MA251 Computer Organization and Architecture [3-0-0-6]

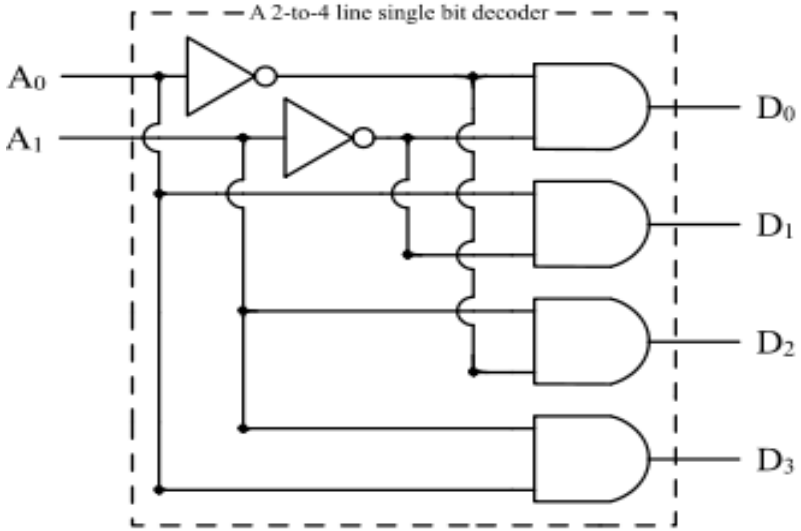
Lecture 5: Decoder, Demultiplexer, Encoder, and Multiplexer..

Spring 2011

Partha Sarathi Mandal

Decoders

- *n* inputs, 2^n outputs
- Each output represents a minterm of an *n* variable function.
- The output that corresponds to the minterm that appears on the inputs is asserted (low or high depending on the particular decoder), all other outputs are deasserted.



Truth Table

A_1	A_0	D_3	D_2	D_1	D_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

Minterm Equations

$D_0 = \overline{A_1} \cdot \overline{A_0}$

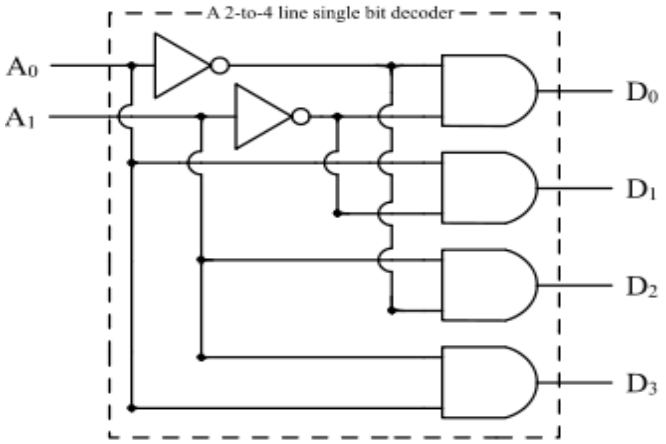
$D_1 = \overline{A_1} \cdot A_0$

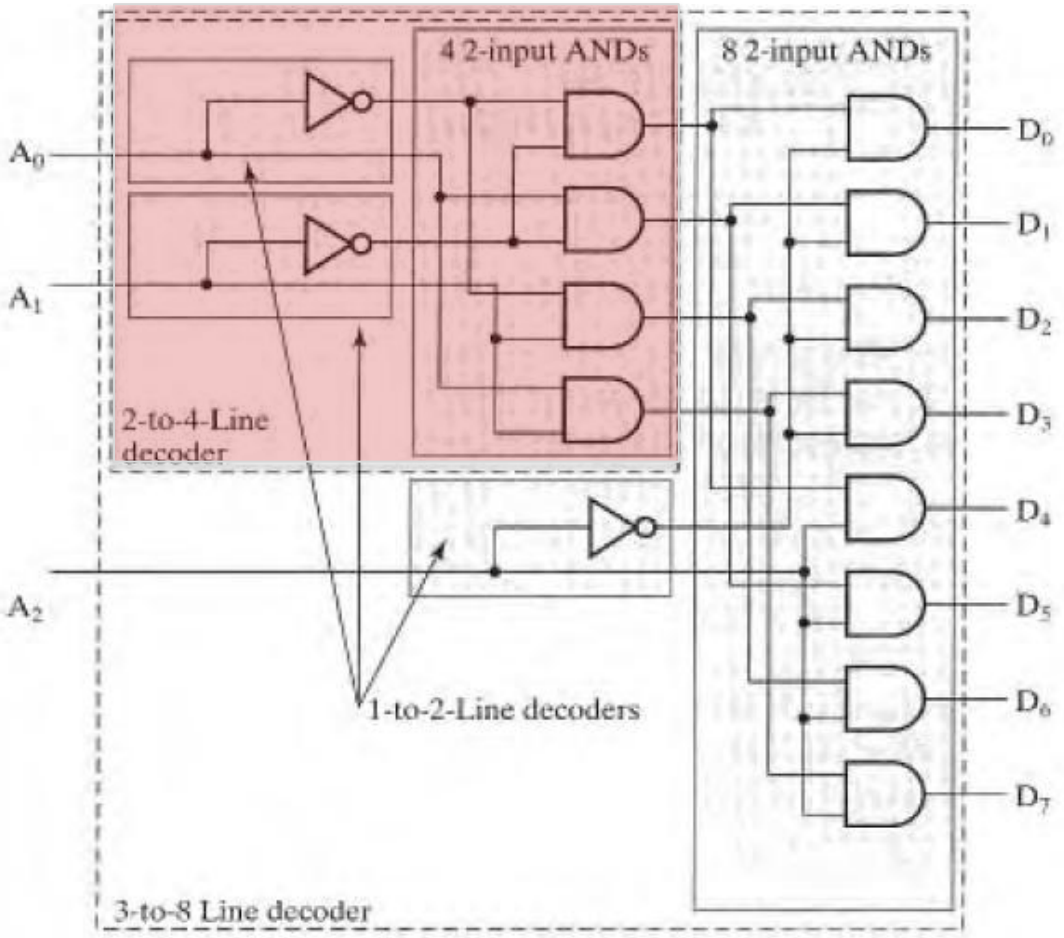
$D_2 = A_1 \cdot \overline{A_0}$

$D_3 = A_1 \cdot A_0$

3-to-8 line decoder

- How to extent a 2-to-4 line decoder to 3-to-8 line decoder ?

