University of Massachusetts Amherst

Engin112 – Lectures 20-22

Decoders, MUXes

Maciej Ciesielski
Department of Electrical and Computer Engineering
10/24-28/2011

Recap from last lectures

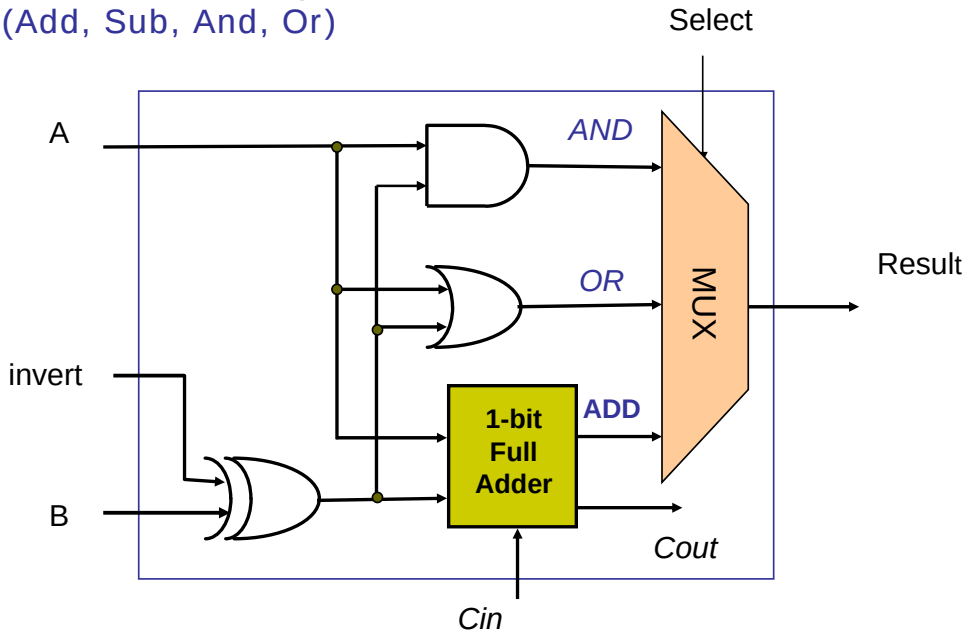
- Standard arithmetic components (combinational)
  - Adders, subtractors
  - Multipliers
  - Comparators

Today's lecture:

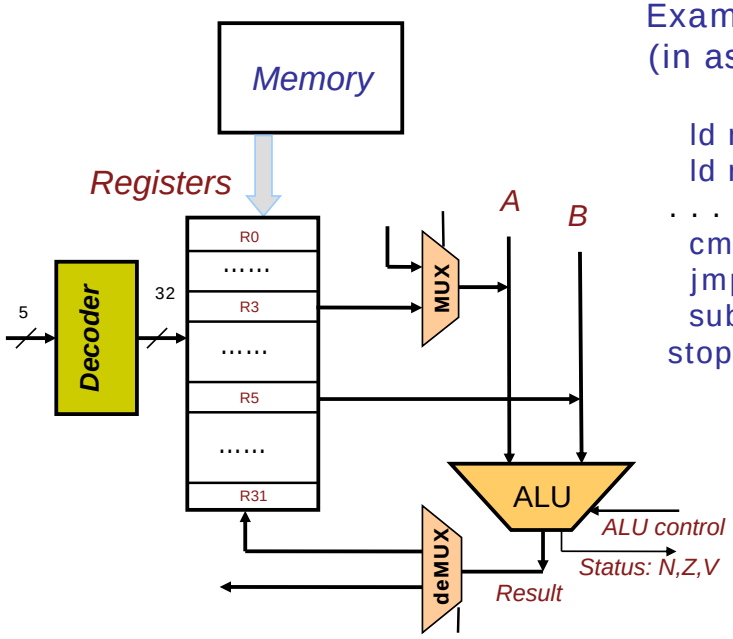
- Other arithmetic blocks
  - Decoders
  - Encoders
  - Multiplexers (MUX)
  - Demultiplexers
  - Tri-state logic

Motivation - ALU Design

ALU = Arithmetic Logic Unit
(Add, Sub, And, Or)



