ENGR 2218 – Digital Logic
Comparators, Decoders, and Multiplexers

Comparators

- Compares binary numbers
- Uses the XOR or XNOR

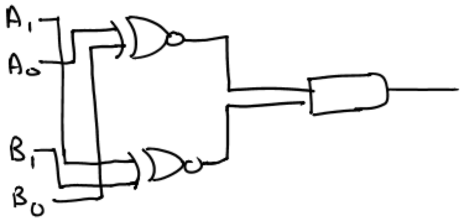
A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

A	B	XNOR
0	0	1
0	1	0
1	0	0
1	1	1

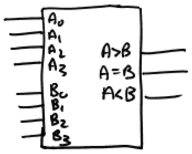
2-bit Comparators

$A_1 A_0$

$B_1 B_0$



4-bit Comparator with Inequality



$\underline{A=B}$

$$X_E = \left(\overline{A_3 \oplus B_3}\right)\left(\overline{A_2 \oplus B_2}\right)\left(\overline{A_1 \oplus B_1}\right)\left(\overline{A_0 \oplus B_0}\right)$$

$\underline{A>B}$

$A_3\ A_2\ A_1\ A_0$

$B_3\ B_2\ B_1\ B_0$

If $A_3=1,\ B_3=0 \Rightarrow A>B$

If $A_3=0,\ B_3=1 \Rightarrow A<B$

If $A_3=B_3 \Rightarrow$ Look at the next significant bit

A_i	B_i	X_{Gi}
0	0	0
0	1	0
1	0	1 $\Rightarrow A_i \overline{B_i}$
1	1	0

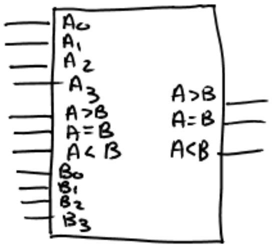
$$\boxed{\begin{aligned} X_G = {} & A_3\overline{B_3} + A_2\overline{B_2}\left(\overline{A_3 \oplus B_3}\right) + A_1\overline{B_1}\left(\overline{A_3 \oplus B_3}\right)\left(\overline{A_2 \oplus B_2}\right) \\ & + A_0\overline{B_0}\left(\overline{A_3 \oplus B_3}\right)\left(\overline{A_2 \oplus B_2}\right)\left(\overline{A_1 \oplus B_1}\right) \end{aligned}}$$

<u>A<B</u>

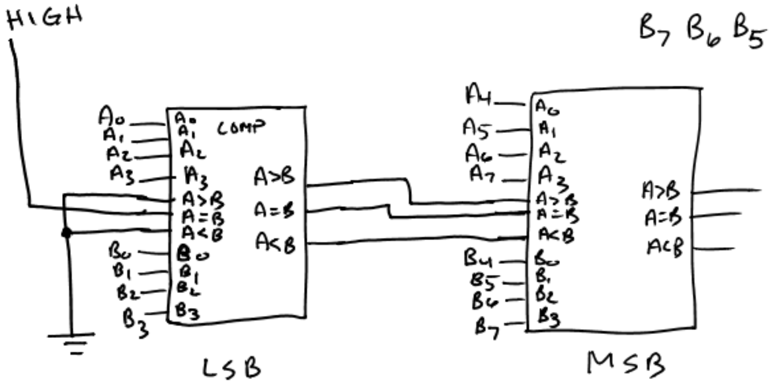
A_i	B_i	X_{Li}
0	0	0
0	1	$1 = \overline{A_i} B_i$
1	0	0
1	1	0

$$\boxed{\begin{aligned} X_L = {} & \overline{A_3} B_3 + \overline{A_2} B_2 (\overline{A_3 \oplus B_3}) + \overline{A_1} B_1 (\overline{A_3 \oplus B_3})(\overline{A_2 \oplus B_2}) \\ & + \overline{A_0} B_0 (\overline{A_3 \oplus B_3})(\overline{A_2 \oplus B_2})(\overline{A_1 \oplus B_1}) \end{aligned}}$$