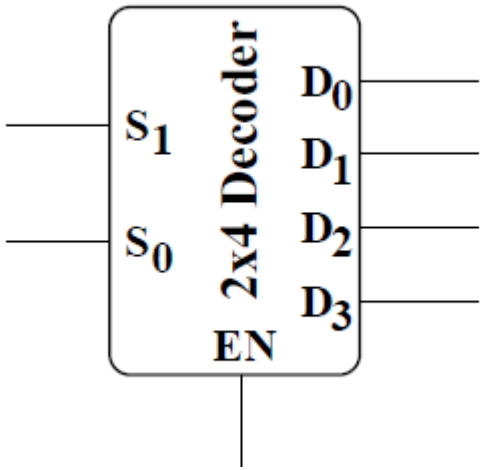




Decoders with an enable

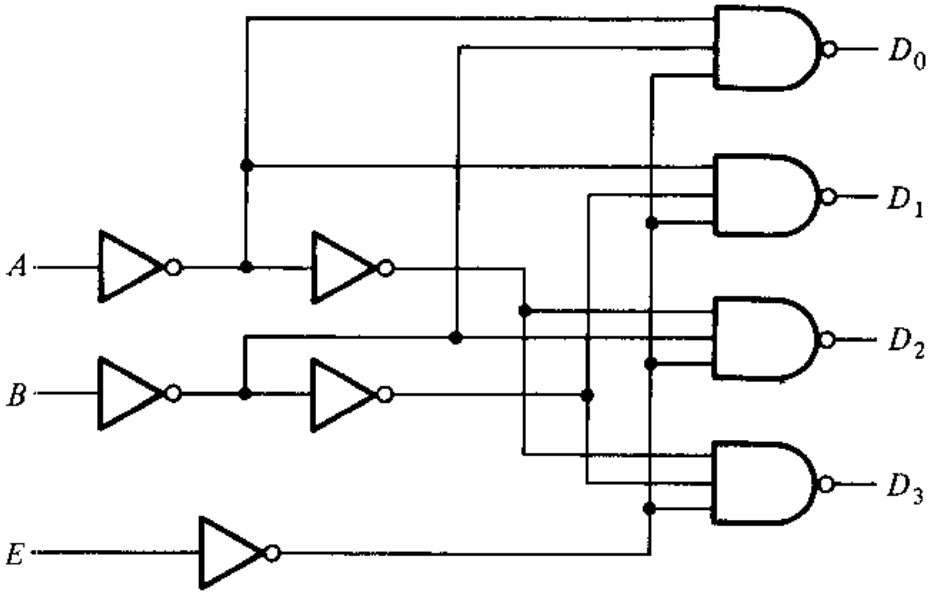
- Some decoders have enable inputs. If the enable is not asserted, all outputs are inactive

S_1	S_0	EN	D_3	D_2	D_1	D_0
0	0	1	0	0	0	1
0	1	1	0	0	1	0
1	0	1	0	1	0	0
1	1	1	1	0	0	0
X	X	0	0	0	0	0



Decoder with an enable

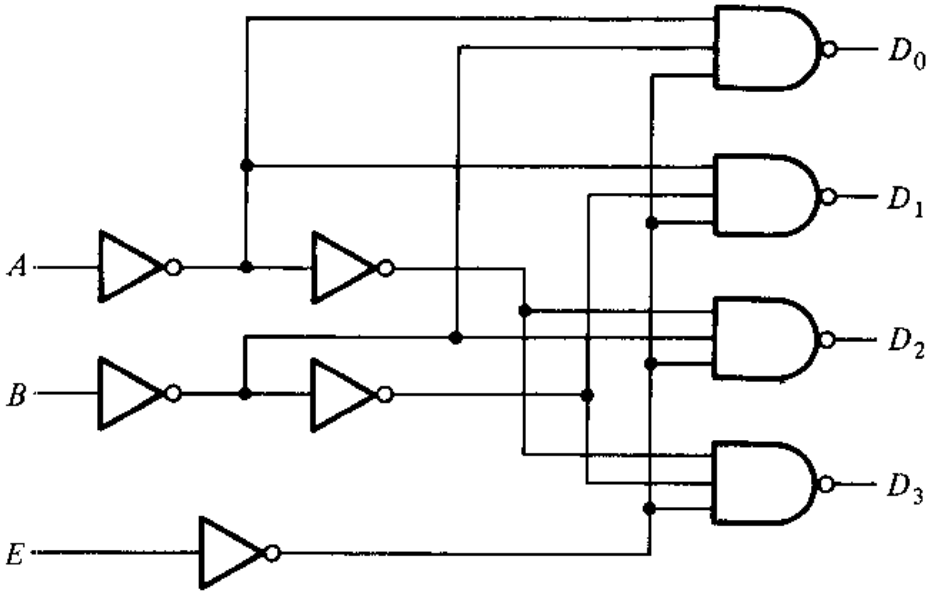
- 2-to-4 decoder with an enable input constructed with NAND gates.
  - If enable input E=1 all outputs are equal to 1
  - If E=0 the circuit operates as a decoder with complemented outputs.

