CASPER

CS230: Digital Logic Design and Computer Architecture

Lecture 3: Combinational Circuits

https://www.cse.iitb.ac.in/~biswa/courses/CS230/main.html

Phones (smart/non-smart) on silence plz, Thanks

Logistics

- Join Piazza now
- Lab-1 will be up soon (mostly on Friday)
- Do come with your queries for Monday's lab.
- Tutorial for selected students on Monday 4:20 PM

- Problem set 1 is up. Ungraded, for your practice only
- Detailed course content is at the ASC.
- Email me: biswa@cse.iitb and not on @iitb

Combinational Circuits

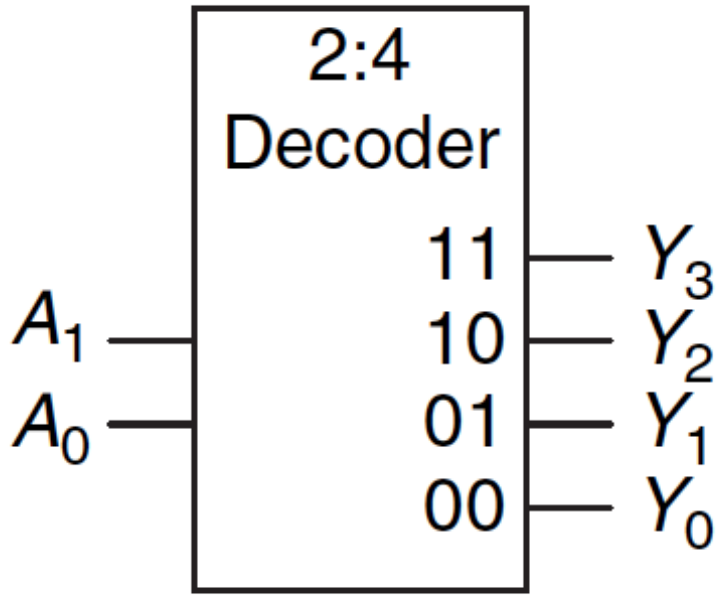
- Combinational logic is often grouped into larger building blocks to build more complex systems
- Hides the unnecessary gate-level details to emphasize the function of the building block
- Output is only dependent on the input
- We now examine:
  - Decoder
  - Multiplexer
  - Full adder

Decoder

- “Input pattern detector”
- n inputs and 2^n outputs
- Exactly one of the outputs is 1 and all the rest are 0s
- The output that is logically 1 is the output corresponding to the input pattern that the logic circuit is expected to detect
- Example: 2-to-4 decoder