Cascading
- Adds three inputs

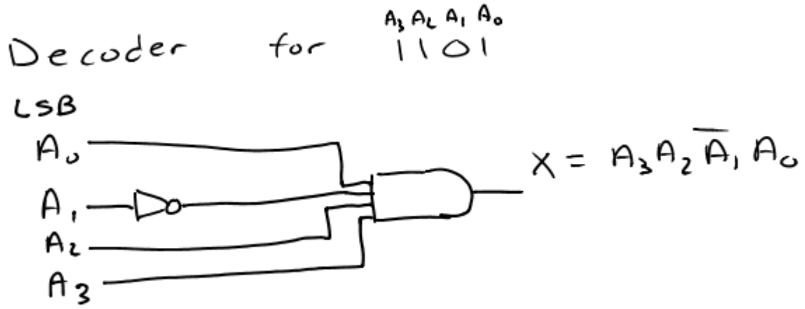


$A_7\ A_6\ A_5\ A_4$	$A_3\ A_2\ A_1\ A_0$
$B_7\ B_6\ B_5\ B_4$	$B_3\ B_2\ B_1\ B_0$

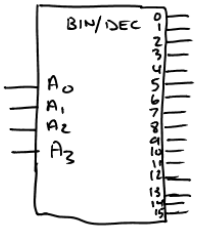


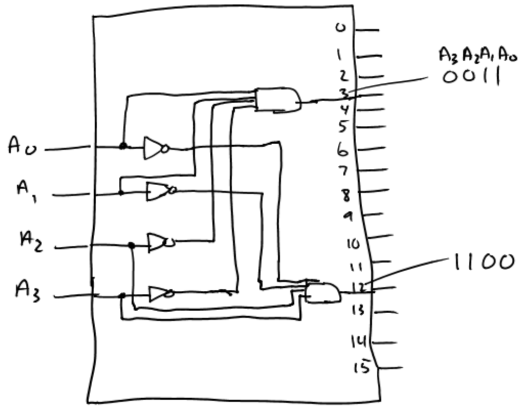
Decoders

Basic Binary Decoder
- Determines whether a binary number occurs in a circuit
- Output is 1 if a binary number occurs



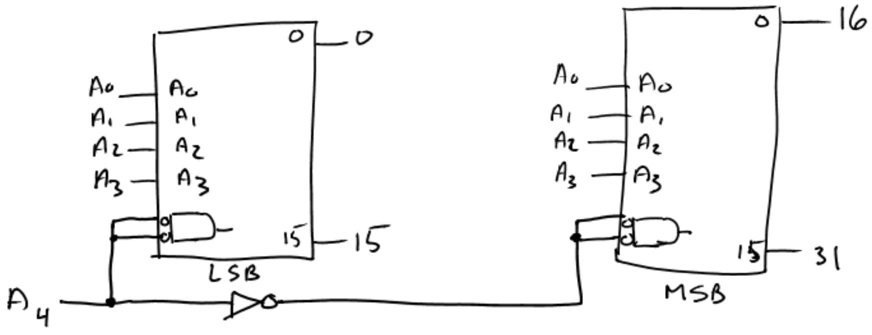
4-bit Decoder (1 of 16 Decoder)





Cascading

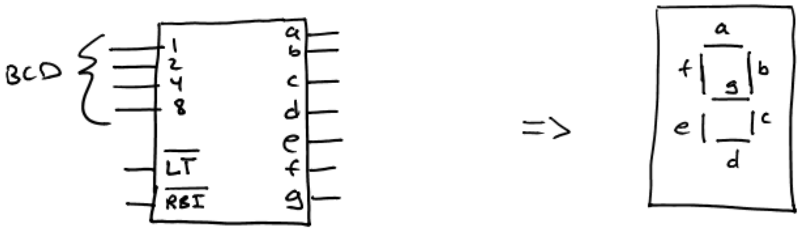
- Two additional inputs (Enablers)
- Two 4-bit decoders can be cascaded to form a 5-bit decoder



If $A_4 = 0$ Enabler LSB = 1
MSB = 0

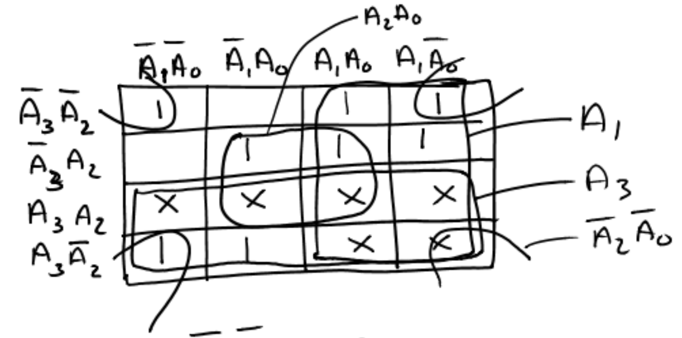
$A_4 = 1$ Enabler LSB = 0
MSB = 1

BCD-to-7 Segment Decoder



Decimal Digit	A_3	A_2	A_1	A_0	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