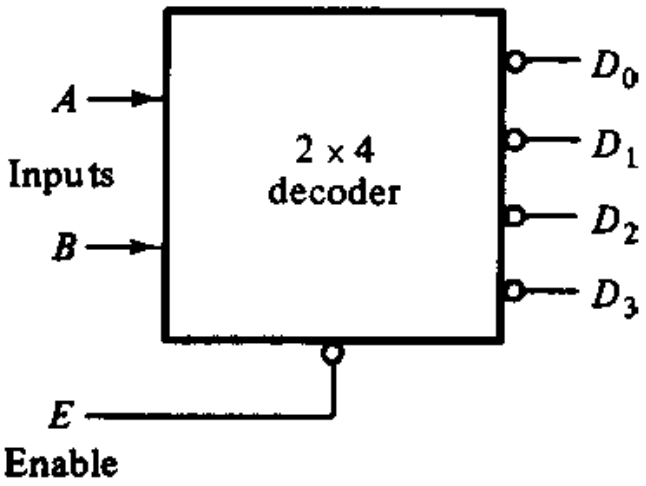


E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

A decoder with an enable

- 2-to-4 decoder with an enable input constructed with NAND gates.
  - If enable input E=1 all outputs are equal to 1
  - If E=0 the circuit operates as a decoder with complemented outputs.
  - The small circle at input E indicates that the decoder is enabled when E=0 in the block diagram.

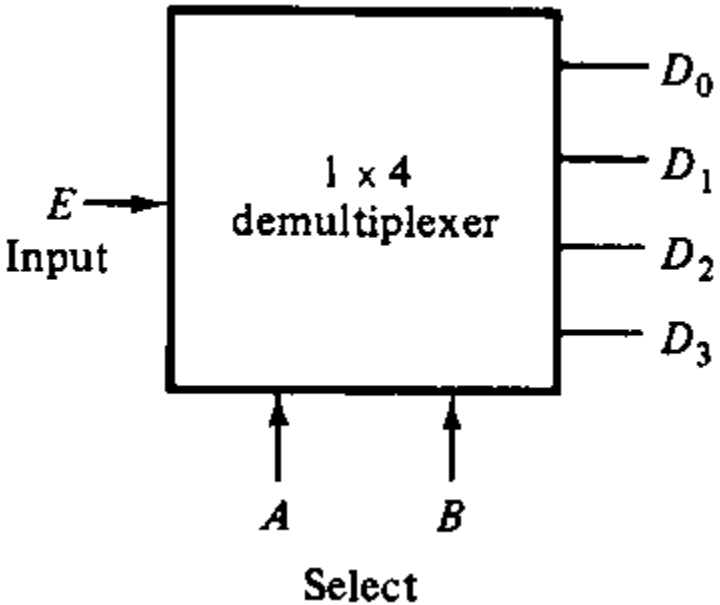




Demultiplexers

- A decoder with an enable input can function as a Demultiplexer.
- A demultiplexer is a circuit that receives information on a single line and transmits this information on one of 2^n possible output lines.
- Selection of a specific output line is controlled by the bit values of n selection lines.
- E line is taken as a data input line and line A & B are taken as the selection lines.
- Example from truth table, AB=10 output D_2 will be same as the input E

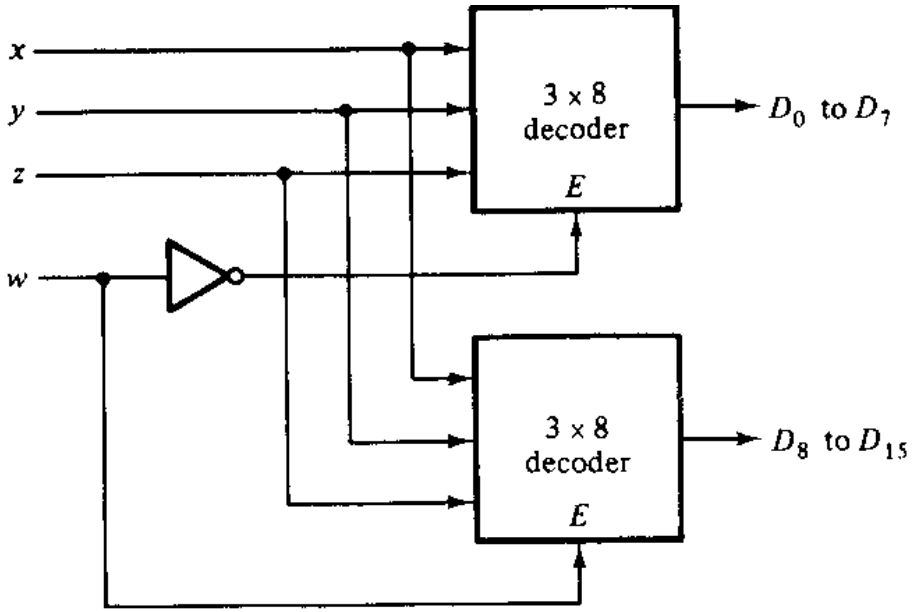
E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0



Decoder/Demultiplexers

- Decoder/Demultiplexer circuits can be connected together to form a large decoder circuit.
- Show how 4 X 16 decoder can be constructed with 3 X 8 decoders connected with enable inputs ?