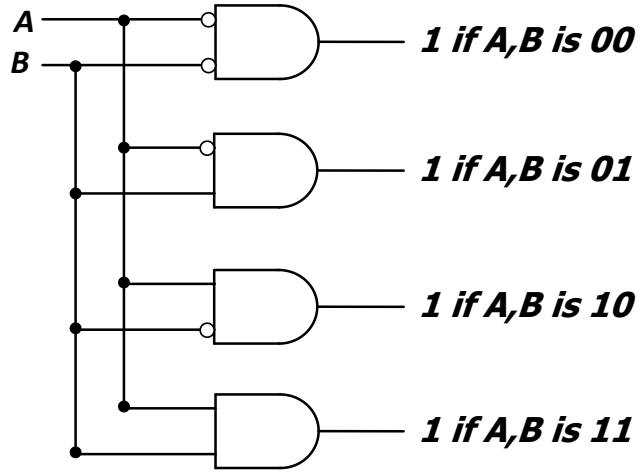
The complement of a decoder is encoder

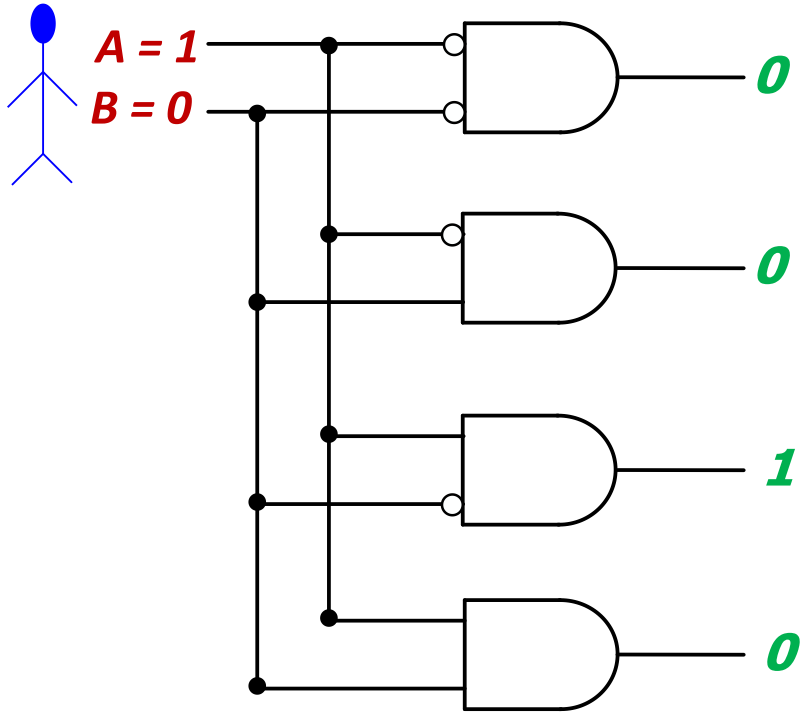
A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0



Contd.

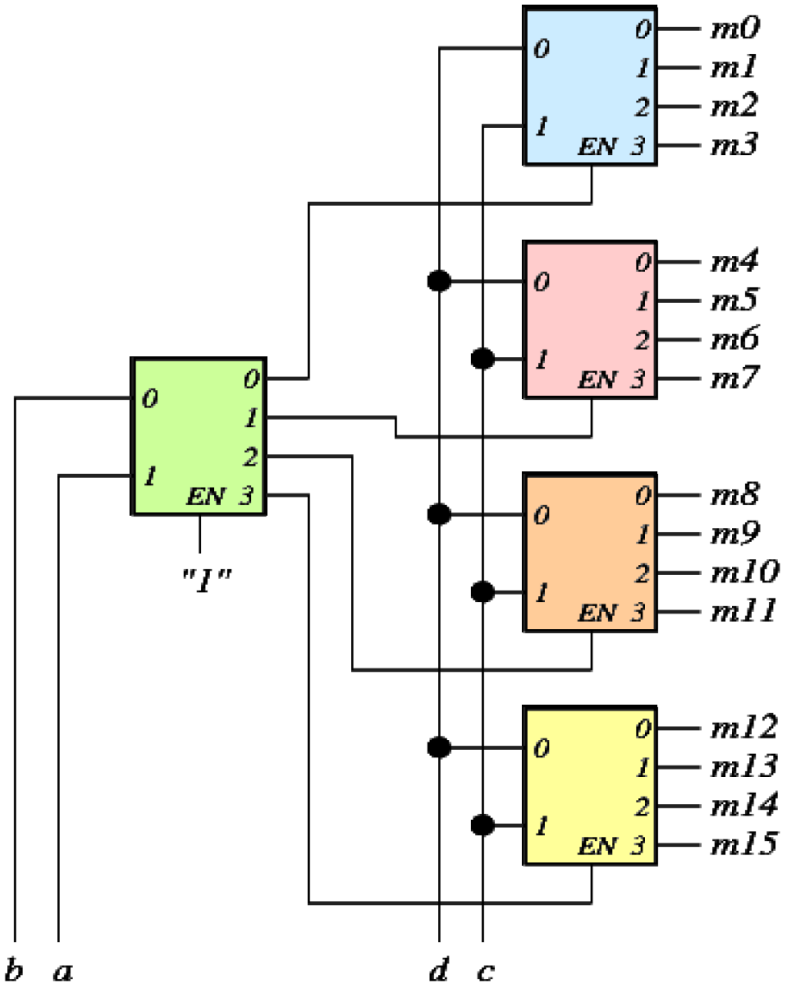
- n inputs and 2^n outputs
- Exactly one of the outputs is 1 and all the rest are 0s
- The output that is logically 1 is the output corresponding to the input pattern that the logic circuit is expected to detect
- Mostly decoders have an enable signal. Output is only generated if the enable is high.





- Build a 4x16 decoder using 2x4 decoders (decoder tree)..

