		Outputs	
X	**Y**	**S**	**C-out**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



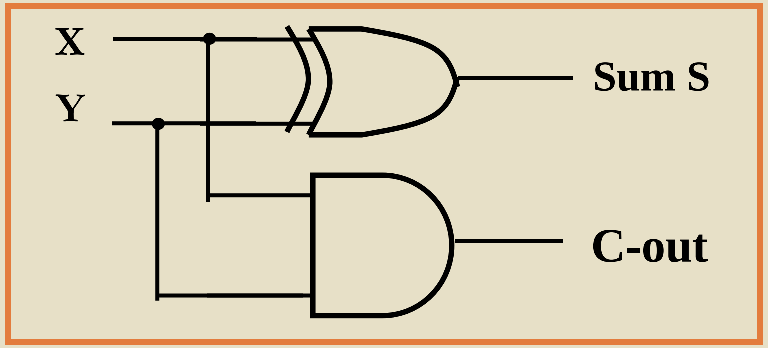
S(X,Y) = Σ (1,2)

S = X'Y + XY'

S = X ⊕ Y

C-out(x, y, C-in) = Σ (3)

C-out = XY



Full Adder

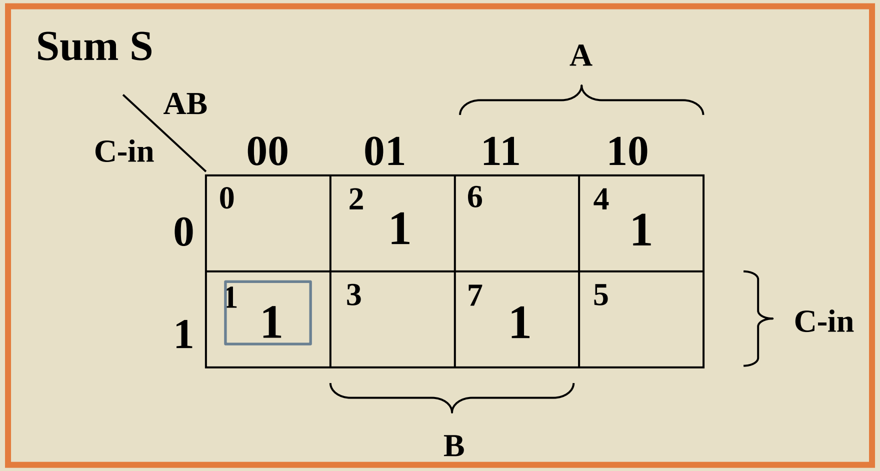
Adding two single-bit binary values, A, B with a carry input bit C-in produces a sum bit S and a carry out C-out bit.

Full Adder Truth Table

Inputs			Outputs	
A	B	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

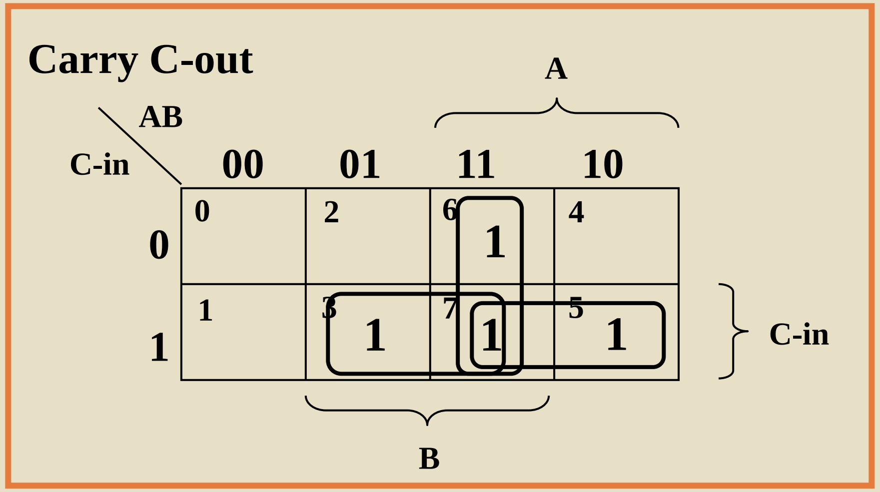
S(A,B, C-in) = Σ (1,2,4,7)