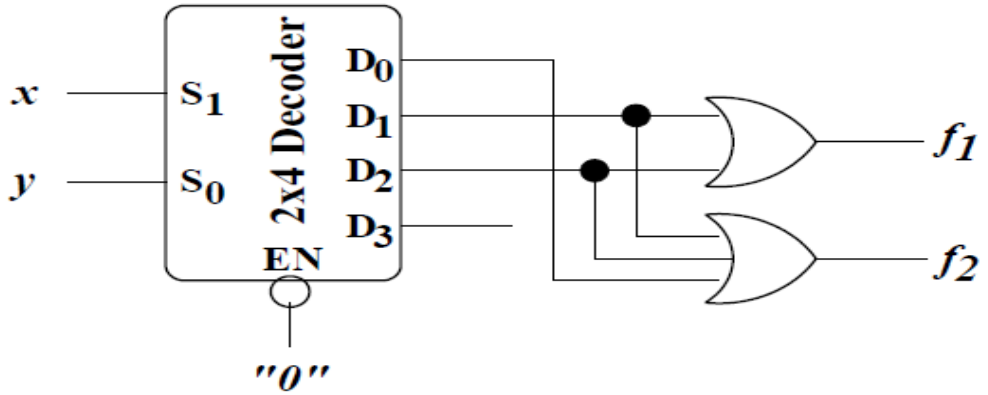
- When w=0, the top decoder is enabled and the other is disabled.
- Bottom decoder outputs are all 0's & top outputs generate minterms 0000 to 0111
- When w=1 bottom decoder outputs generate minterms 1000 to 1111
- Enable lines are a convenient feature for connecting two or more IC packages for the purpose of expanding the digital function into a similar function with more inputs and outputs.



Decoder

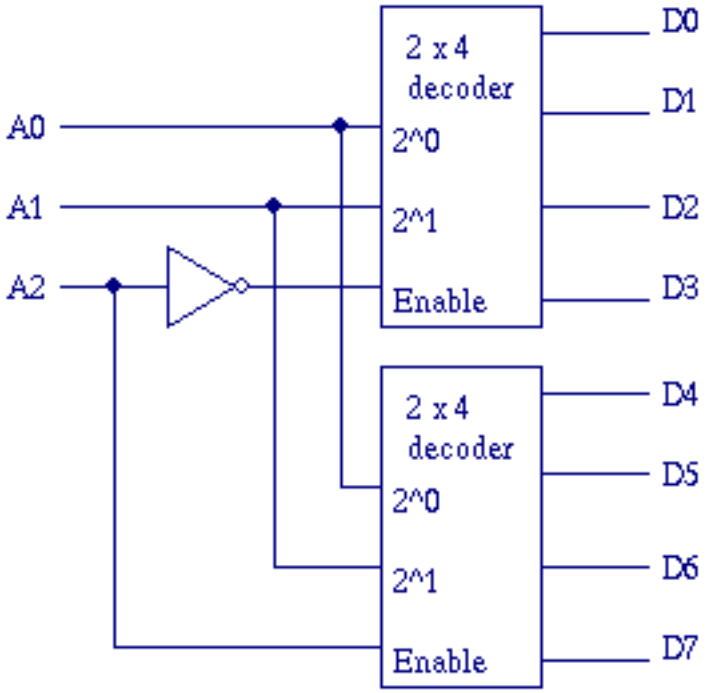
- Decoders can be used to implement logic functions as follows:

x	y	f_1	f_2
0	0	0	1
0	1	1	1
1	0	1	1
1	1	0	0



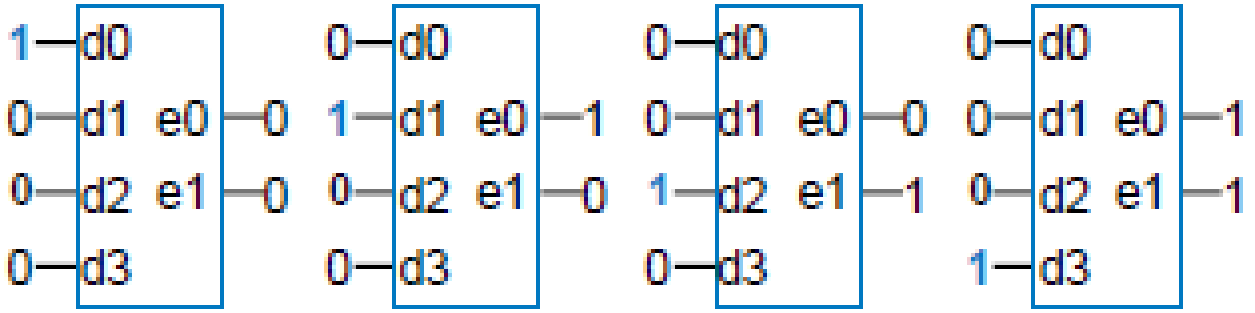
- A large size of decoder can be constructed by cascading smaller decoders with enable lines to form a decoder tree.

How to build a 3X8 decoder using 2X4 decoders?

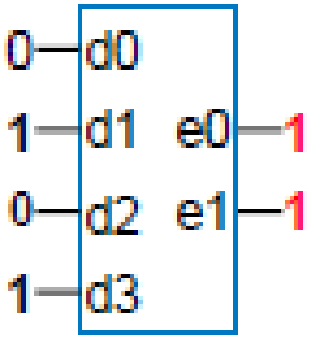


Encoder

- Encoder: Combinational logic building block with opposite functionality of decoder.
  - Outputs binary encoding for input signal that is 1
  - 4x2 encoder would have four inputs and 2 outputs.



