CX1005
Digital Logic

Combinational Circuits

So far, you have covered…

- Number Systems and Digital Arithmetic
- Basic logic gates and Boolean Algebra
- Truth Tables and K-Map
- Gate level combinational circuits

Adding two signed binary numbers

	+6		0110
+	-3	⇨ +	1101
	+3		0011

X	Y	C_i	C_o	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

Truth table for Full Adder

Minterm (or Maxterm) expression

$$C_o = \sum (3, 5, 6, 7)$$

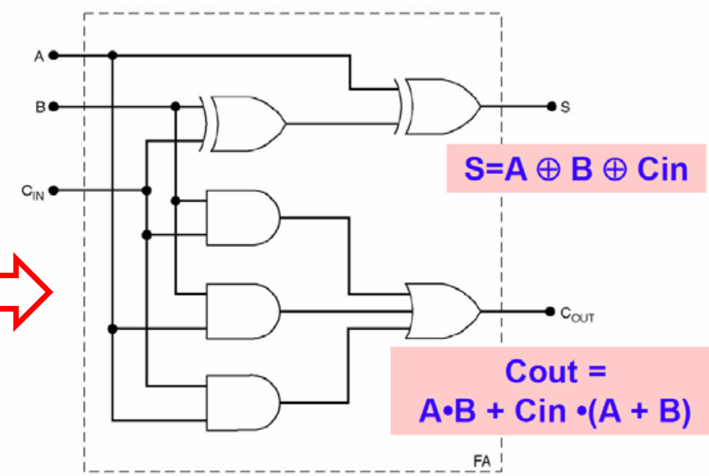
$$S = \sum (1, 2, 4, 7)$$

$$S = X \oplus Y \oplus C_i$$

$$C_o = X.Y + X.C_i + Y.C_i$$

Minimize using Boolean Algebra or K-Maps

Implementation of Full Adder using basic logic gates



Where is this going?

- Combinational circuits are used extensively within digital systems that are found in many electronic devices

Have you tried to count the number of computers in your house?

Things like microwave ovens, dishwashers, refrigerators, televisions may be obvious. But what about remote controllers for the air conditioner, the stereo, the Blu-Ray player and so on ....

***Source**: http://www.mydentistrocks.org/wp-content/uploads/2014/06/20140324200975887588.jpg*

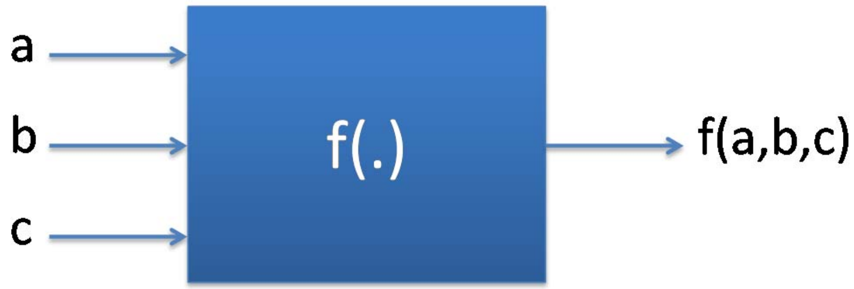
In Today's Lecture...

- Outline
  - Combinational design process
  - Commonly used combinational circuits
    - 7-Segment Decoder
    - Decoder (One-Hot)
    - Multiplexer

- Outcomes
  - Learn how to design simple combinational circuits
  - Understand the working principles of commonly-used combinational circuits and appreciate how they can be used to build digital systems

Combinational Circuits

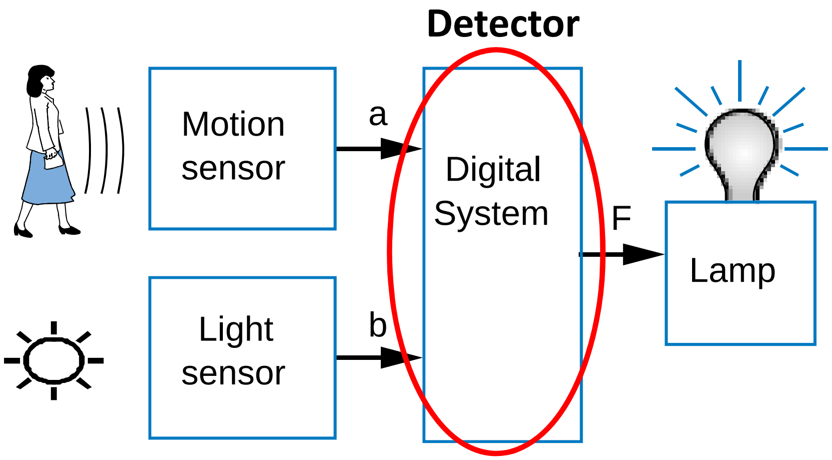
- Combinational circuits are functions:
  - *a function is a relation between a set of inputs and a set of permissible outputs with the property that each input is related to exactly one output*



- Remember, a combinational circuit takes a set of inputs, and for each input combination, always produces a corresponding output
- If I apply the same input values, I always get the same output values:
  - In any order
  - At any time

Example of Combinational Circuits

- Combinational circuits:
  - Outputs depend solely on the **present combination** of the input values