C-out(x, y, C-in) = Σ (3,5,6,7)



S = A’B’(C-in) + A’B(C-in)’ + AB’(C-in)’ + AB(C-in)

S = A ⊕ B ⊕ (C-in)



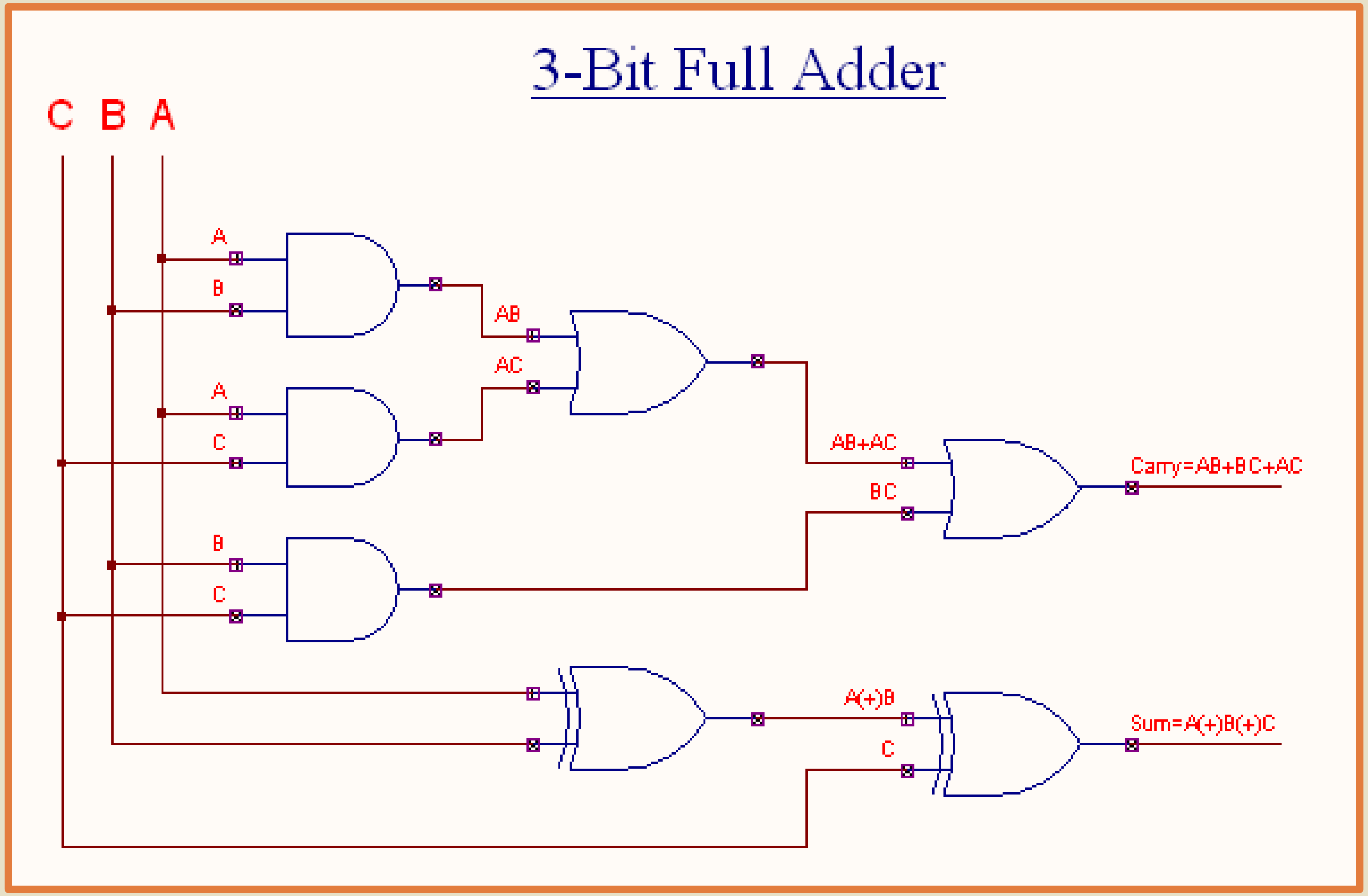
C-out = AB + A(C-in) + B(C-in)