Motivation – CPU Design



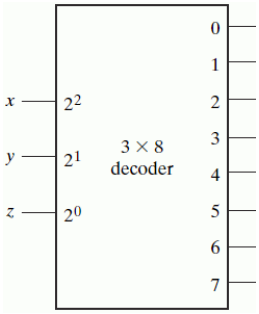
Example of computer program
(in assembly language)

```
  ld r3, 5       (load  r3 ← 5)
  ld r5, 8       (load  r5 ← 8)
. . . .
  cmp r3,r5      (compare r3,r5)
  jmpn  stop     ( r3 < r5 ? )
  sub r5, r3     (r5 ← r5-r3)
stop
```

Decoders

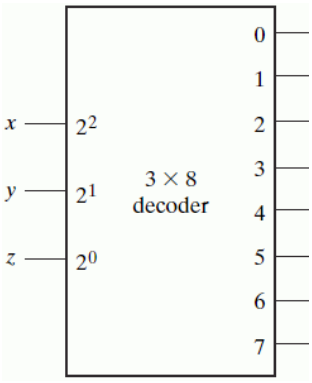
- *Decoder* : selects one output based on binary input
  - Converts *n*-bit code into 2^n outputs, only one being active for any combination of inputs
  - Selects output *x* if input is binary representation of *x*

- Applications
  - Binary-to-octal decoder
  - Memory address selection
  - Selection of any kind
  - Can be used to construct arbitrary logic function



Decoder

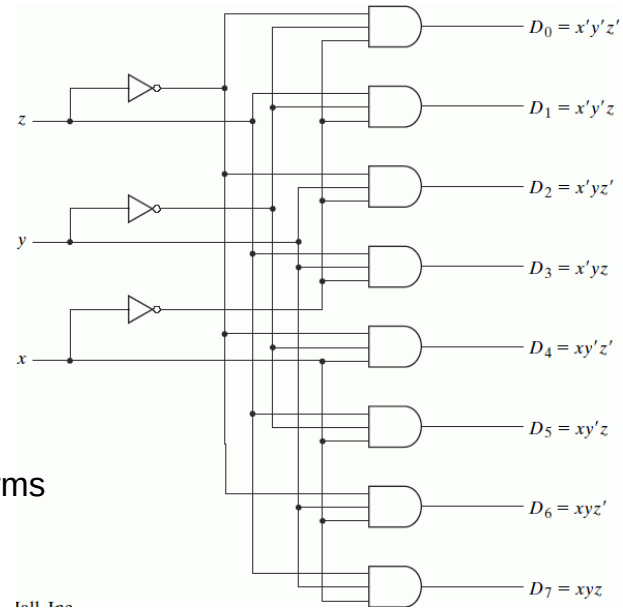
- Example: 3-to-8-line decoder



x	y	z	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**

Decoder Circuit

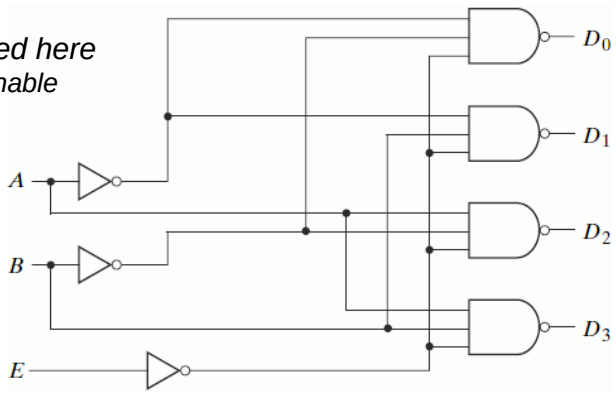
- When is output 0 chosen?
  - If *x' y' z'*
- When is output 1 chosen?
  - If *x' y' z*
- … and so on …
- Circuit for line decoder
  - Sequence of minterms
  - Combine variables to minterms



Advanced Decoder

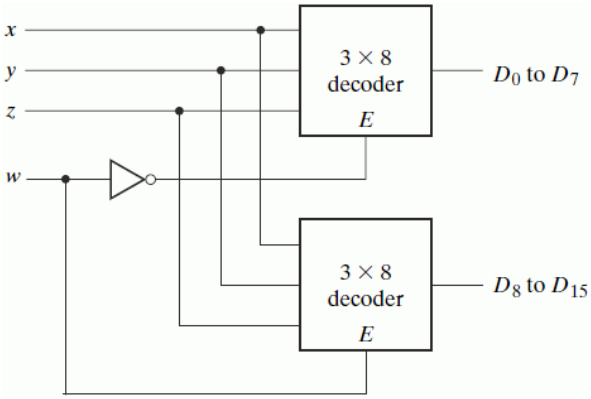
- Additional feature: *Enable* input
  - Circuit generates output only if *Enable* is selected (*E=0*)
    - » *active-low enable*
  - If disabled (*E=1*), no output line is picked
- Example:
  - 2-to-4 line decoder with *Enable*
  - NAND implementation
- Truth table for NAND decoder
  - *Note: NEGATIVE polarity is used here*
    - » Complemented outputs and *Enable*