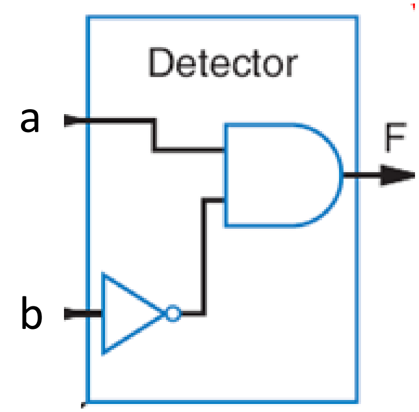
Source: Frank Vahid, "Digital Design", Wiley, 2nd Edition, pg. 35

if a=0 and b=0, then F=0

if a=0 and b=1, then F=0

if a=1 and b=0, then F=1

if a=1 and b=1, then F=0



Source: Frank Vahid, "Digital Design", Wiley, 2nd Edition, pg. 49

- Motion-in-dark example
  - Turn on lamp (F=1) when
    - Motion sensed (a=1) and no light (b=0)
  - F = a **AND NOT**(b)
  - Build using logic gates, **AND** and **NOT**, as shown

Combinational Design Process

- Hence, we can define a two-step process for implementing combinational designs:
  - **Step 1: Capture the function**
    - Use truth-table or equations, depending on what makes sense for the application
  - **Step 2: Convert to circuit**
    - 2A: First, if you used a truth table in Step 1, create equations
    - 2B: For each output, create a circuit corresponding to that output's equation

Example: Three 1s Pattern Detector

- **Problem 1**: Detect three adjacent 1's in an 8-bit input

  abcdefgh

  0001**11**01 → 1

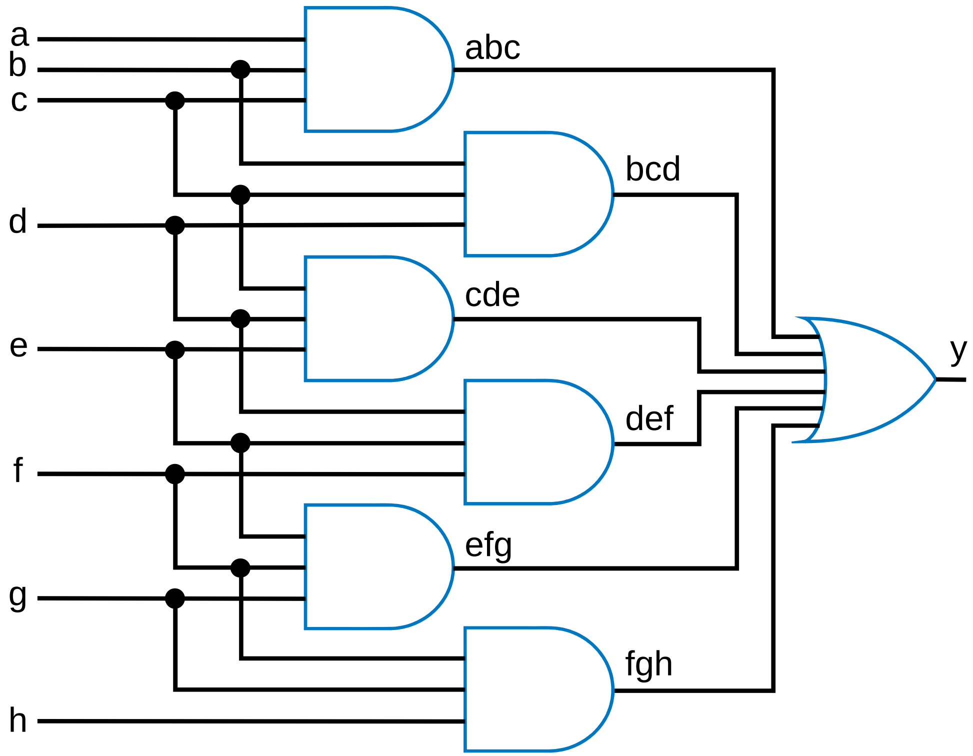
  10110011 → 0

  0**111**0001 → 1

- **Step 1**: Capture the function
  - Truth table would be too big (2^8 = 256 rows!)
  - Use an equation, with a term for each possible case:
  - y = abc + bcd + cde + def + efg + fgh
- **Step 2A**: Create equation (Already Done)
- **Step 2B**: Implement using gates

