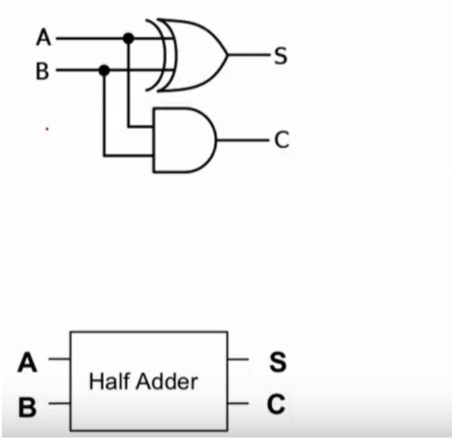
Combinational Building Blocks

Joannah Nanjekye

July 16, 2024

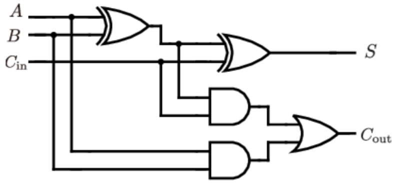
A Half Adder

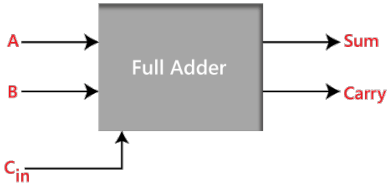


- Add two bits A and B
- Output the sum S
- Output the carry C

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

A Full Adder

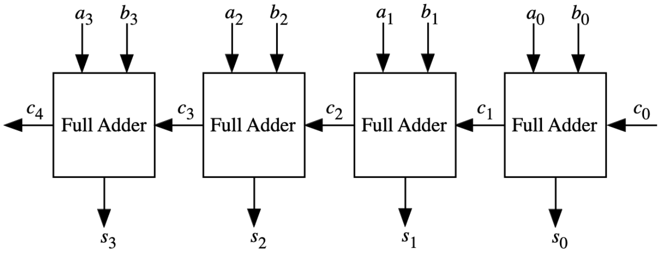




- Add three bits A, B, and C_{in}
- Output the sum S
- Output the carry C_{out}

A	B	C_{in}	C_{out}	S
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

A 4-bit Adder



- Constructed with full adders connected in cascaded
- Carry output from each adder is connected to the carry input of the next adder in the chain
- Carry input is from the least significant bit (LSB)
- The sum output is represented by the bits s_0-s_3

Ripple Carry Adder Delays

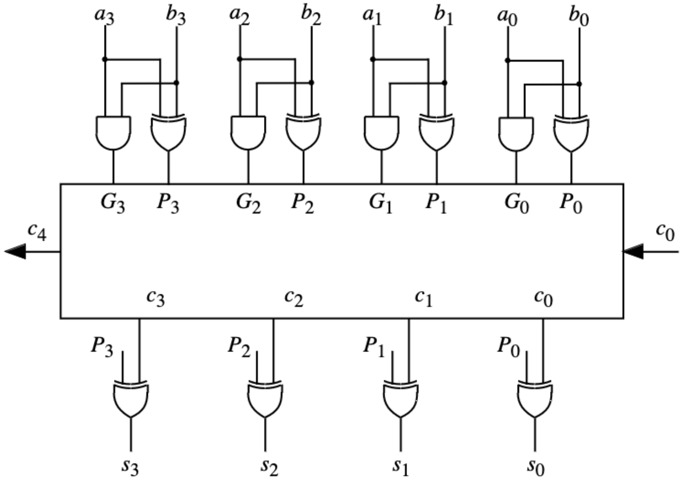
- An output sum and carry is determined after the carry created from the previous stages is produced
- The sum of the MSB is known after the carry signal has rippled through the adders from the LSB
- Final sum and carry bits will be valid after a considerable delay
- For an *n*- bit the delay is:

$$Sum_{s_{n-1}} delay = 4n + 2$$

$$Carry_{c_n} delay = 4n + 3$$

Carry Lookahead Adder (CLA)

- Calculate the carry signals in advance, based on the input signals



Carry lookahead adder (CLA)