E	A	B	D0	D1	D2	D3
1	X	X	**1**	**1**	**1**	**1**
0	0	0	**0**	1	1	1
0	0	1	1	**0**	1	1
0	1	0	1	1	**0**	1
0	1	1	1	1	1	**0**



Larger Decoders

- Enable bit can be used for building larger decoders
  (assume active-high enable, i.e., *E=1* activates the encoder)
  - *w = 0* activates upper decoder (bits $D_7 \ldots D_0$)
  - *w = 1* activates lower decoder (bits $D_{15} \ldots D_8$)



- Effect: *w* adds one input bit
  - *n* = 3 → 4
- Can we use this new decoder to get a 5-to-32 line decoder?
  - No!
  - 4-to-16 line decoder does not have *Enable*

Encoders

- *Encoder:* translates 2^n input lines into *n* output lines
  - Input: 2^n lines
  - Output: *n* bits
  - Output is binary coding of input that is 1
- Truth table (*n=3*) (positive polarity):
  - Note: only one input is allowed to be active

D0	D1	D2	D3	D4	D5	D6	D7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	**1**	0	0	0	0	0	0	0	0	1
0	0	**1**	0	0	0	0	0	0	1	0
0	0	0	**1**	0	0	0	0	0	1	1
0	0	0	0	**1**	0	0	0	1	0	0
0	0	0	0	0	**1**	0	0	1	0	1
0	0	0	0	0	0	**1**	0	1	1	0
0	0	0	0	0	0	0	**1**	1	1	1

Encoders

- What is the problem with a 2^n-to-n line encoder?
  - Number of output combinations: 2^n (all used)
  - Number of input combinations: 2^{2^n} (only 2^n used)
  - Only one input can be active (line =1)!
  - Possibly invalid input patterns
    - » More than one input active at any time
    - » No line is active
  - Ambiguity
- How can we solve this problem?
  - What should the output be if multiple lines are active ?
  - Different solutions:
    - » Any one (random)
    - » Give priority to lower or higher lines
    - » Indicate invalid input (requires extra bit, *valid bit* ***V***)

Priority Encoder

- Priority encoder
  - Like encoder, with additional functionality:
    if multiple inputs are 1, give priority to one of the bits
- Example: 4-to-1 priority encoder with priority given to one bit

Valid bit

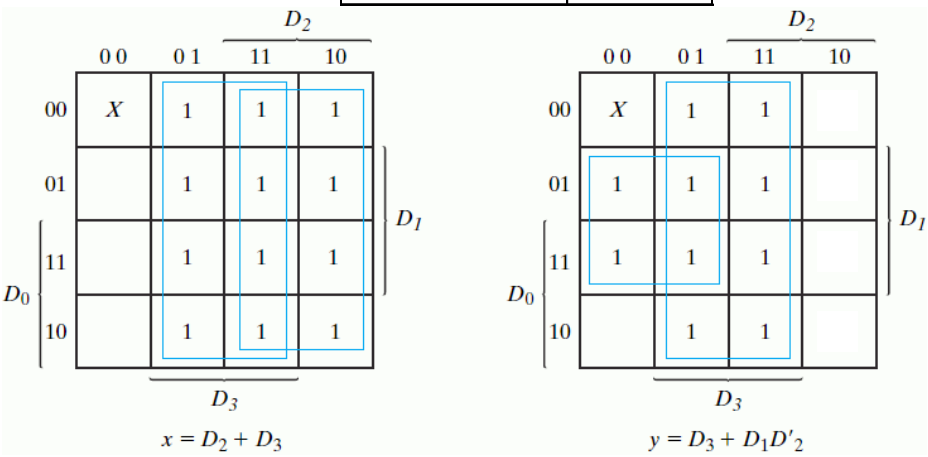
D0	D1	D2	D3	x	y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	**1**	0	0	0	1	1
X	X	**1**	0	1	0	1
X	X	X	**1**	1	1	1