Example: Three 1s Pattern Detector

- y = abc + bcd + cde + def + efg + fgh

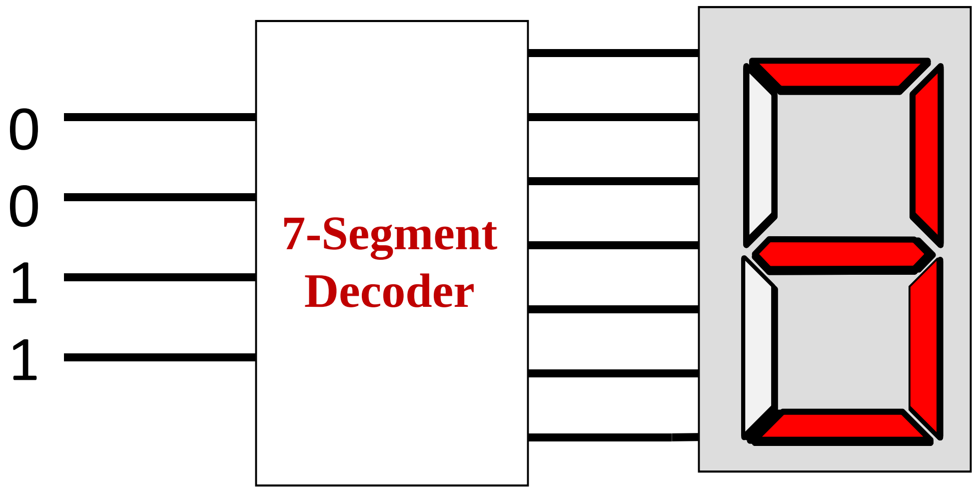


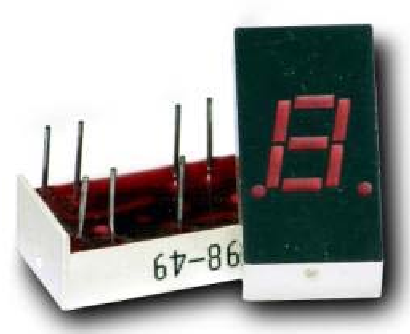
***Source**: Frank Vahid, "Digital Design", Wiley, 2nd Edition, pg. 75*

SEVEN SEGMENT DECODER

Seven Segment Decoder

- Consider a seven segment display
- We want a circuit that maps binary numbers to the correct digit on the display

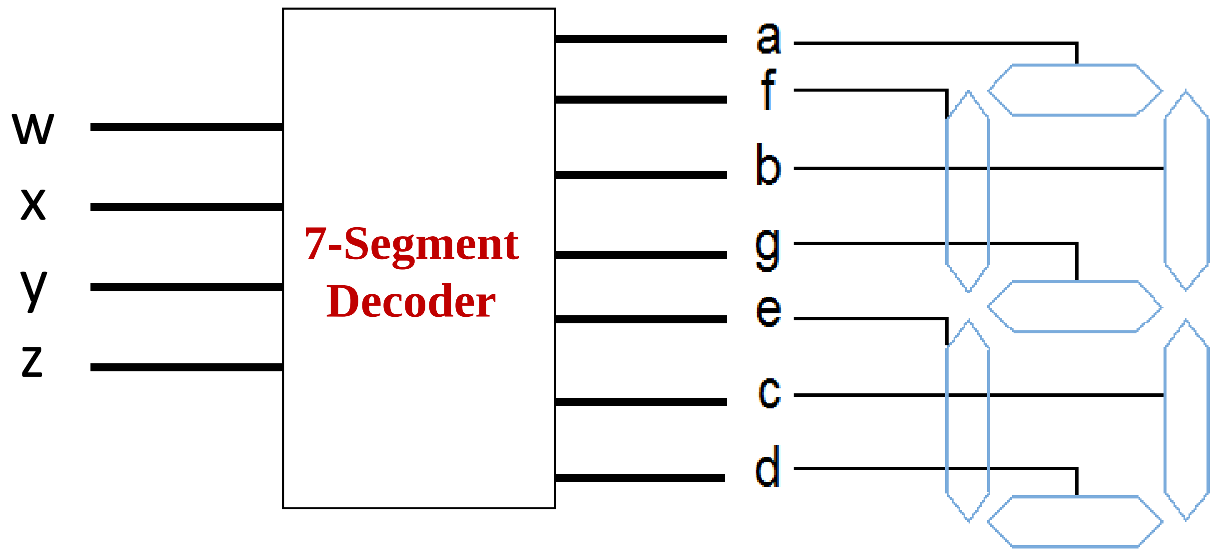


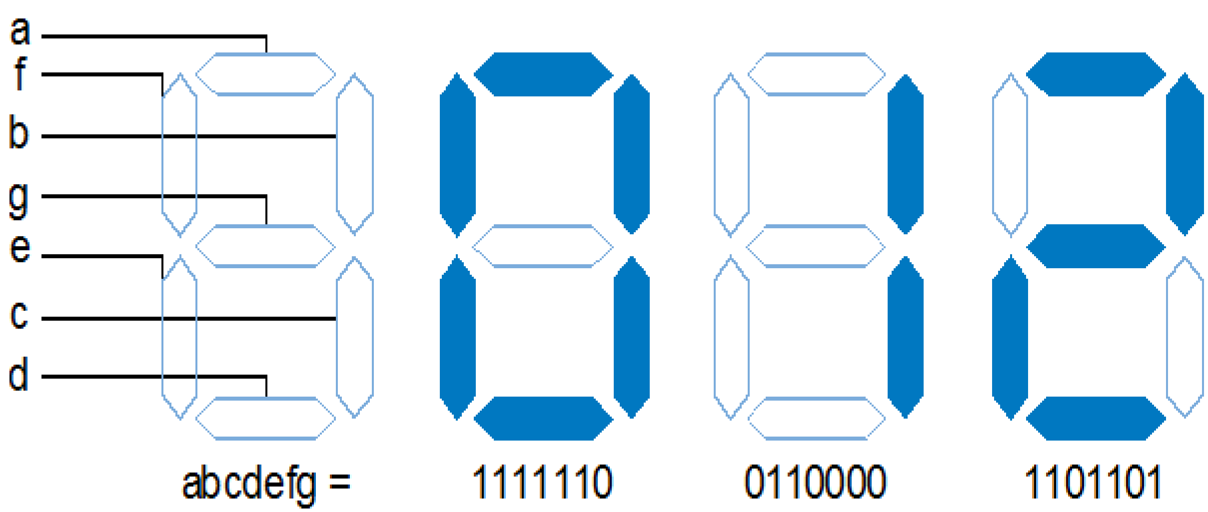


***Source**: Frank Vahid, "Digital Design", Wiley, 2nd Edition, pg. 72*

- The 7-Segment Decoder needs to decide, for each input set, which segments should be lit to display the correct digit