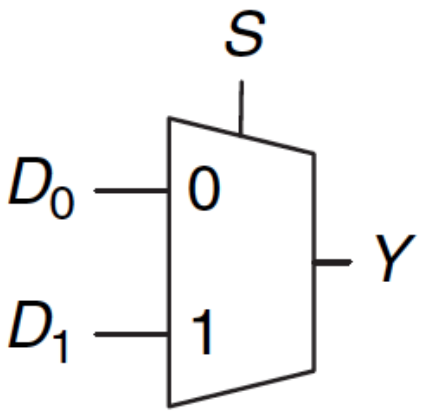
a	*b*	*c*	*d*	*EN*	*minterm*
0	0	0	0	1	m0
0	0	0	1	1	m1
0	0	1	0	1	m2
0	0	1	1	1	m3
0	1	0	0	1	m4
0	1	0	1	1	m5
0	1	1	0	1	m6
0	1	1	1	1	m7
1	0	0	0	1	m8
1	0	0	1	1	m9
1	0	1	0	1	m10
1	0	1	1	1	m11
1	1	0	0	1	m12
1	1	0	1	1	m13
1	1	1	0	1	m14
1	1	1	1	1	m15
X	X	X	X	0	0



Multiplexer

- Selects one of the *N* inputs to connect it to the output
  - based on the value of a $\log_2 N$-bit control input called select
- Example: 2-to-1 MUX

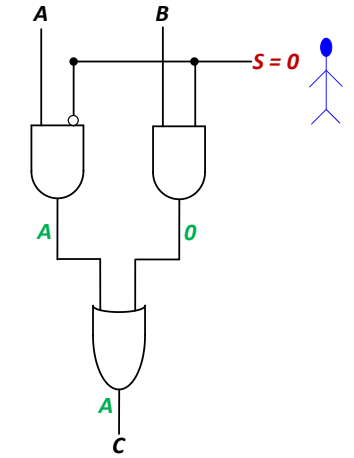
- When S is 0, Y is D_0 and when S is 1, Y is D_1

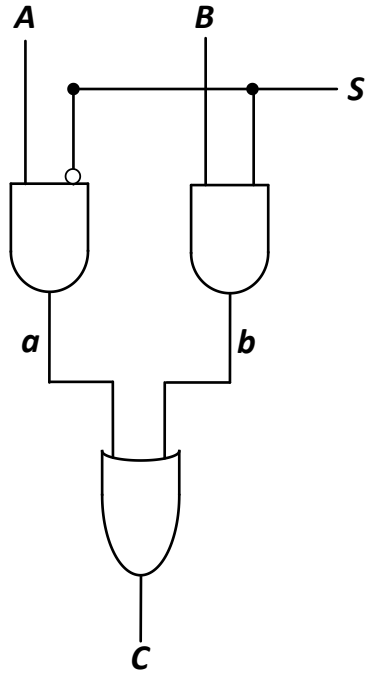


S	D_1	D_0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

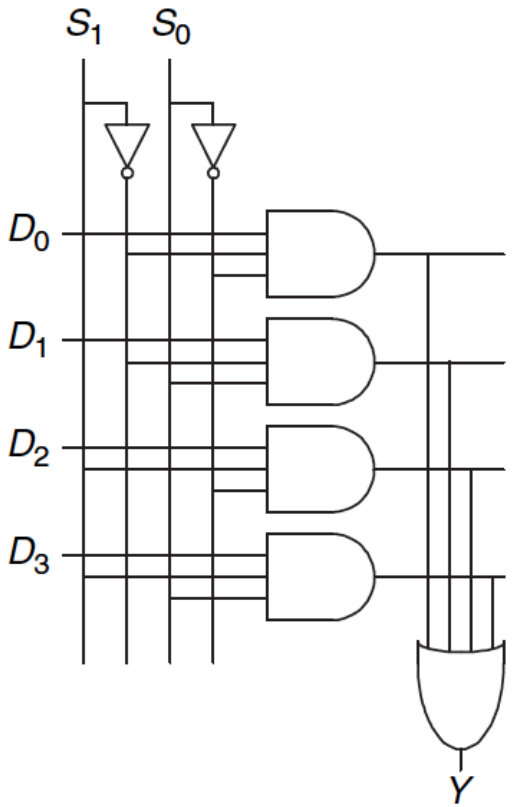
Contd.