- Which bit has highest priority?
  - D_3

Priority Encoder Circuit

- Karnaugh maps

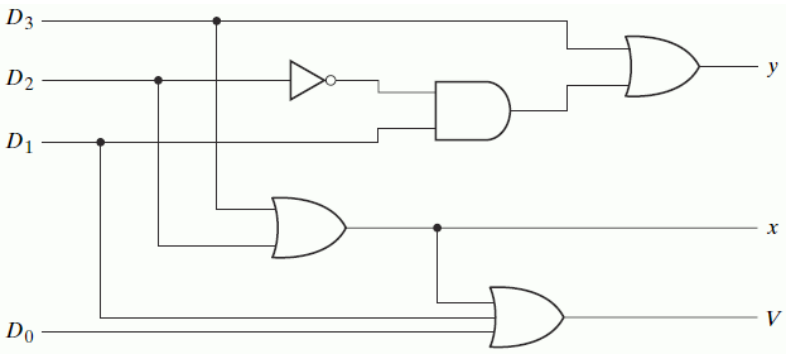
D0	D1	D2	D3	x	y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1



- Valid bit $V = D_0 + D_1 + D_2 + D_3$

Priority Encoder

- Circuit diagram:



$x = D_2 + D_3$

$y = D_3 + D_1 D_2'$

$V = D_0 + D_1 + D_2 + D_3$

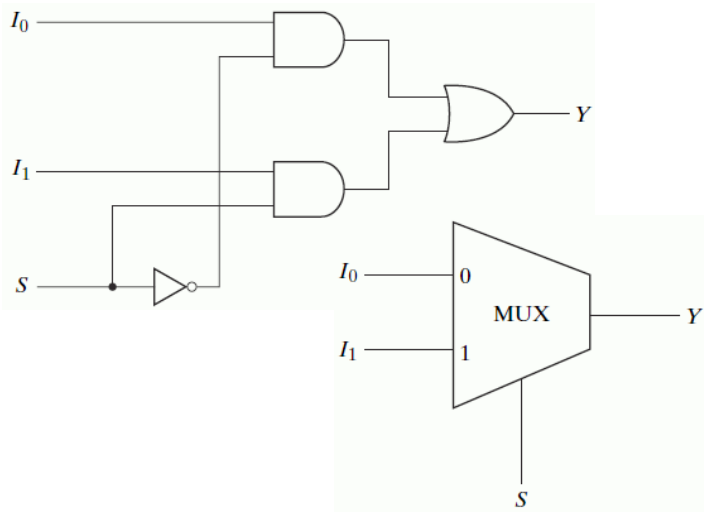
Multiplexer (MUX)

- *Multiplexer* selects binary information from one of many input lines and directs it into a single output line.