Seven Segment Decoder

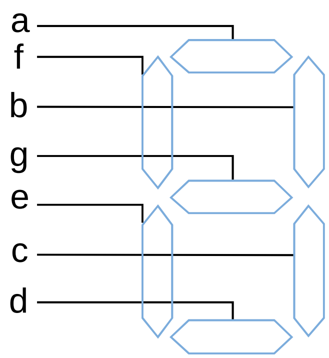


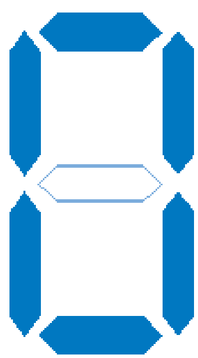


***Source**: Frank Vahid, "Digital Design", Wiley, 2nd Edition, pg. 72*

Seven Segment Decoder

- We can determine a function for each segment, that indicates whether it should be lit for a specific input pattern





1111110

Inputs X[3:0]	Outputs Seg[0:6]						
Hexadecimal digits (binary)	**Segments (1: on, 0: off)**						
	a	b	c	d	e	f	g
0 (0000)	1	1	1	1	1	1	0
1 (0001)	0	1	1	0	0	0	0
2 (0010)	1						
3 (0011)	1						
4 (0100)	0						
5 (0101)	1						
6 (0110)	1						
7 (0111)	1	1	1	0	0	0	0
8 (1000)	1	1	1	1	1	1	1
9 (1001)	1	1	1	1	0	1	1
A (1010)	1						
B (1011)	0	0	1	1	1	1	1
C (1100)	1	0	0	1	1	1	0
D (1101)	0						
E (1110)	1						
F (1111)	1						

- Step 1: We can construct a truth table

Seven Segment Decoder

Inputs X[3:0]	Outputs Seg[0:6]						
Hexadecimal digits (binary)	Segments (1: on, 0: off)						
	a	b	c	d	e	f	g
0 (0000)	1	1	1	1	1	1	0
1 (0001)	0	1	1	0	0	0	0
2 (0010)	1						
3 (0011)	1						
4 (0100)	0						
5 (0101)	1						
6 (0110)	1						
7 (0111)	1	1	1	0	0	0	0
8 (1000)	1	1	1	1	1	1	1
9 (1001)	1	1	1	1	0	1	1
A (1010)	1						
B (1011)	0	0	1	1	1	1	1
C (1100)	1	0	0	1	1	1	0
D (1101)	0						
E (1110)	1						
F (1111)	1						

- Step 2A: Create equations for each output from truth table
- Step 2B: Create a circuit corresponding to that output's equation

Kmap for a		X1,X0			
		00	01	11	10
X3,X2	00	1	0	1	1
	01	0	1	1	1
	11	1	0	1	1
	10	1	1	0	1

a = X3'X2X0 + X2X1 + X3'X1 + X3X0' + X3X2'X1' + X2'X0'

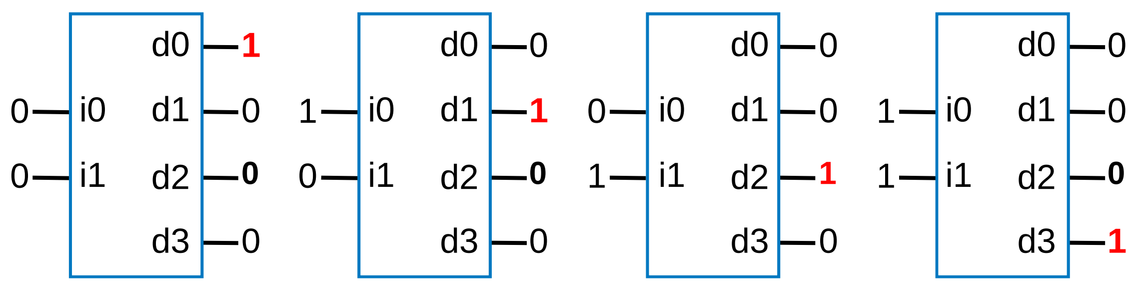
DECODER

Decoder

- Decoders are an important basic circuit, similar to the 7-Segment Decoder
- Take a binary input number, and output a corresponding one-hot output
  - Only one bit of the output is high, its position corresponds to the input value
  - Output width is always 2 to the power of input width
    - N-input Decoder: 2^N outputs

***Source**: Frank Vahid, "Digital Design", Wiley, 2nd Edition, pg. 84*

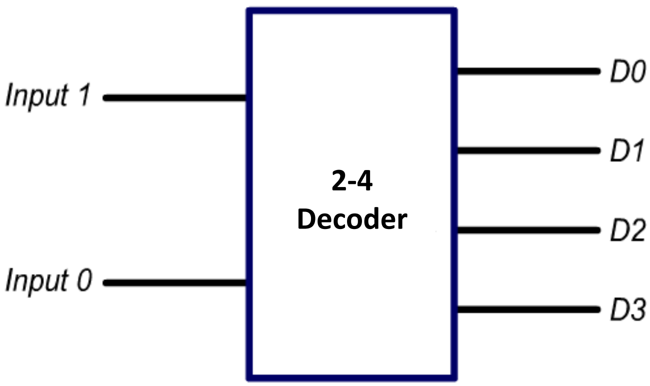
Input		Output
00	→	0001
01	→	0010
10	→	0100
11	→	1000