- What if two inputs are 1 ?
  - Can use a priority encoder
  - Gives priority to the highest input that is 1, and outputs binary encoding for that input
  - Example: If d3=1 and d1=1, will output e0=1 and e1=1 because d3 has priority



Priority encoder

- An example of a single bit 4 to 2 encoder is shown, where highest-priority inputs are to the left and "x" indicates an irrelevant value - i.e. any input value there yields the same output since it is superseded by higher-priority input.
- If the input n is active, all lower inputs (n-1 .. 0) are ignored:

d3	d2	d1	d0	a1	a0	v
0	0	0	0	x	x	0
0	0	0	x	0	0	1
0	0	1	x	0	1	1
0	1	x	x	1	0	1
1	x	x	x	1	1	1

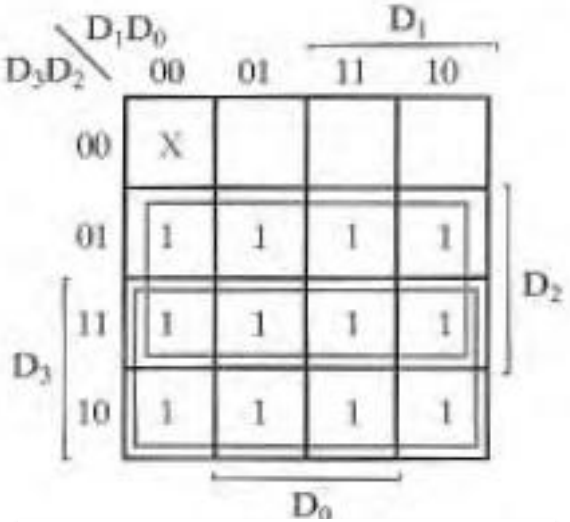
Priority encoder

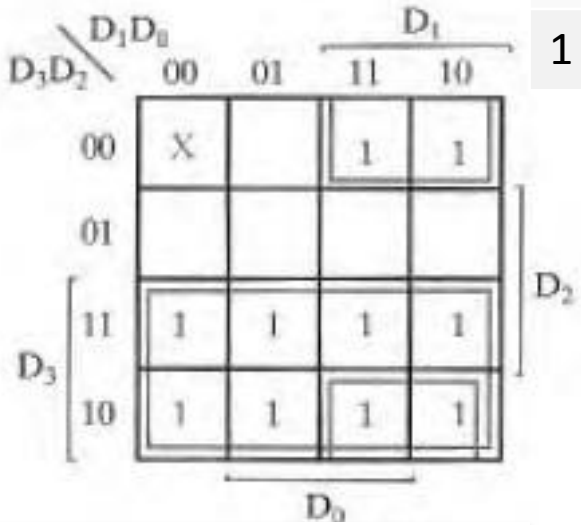
$$A_0 = D_3 + D_1\overline{D}_2$$

$$A_1 = D_2 + D_3$$

$$V = D_0 + D_1 + D_2 + D_3$$

D3	D2	D1	D0	A1	A0	V
0	0	0	0	X	X	0
0	0	0	x	0	0	1
0	0	1	x	0	1	1
0	1	x	x	1	0	1
1	x	x	x	1	1	1



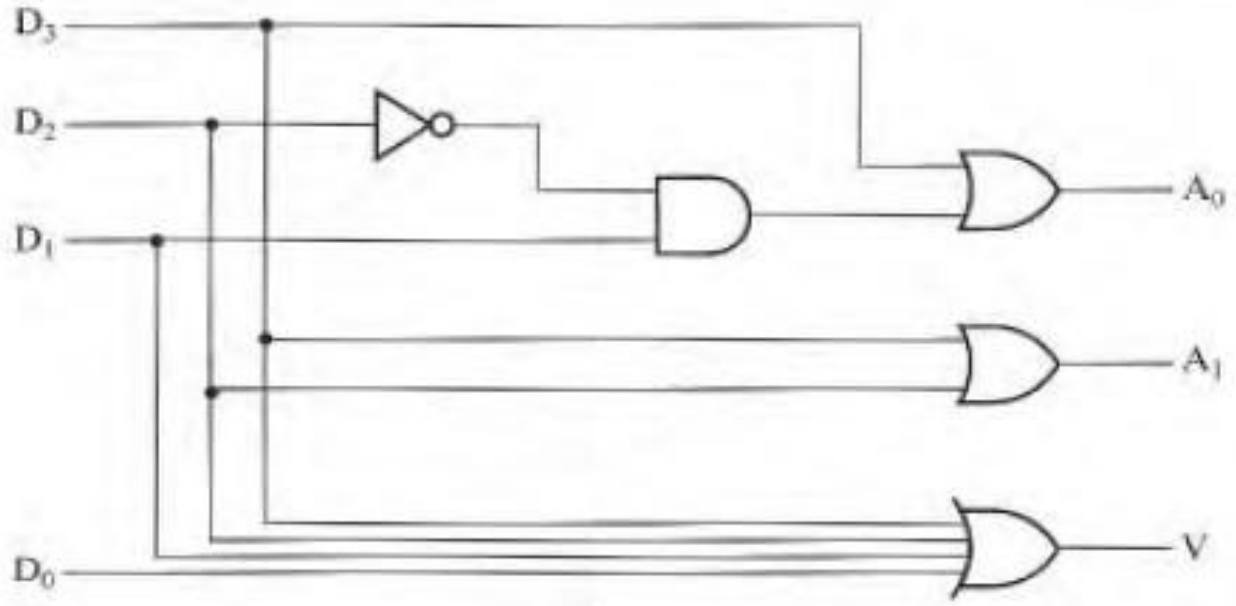


Priority encoder circuit

$$A_0 = D_3 + D_1\overline{D}_2$$

$$A_1 = D_2 + D_3$$

$$V = D_0 + D_1 + D_2 + D_3$$

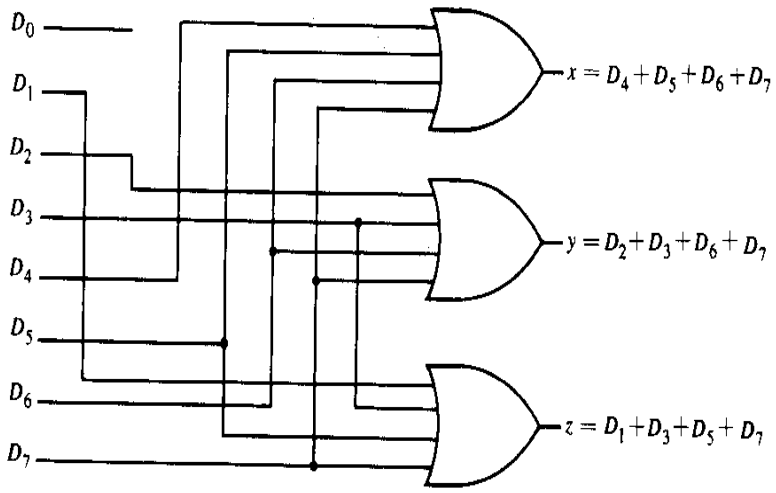


Encoder

- Following example has shown, Octal-to-binary encoding

Truth Table of Octal-to-Binary Encoder

Inputs								Outputs		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	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