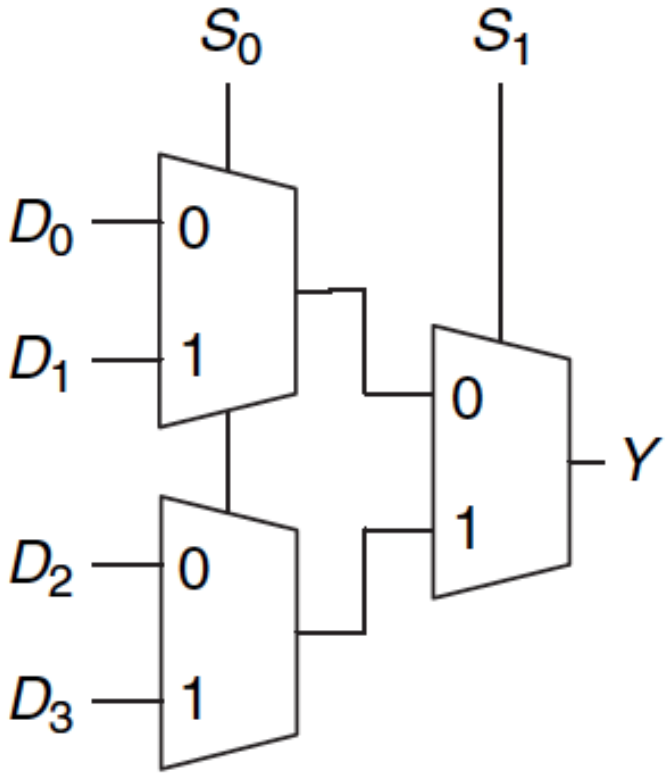
- Selects one of the *N* inputs to connect it to the output
  - based on the value of a $\log_2 N$-bit control input called select
- Example: 2-to-1 MUX





From 2 to 4:1 MUX

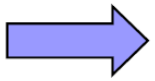




Adder

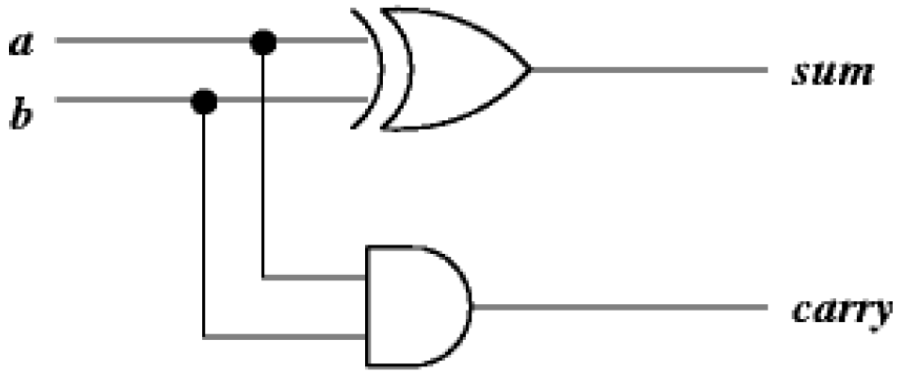
- The most basic arithmetic operation in a digital computer is addition.
- Half Adder is a combination circuit that performs addition of 2 bits.

Inputs		Outputs	
a	*b*	*Carry*	*Sum*
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



$$Sum = \bar{a}b + a\bar{b} = a \oplus b$$
$$Carry = ab$$

Half adder



$$Sum = \bar{a}b + a\bar{b} = a \oplus b$$
$$Carry = ab$$

- Half adders cannot accept a carry input and hence it is not possible to cascade them to construct an n-bit binary adder.