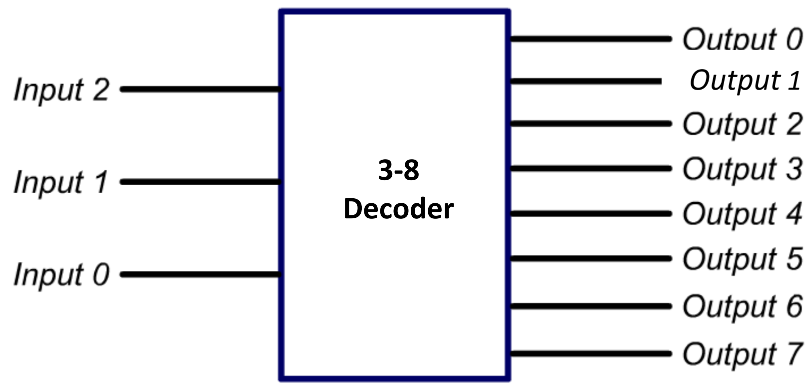


Example: A two-input decoder will have four outputs (2-4 Decoder)

Decoder

- In general, decoders can be referred to n-m decoders





Input		Output			
1	0	D0	D1	D2	D3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

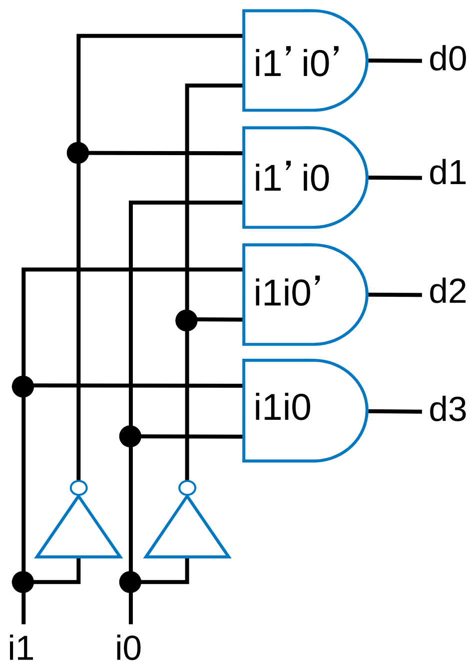
Input			Output							
2	1	0	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Source: *https://filebox.ece.vt.edu/~jgtront/introcomp/decoder.swf*

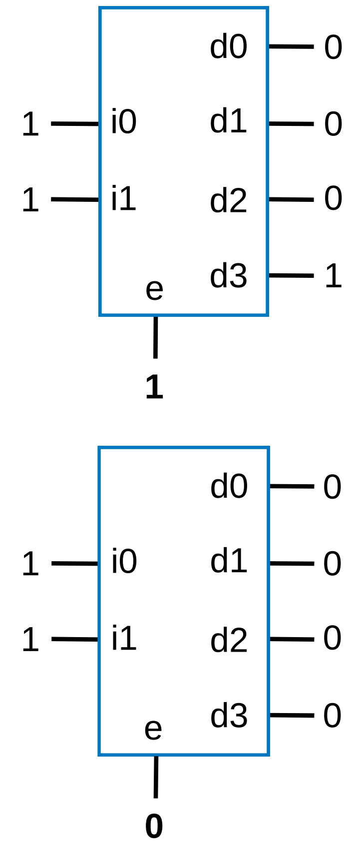
Decoder

- Internal design
  - **AND** gate for each output to detect input combination
- Decoder with enable e
  - e = 0: Outputs all 0
  - e = 1: Regular behavior

Input		Output
00	→	0001
01	→	0010
10	→	0100
11	→	1000



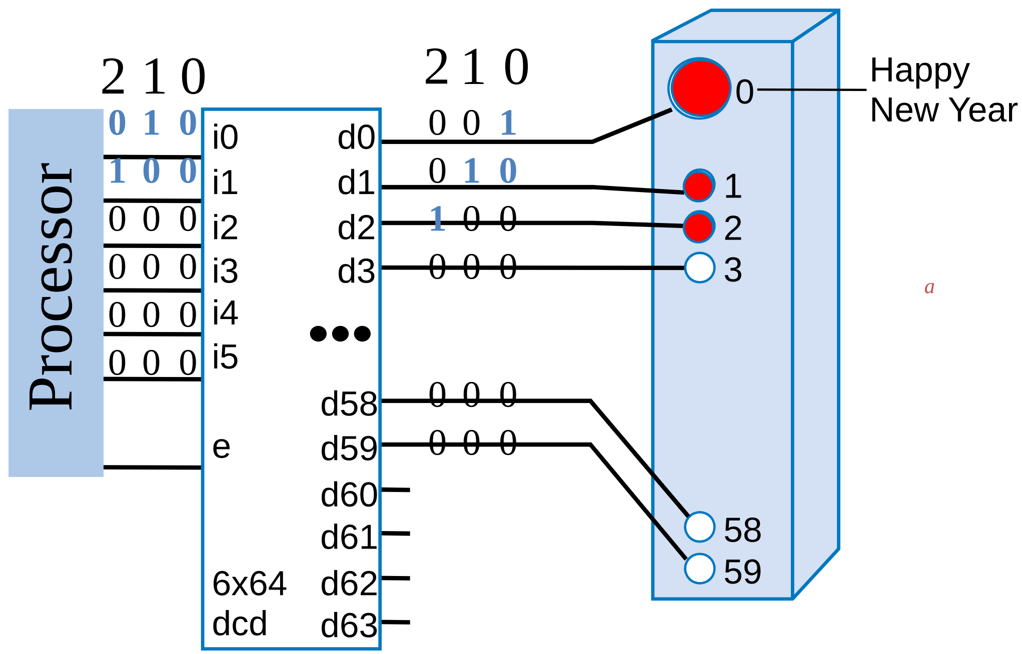
Source: Frank Vahid, "Digital Design", Wiley, 2nd Edition, pg. 84