- Assumes a carry signal can be generated in two cases:
  - When both bits are 1
  - When one of the two bits is 1 and the carry-in is 1
- We can therefore write the equations:

$$C_{i+1} = G_i + P_i \cdot C_i$$

$$S_i = P_i \oplus C_i$$

- Where:

$$G_i = a_i \cdot b_i$$

$$P_i = a_i \oplus b_i$$

Advantages and Disadvantages of Carry Look-Ahead Adder

Advantages:

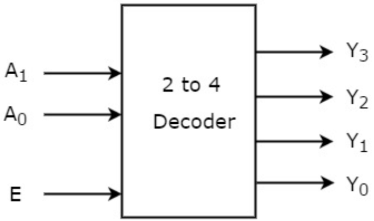
- The propagation delay is reduced
- It provides the fastest addition logic

Disadvantages:

- The Carry Look-ahead adder circuit gets complicated as the number of variables increase
- The circuit is costlier as it involves more number of hardware

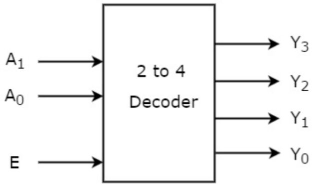
Decoders

- A decoder selects one of 2^n outputs according to an encoded input (n bits)
- Only one of the outputs is asserted based on the combination of inputs present
- When the decoder is enabled
- When an "enable" input, enable = 0 then all outputs are 0



2 to 4 Decoder

The outputs of the decoder are the min terms of *n* input variables lines, when it is enabled