Full adder

- Full Adder is a combinational circuit that forms the arithmetic sum of three input bits. It is described by the following truth table:

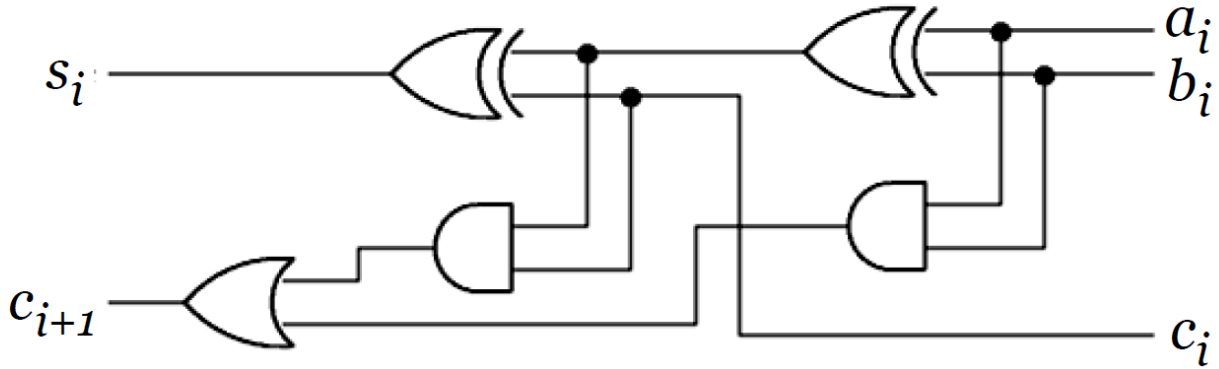
Inputs			Outputs	
c	b	a	C_{out}	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$Sum = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc = a \oplus b \oplus c$$

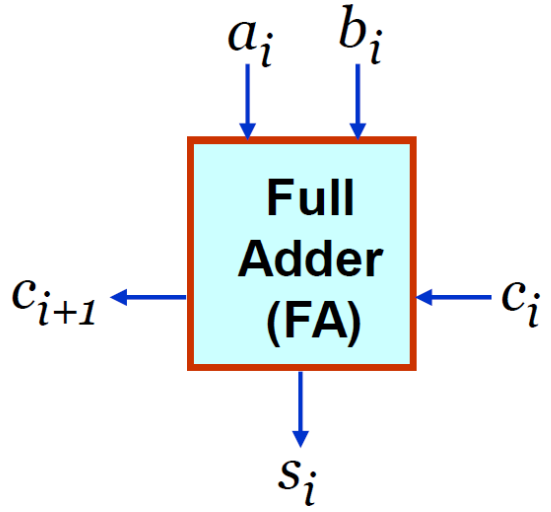
$$C_{out} = ab + ac + bc = ab + c(a + b)$$

Implementation

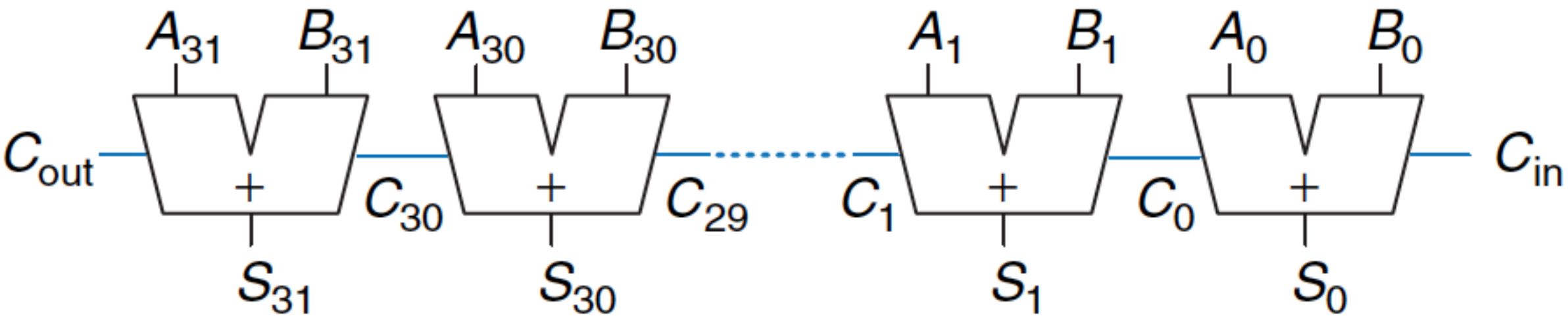
$$Sum = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc = a \oplus b \oplus c$$
$$C_{out} = ab + ac + bc = ab + c(a+b)$$