Source: Frank Vahid, "Digital Design", Wiley, 2nd Edition, pg. 85

Example: New Year's Eve Countdown Display

- Processor counts from 59 to 0 in binary
- Need to convert the 6-bit count to light up one bulb on the display
- 6-64 Decoder is used (4 outputs unused)

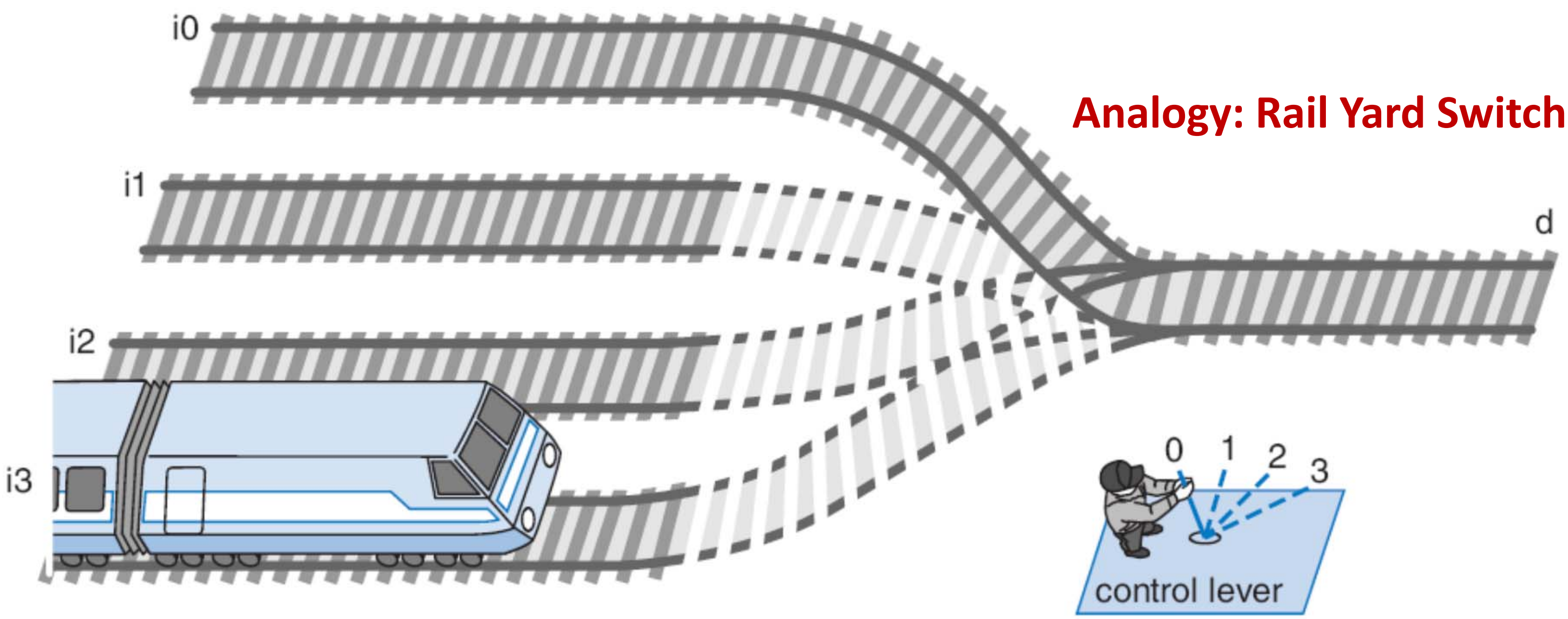


Source: Frank Vahid, "Digital Design", Wiley, 2nd Edition, pg. 86

MULTIPLEXER

Multiplexer (Mux)

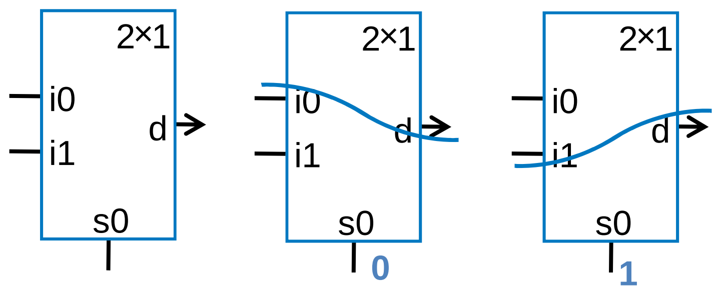
- Another important combinational circuit is the multiplexer (selects one from several inputs)
- Consists of multiple inputs, and a single output
- A select input determines which input should be connected to the output



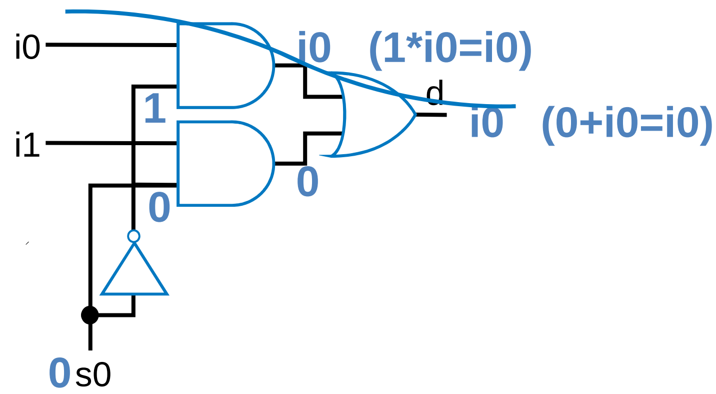
Source: *Frank Vahid, "Digital Design", Wiley, 2nd Edition, pg. 87*