E	A	B	Y_0	Y_1	Y_2	Y_3
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

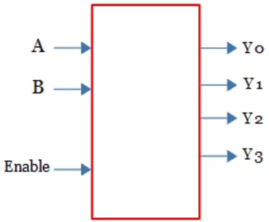
$$Y_3 = E \cdot A \cdot B$$

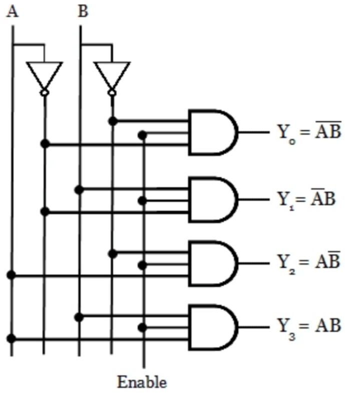
$$Y_2 = E \cdot A \cdot \overline{B}$$

$$Y_1 = E \cdot \overline{A} \cdot B$$

$$Y_0 = E \cdot \overline{A} \cdot \overline{B}$$

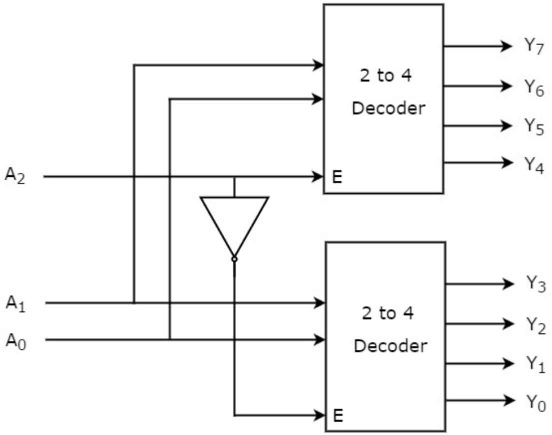
2 to 4 Decoder





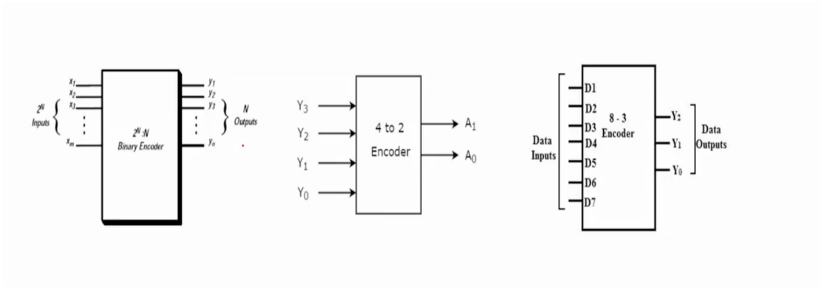
Constructing Higher Order Decoders

# of decoders = m_2/m_1
# of 2 to 4 decoders = 8/2 for a 3 to 8 decoder



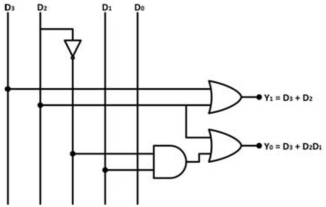
Encoders

- A encoder is the inverse of a decoder
- It is not used frequently as decoders



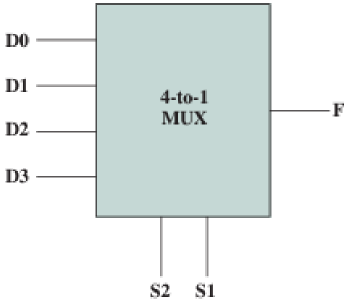
4 to 2 Encoder

HIGHEST	*INPUTS*		*LOWEST*	*OUTPUT*	
D3	**D2**	**D1**	**D0**	**Y0**	**Y1**
0	0	0	0	X	X
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

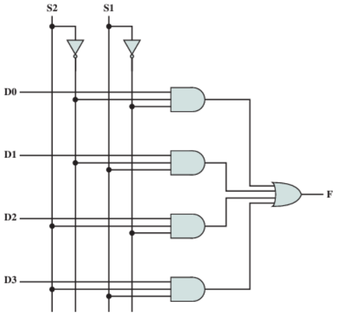


Multiplexer

- A multiplexer or mux is a device that selects one of several input signals and forwards the selected input into a single line
- A multiplexer of 2^n inputs has n select lines

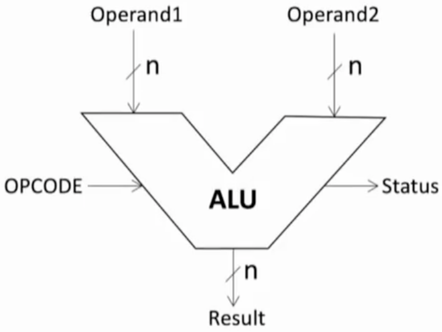


Multiplexer

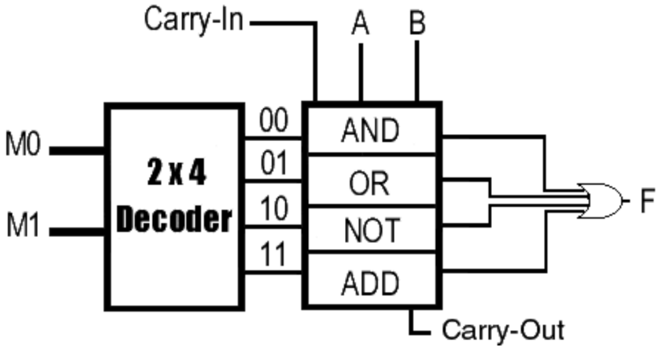


S2	S1	F
0	0	D0
0	1	D1
1	0	D2
1	1	D3

Arithmetic and Logic Unit (ALU)



Constructing an ALU



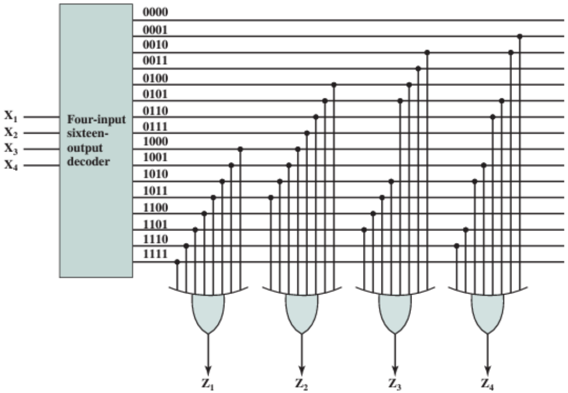
Memory in Combinational Circuits

- These circuits can also have memory
- The kind is called read-only memory (ROM)
- ROM is permanent and created during fabrication
- Outputs are a function of the present inputs
- Can be implemented with a decoder and a set of OR gates

ROM

Input				Output			
X_1	X_2	X_3	X_4	Z_1	Z_2	Z_3	Z_4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

ROM



X1
X2
X3
X4
Four-input sixteen-output decoder
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
Z1
Z2
Z3
Z4