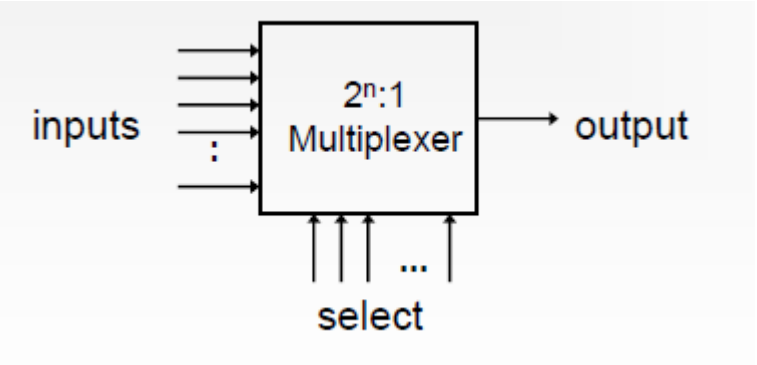


Multiplexor (MUX)

A multiplexer is a device which has

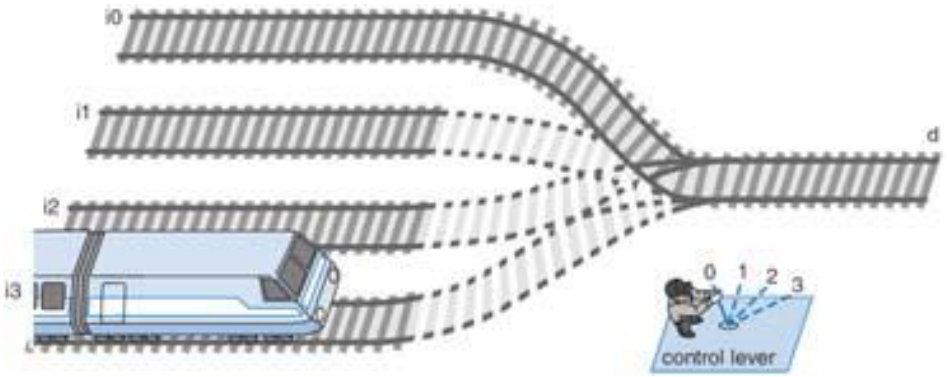
- a number of *inputlines*
- a number of *selectionlines*
- one *outputline*

- It steers one of 2^n inputs to a single output line, using *n* selection lines. Also known as a data selector.



Multiplexor (Mux)

- Mux: Another popular combinational building block
  - Routes one of its N data inputs to its one output, based on binary value of select inputs
    - 4 input mux needs 2 select inputs to indicate which input to route through
    - 8 input mux needs 3 select inputs
    - N inputs → $\log_2(N)$ selects
  - Like a railyard switch

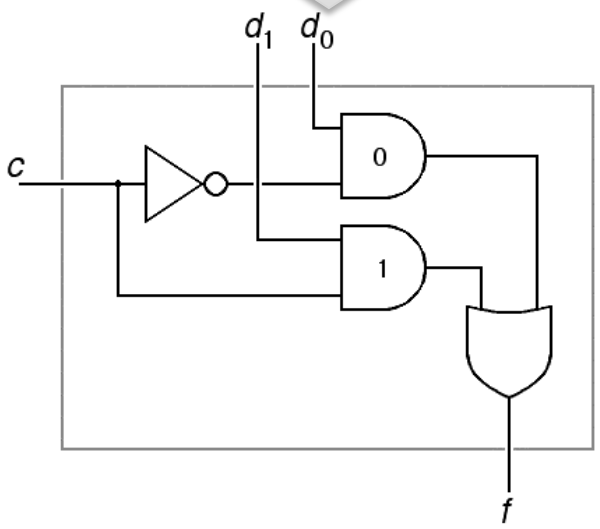


A 2-input multiplexor

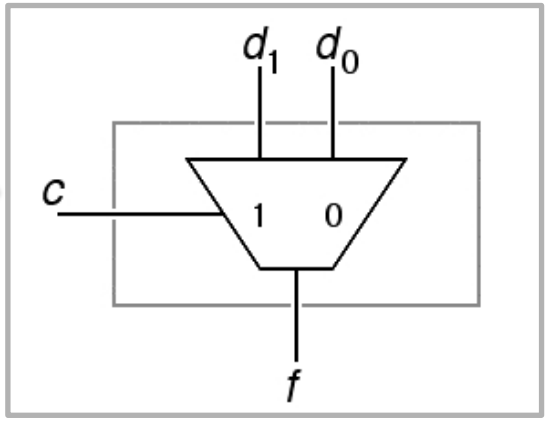
- Truth table for a multiplexor with 2 data inputs d_0 and d_1 and one control input c is as follows:

inputs			f
c	d_0	d_1	
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$$f = (d_0\,\bar{c}) + (d_1\,c)$$