Multiplexer (Mux)

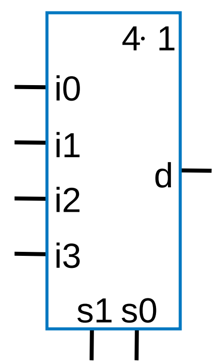
- Internal design of a 2x1 Mux



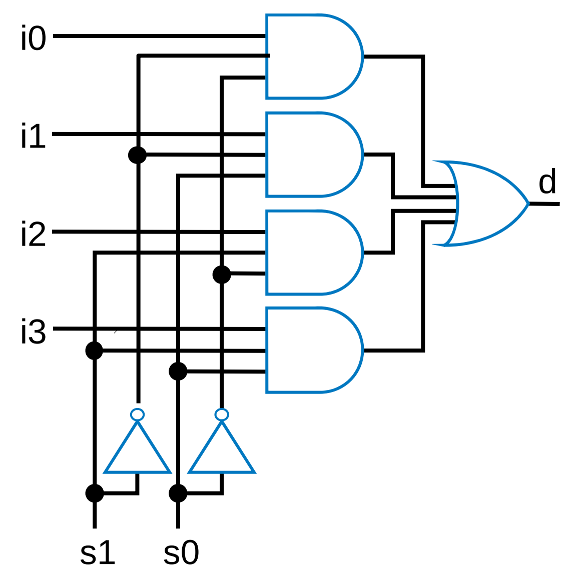
Source: Frank Vahid, "Digital Design", Wiley, 2nd Edition, pg. 87



- 2 input multiplexer (mux) needs a 1-bit select input
- 4 input mux requires a 2-bit select input
- An **n** input mux requires a **$\log_2(n)$**-bit select



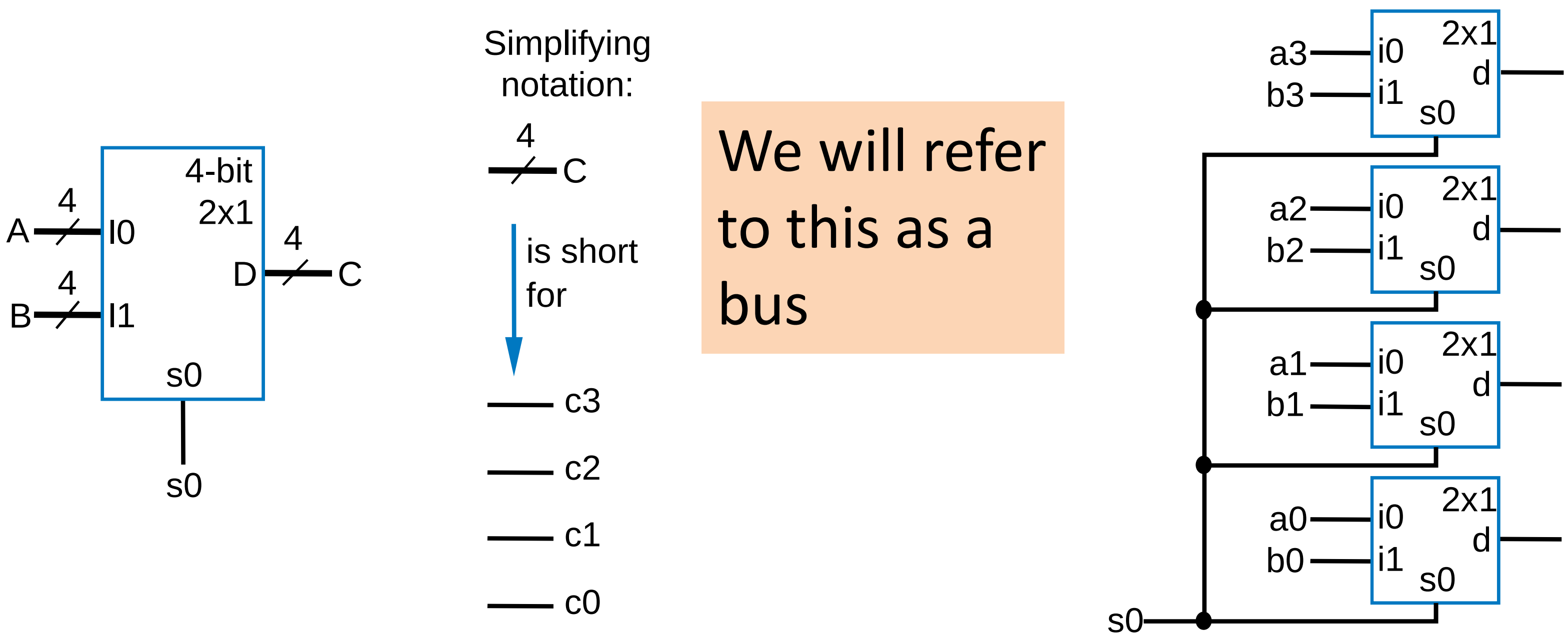
4x1 mux



Source: Frank Vahid, "Digital Design", Wiley, 2nd Edition, pg. 88

Multiplexer (Mux)