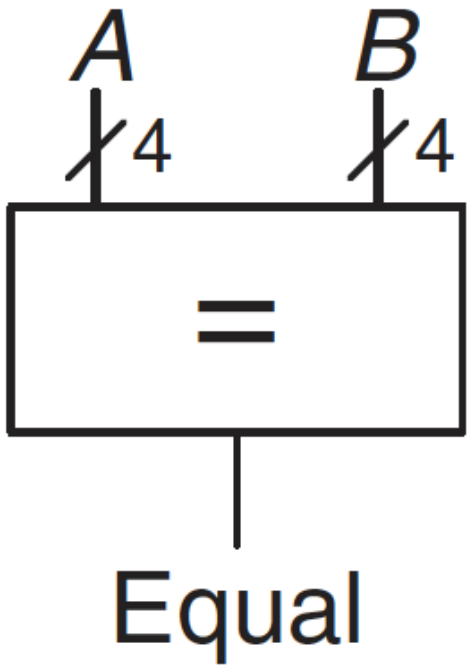
Full Adder at bit i

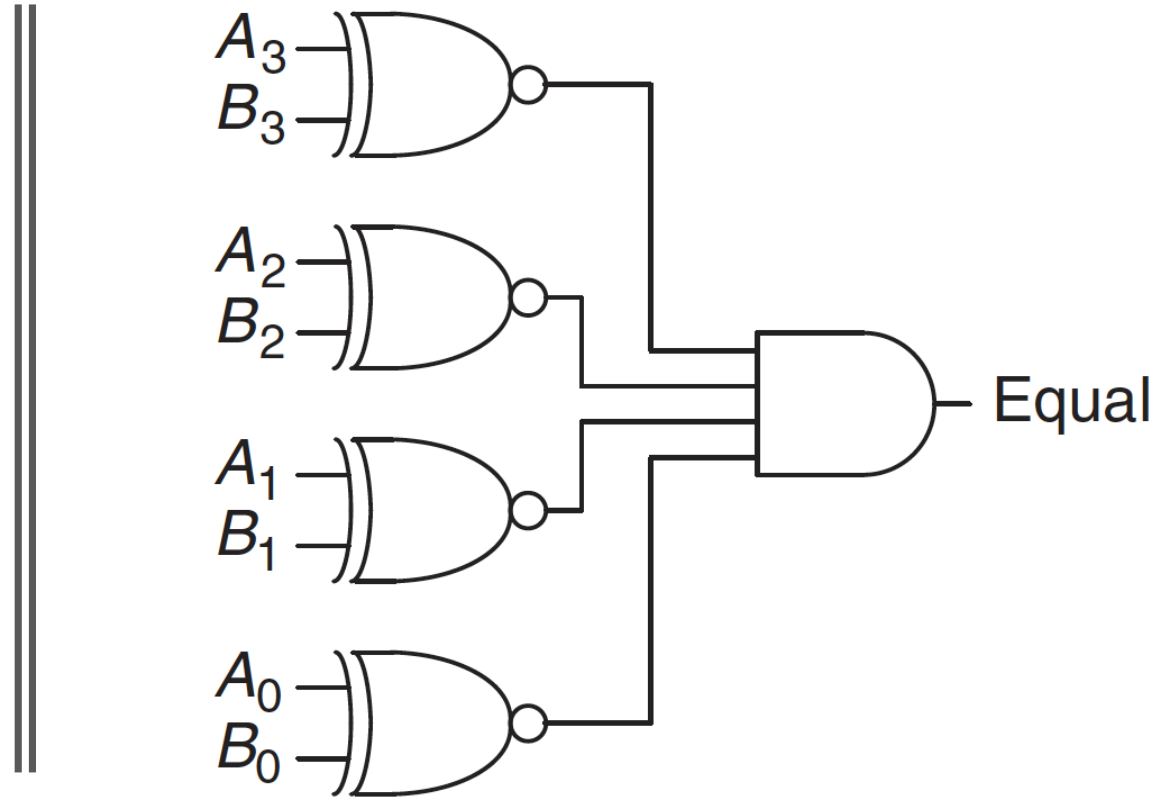


Ripple Carry Adder

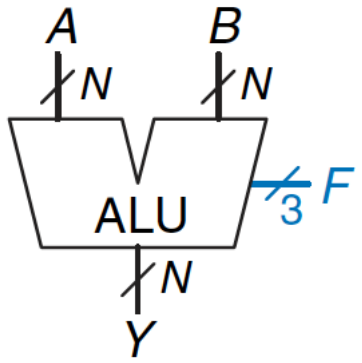


Comparator (Equality Checker)