logic circuit involving only 4 gates



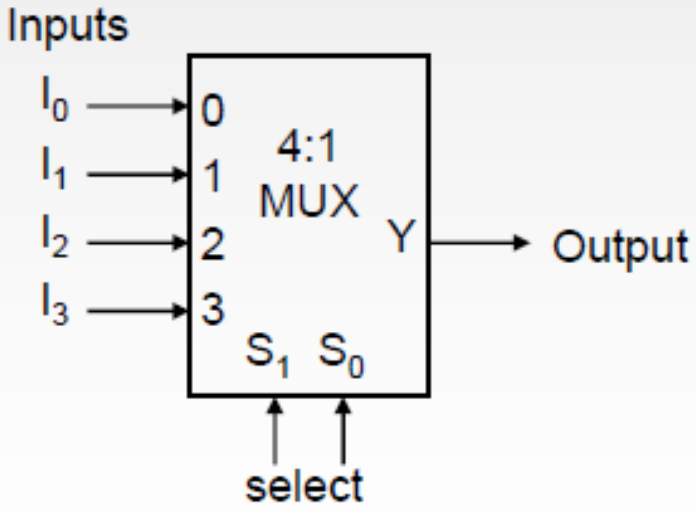
schematic symbol

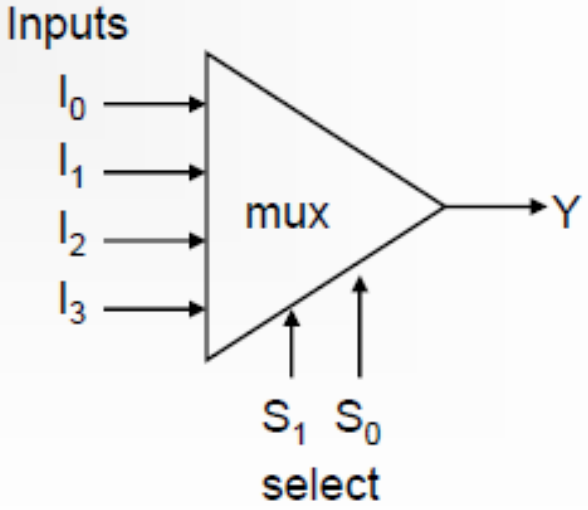
4-input multiplexor

- Truth table for a 4-to-1 multiplexer:

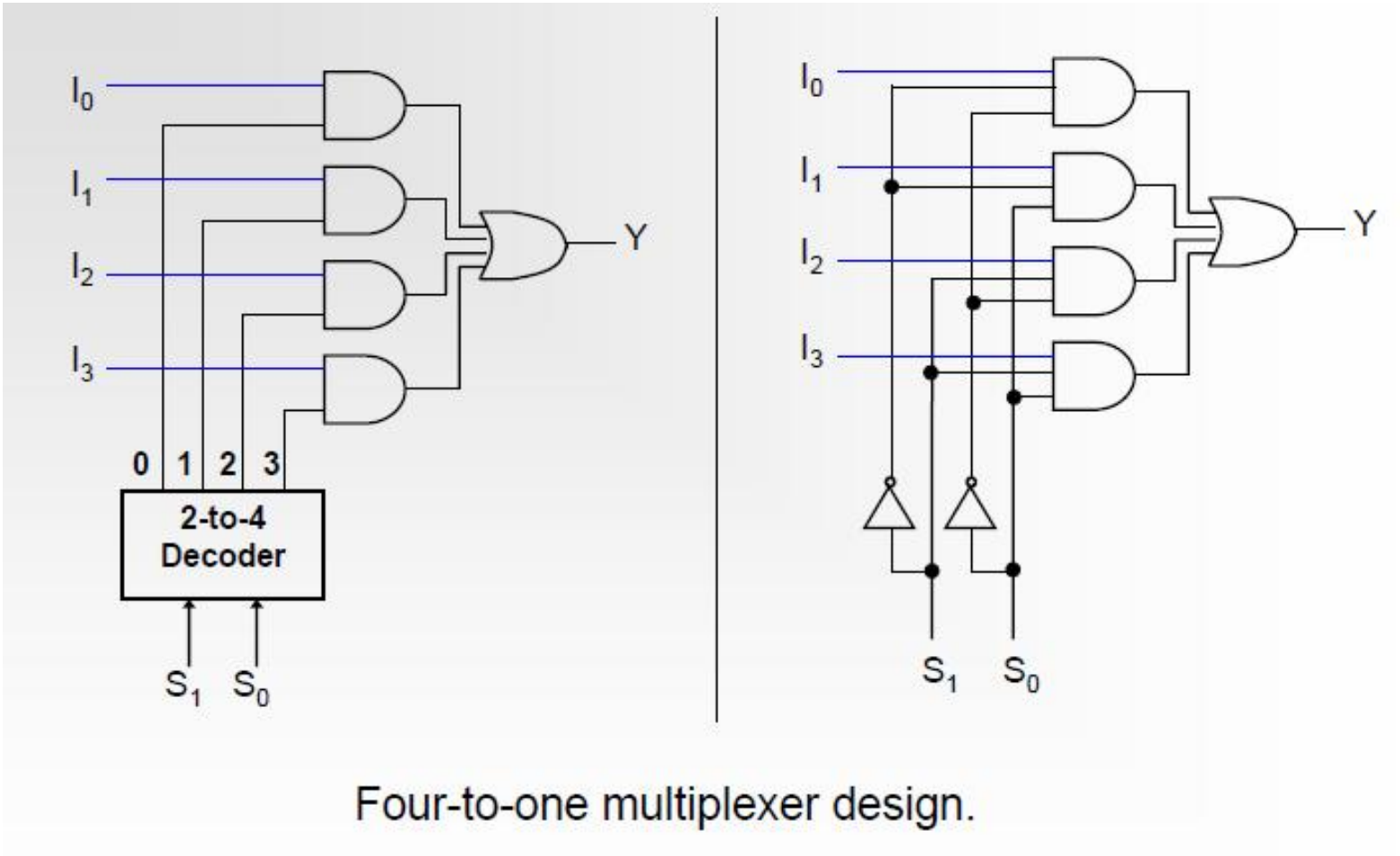
I_0	I_1	I_2	I_3	S_1	S_0	Y
d_0	d_1	d_2	d_3	0	0	d_0
d_0	d_1	d_2	d_3	0	1	d_1
d_0	d_1	d_2	d_3	1	0	d_2
d_0	d_1	d_2	d_3	1	1	d_3