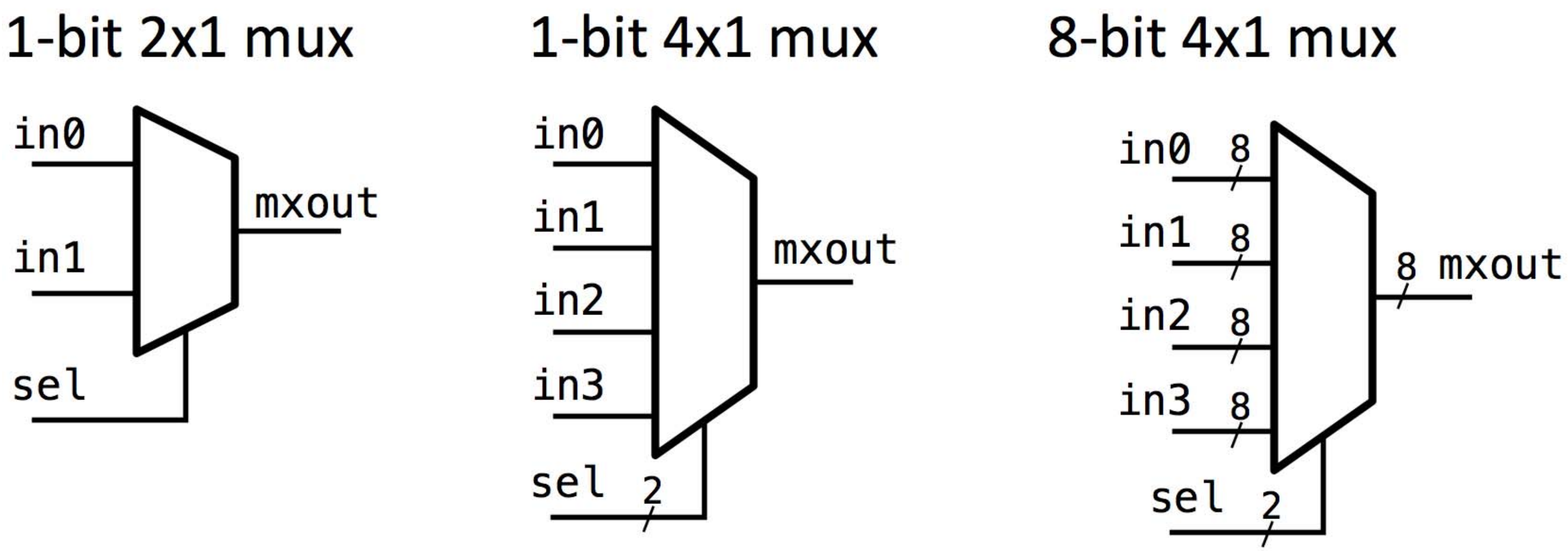
- We can combine Muxes to select multi-bit inputs, e.g. numbers
- A 4-bit 2x1 mux
  - Four 2x1 Muxes sharing the same select line to select between A and B



Source: Frank Vahid, "Digital Design", Wiley, 2nd Edition, pg. 89

Multiplexer (Mux)

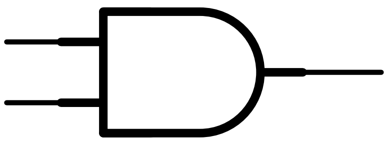
- Muxes are so common, they are often drawn using their own symbol:



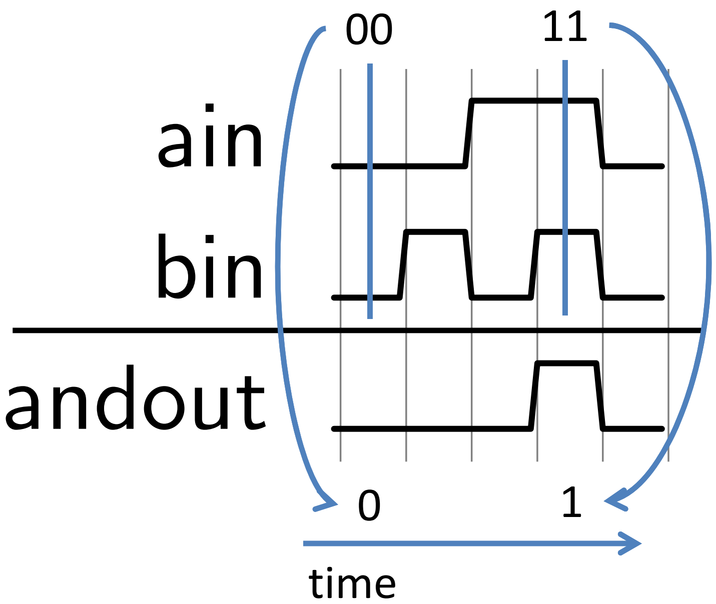
TIMING DIAGRAMS

Timing Diagrams

- Timing diagrams show the behavior of a circuit with progression of time
- Input values are changed and the resultant outputs shown
- Consider an **AND** gate:



- A timing diagram can show any combination or order of input values
- The output at any point is calculated by looking at the input values at that instance



Summary

- Combinational circuit always produces a corresponding output for each input combination
- Two-step process for implementing simple combinational circuits
  - Capture the function
  - Convert to circuit
- Commonly-used combinational circuits e.g. 7-segment decoder, decoders and multiplexers are used to build digital systems