- Inputs:
  - 2^n input lines
  - n selection bits

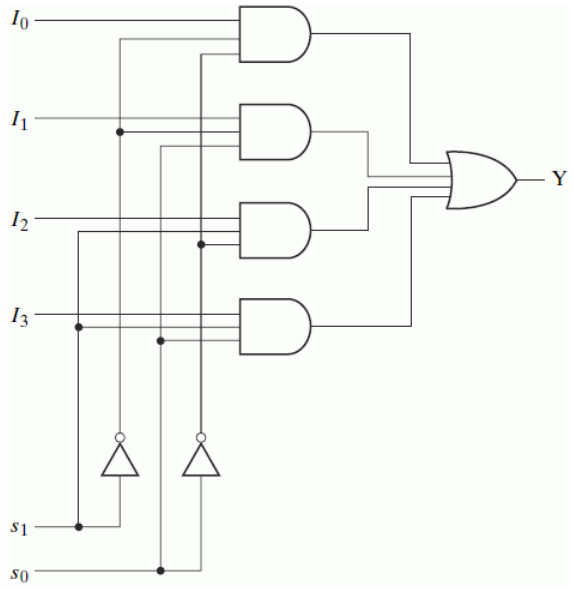
- Output:
  - 1 output line

- Example: 2-to-1 MUX
  - Block symbol



Multiplexer

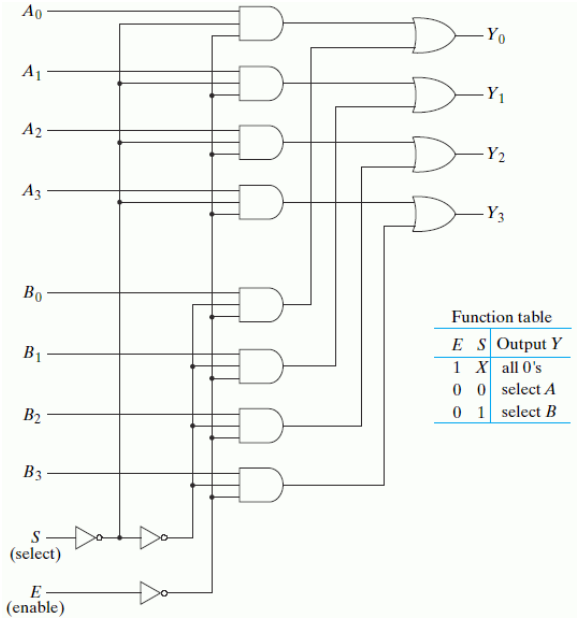
- 4-to-1 multiplexer:



s_1	s_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Multiplexer

- What if we want to select more than one bit?
  - Example: choose one of two 4-bit numbers
- "Quadruple 2-to-1 line multiplexer"
- Select chooses input
- Enable bit sets output to 0 if 1

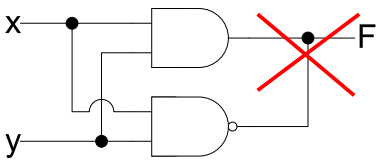


Function table

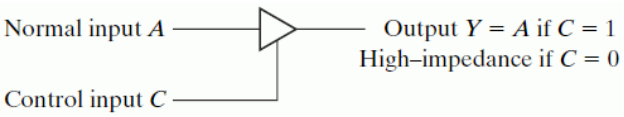
E	S	Output Y
1	X	all 0's
0	0	select A
0	1	select B

Three-State (Tri-state) Gates

- What is the truth table for *F* ?



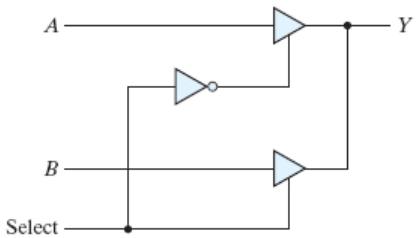
- The two gates will try to drive *F* at the same time
  - Not a good idea to wire their outputs
- Sometimes it is necessary to "disconnect" a gate
- Three state:
  - 0 or 1 Boolean value
  - "High impedance", *Z* state



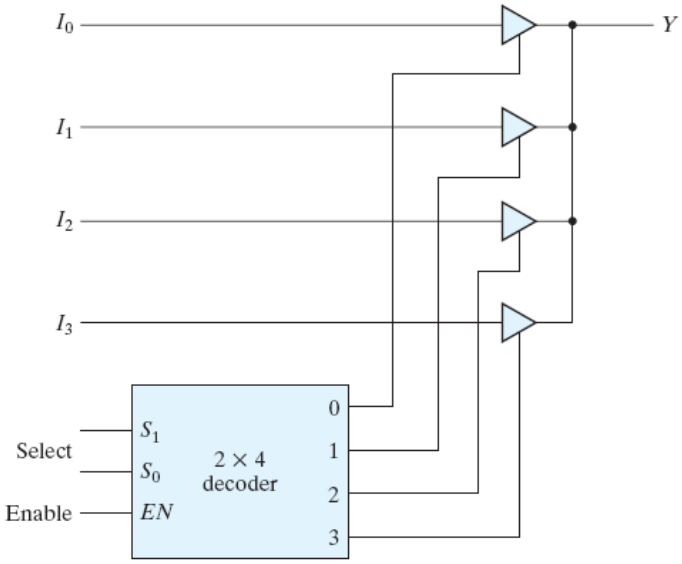
- High impedance acts as if gate were disconnected

Multiplexer with Tri-state Gates

- 2-to-1 MUX:



- 4-to-1 MUX with Enable:



Demultiplexer (deMUX)

- *Demultiplexer* is circuit that receives information from a single line and directs it to one of 2^n possible output lines.

- Inputs:
  - Data bit
  - *n* selection lines

- Outputs:
  - 2^n lines

D	S1	S2	D0	D1	D2	D3
1	X	X	1	1	1	1
0	0	0	**0**	1	1	1
0	0	1	1	**0**	1	1
0	1	0	1	1	**0**	1
0	1	1	1	1	1	**0**

- Can be achieved by decoder circuit with enable *D* active low
  - Enable *D* acts as data bit (“disable”)
  - *n* code lines act as selection

Reading Assignment

- Mano 4.11 (modular design)

- Coming soon:
  - Mano 5.1 - 5.3 (sequential circuits and Latches).