ALU (Arithmetic Logic Unit)



ALU symbol

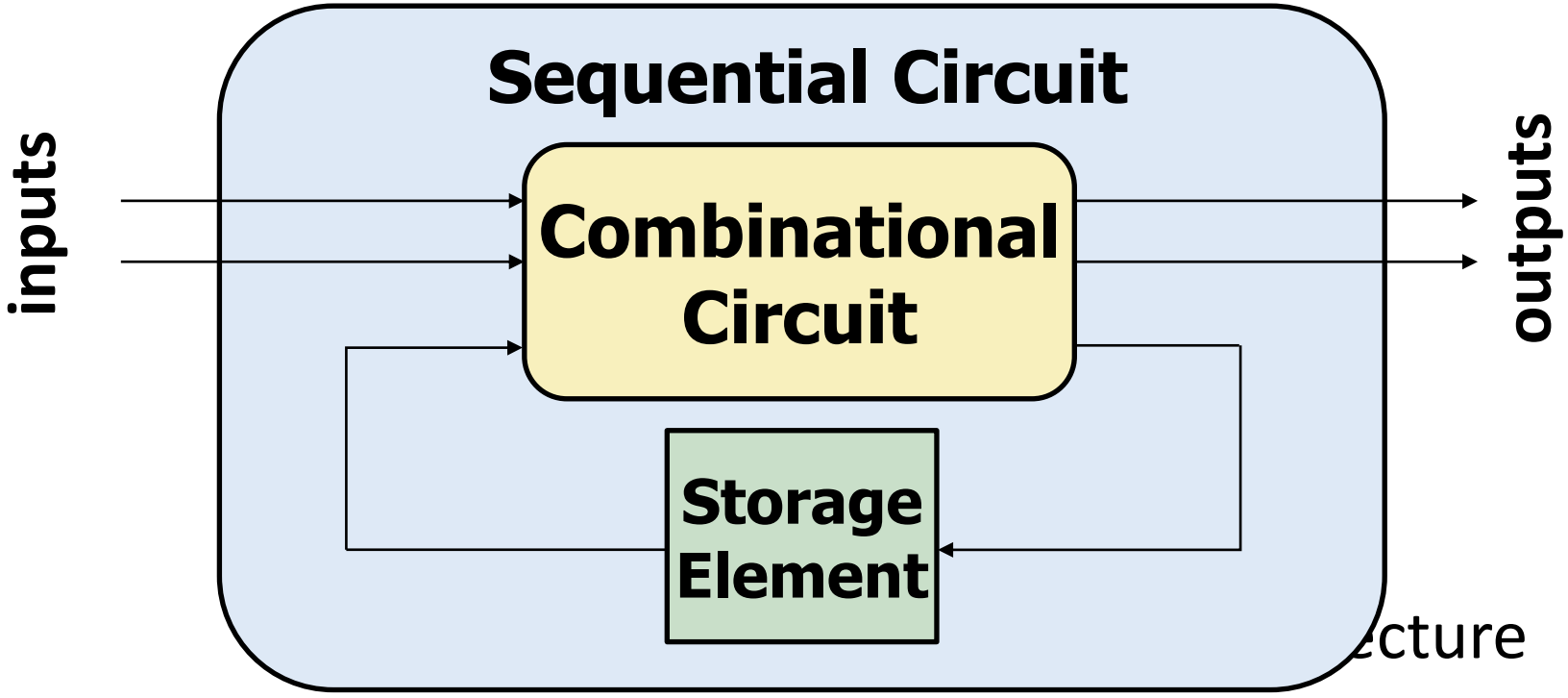
ALU operations

$F_{2:0}$	Function
000	A AND B
001	A OR B
010	A + B
011	not used
100	A AND $\overline{B}$
101	A OR $\overline{B}$
110	A – B
111	SLT


PAUSE

Sequential Circuit

- Combinational circuit output depends **only** on current input
- We want circuits that produce output depending on **current** and **past** input values – circuits with **memory**
- How can we design a circuit that **stores information**?



Textbook Reading

Chapter 2.8 H&H

Coffee points:

- Atharva 210070014


আপনার দিনটি শুভ হোক