Full Adder Circuit

- The full adder will take three inputs named as A, B, C_{in} then it will give two outputs named as Sum, Carry out.
- **S = A ⊕ B ⊕ (C-in)**
- **C-out = A.B + A(C-in) + B(C-in) is**

equivalent to (A ⊕ B) (Cin) + AB

<u>It is explain in my next slide.</u>



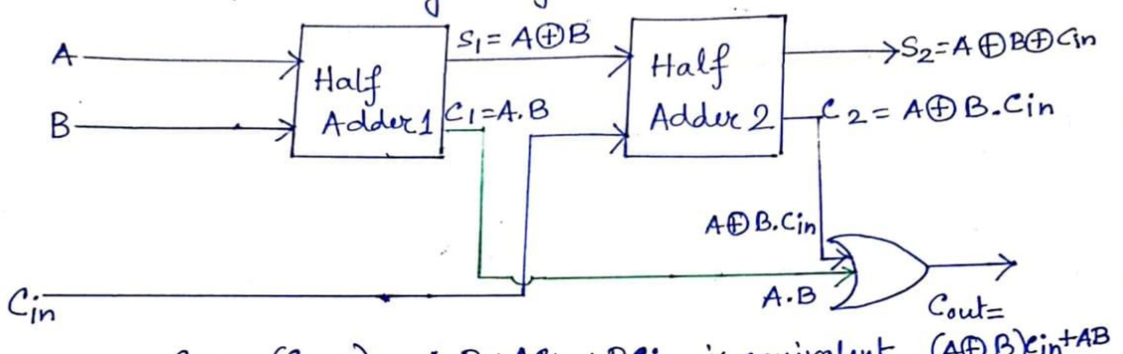
Full Adder Using Half Adder

S = A’B’(Cin) + A’B(Cin)’ + AB’(Cin)’ + AB(Cin)

S = A ⊕ B ⊕ (Cin)

Cout = AB + A(Cin) + B(Cin) is equivalent to
(A ⊕ B) (Cin) + AB

Full Adder Using Half Adder



Carry (C_{out}) $= A.B + AC_{in} + BC_{in}$ is equivalent to $(A \oplus B)C_{in} + AB$.

Proof → $(A \oplus B)C_{in} + AB$

$= (\bar{A}B + A\bar{B})C_{in} + AB$

$= \bar{A}BC_{in} + A\bar{B}C_{in} + AB$

$= B(A + \bar{A}C_{in}) + A\bar{B}C_{in}$

$= B(A + C_{in}) + A\bar{B}C_{in}$

$= B.A + B.C_{in} + A.\bar{B}.C_{in}$

$= B.A + C_{in}(B + A\bar{B})$

$= B.A + C_{in}(B + A)$

$= A.B + C_{in}(A + B)$

$= AB + AC_{in} + BC_{in}$

Proved.

∵ Distributive Law $A + BC = (A+B)(A+C)$

∴ $A + \bar{A}C_{in} = (A + \bar{A})(A + C_{in}) = (A + C_{in})$

∴ $B + A\bar{B} = (B + A)(B + \bar{B}) = (B + A) \cdot 1$.

Full Adder using NAND GATE

- **S = A’B’(C-in) + A’B(C-in)’ + AB’(C-in)’ + AB**
- **S = A ⊕ B ⊕ (C-in)**
- **C-out = AB + A(C-in) + B(C-in) is equivalent**

  to (A ⊕ B) (Cin) + AB