S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3





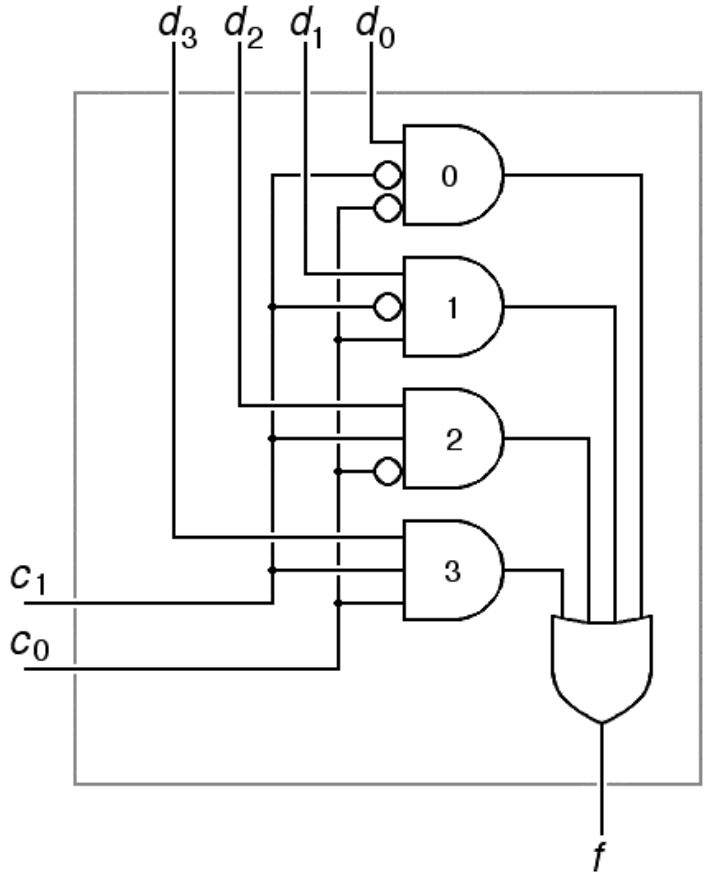
4-input multiplexor

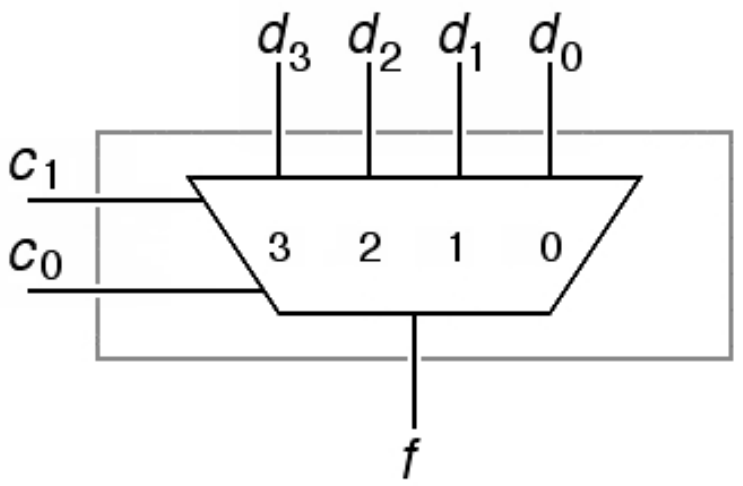


Four-to-one multiplexer design.

Gate-level design for a 4-input multiplexor

$$f = (d_0\, c'_1\, c'_0) + (d_1\, c'_1\, c_0) + (d_2\, c_1\, c'_0) + (d_3\, c_1\, c_0)$$

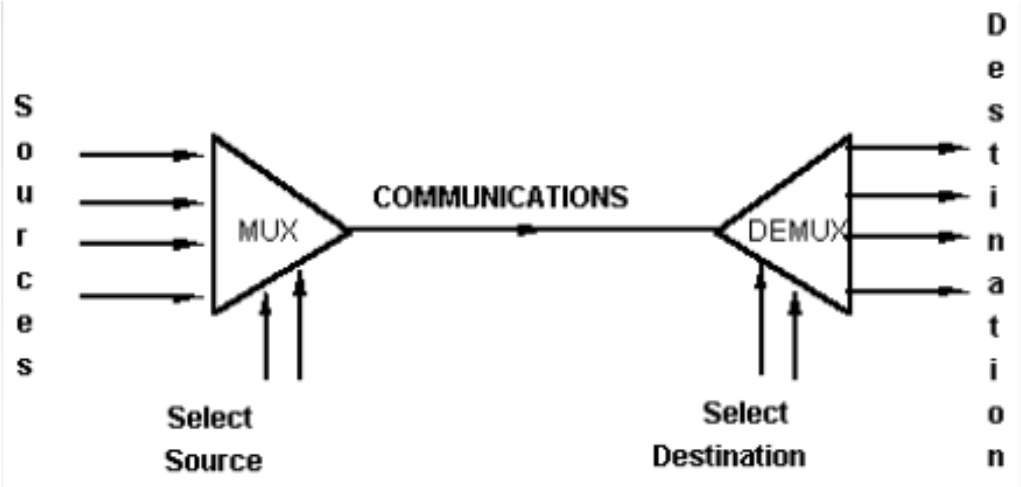




schematic symbol

Multiplexer

An application:



- Helps share a single communication line among a number of devices.
- At any time, only one source and one destination can use the communication line.