a

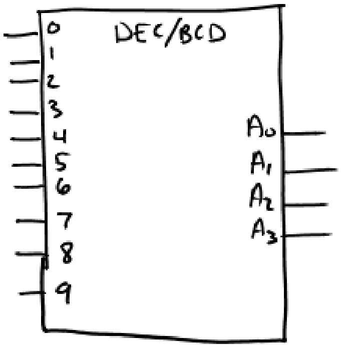


	$\bar{A}_1\bar{A}_0$	$\bar{A}_1A_0$	A_1A_0	$A_1\bar{A}_0$
$\bar{A}_3\bar{A}_2$	1		1	1
$\bar{A}_3A_2$		1	1	1
A_3A_2	X	X	X	X
$A_3\bar{A}_2$	1	1	X	X

$$a = A_1 + A_3 + \bar{A}_2\bar{A}_0 + A_2A_0$$
$$= A_1 + A_3 + (\overline{A_2 \oplus A_0})$$

Encoders

- Decimal to BCD Encoder



Decimal	A_3	A_2	A_1	A_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

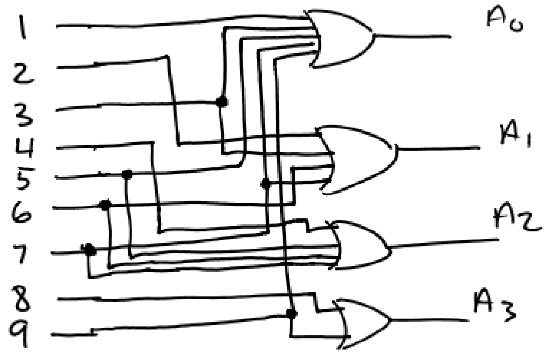
$A_3 = 1$ if 8 OR 9

$A_2 = 1$ if 4, 5, 6, OR 7

$A_1 = 1$ if 2, 3, 6, OR 7

$A_0 = 1$ if 1, 3, 5, 7, OR 9

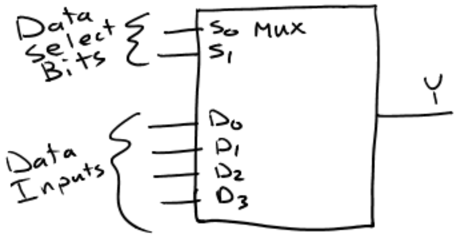
Don't Need a 0 input



Multiplexers (Data Selectors, MUX)

- Allows for multiple lines of data to go onto a single line
- Parallel to Serial Data Transfer

4-input Multiplexer



Data Select Bits

Combinations of the Data Select Bits represent each of the data inputs

S_1	S_0	Input selected	
0	0	D_0	$D_0 \overline{S_1}\,\overline{S_0}$
0	1	D_1	$D_1 \overline{S_1} S_0$
1	0	D_2	$D_2 S_1 \overline{S_0}$
1	1	D_3	$D_3 S_1 S_0$

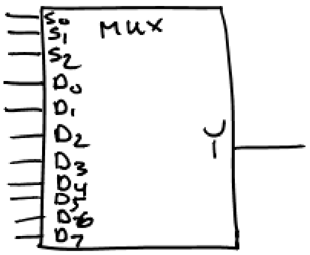
$$\boxed{Y = D_0 \overline{S_1}\,\overline{S_0} + D_1 \overline{S_1} S_0 + D_2 S_1 \overline{S_0} + D_3 S_1 S_0}$$

8-input Multiplexer

- Additional Data Select Bit

S_2	S_1	S_0	Input Selected	
0	0	0	D_0	$D_0 \overline{S_2}\,\overline{S_1}\,\overline{S_0}$
0	0	1	D_1	$D_1 \overline{S_2}\,\overline{S_1} S_0$
0	1	0	D_2	$D_2 \overline{S_2} S_1 \overline{S_0}$
0	1	1	D_3	$D_3 \overline{S_2} S_1 S_0$
1	0	0	D_4	$D_4 S_2 \overline{S_1}\,\overline{S_0}$
1	0	1	D_5	$D_5 S_2 \overline{S_1} S_0$
1	1	0	D_6	$D_6 S_2 S_1 \overline{S_0}$
1	1	1	D_7	$D_7 S_2 S_1 S_0$

$$\boxed{\begin{aligned} Y = {} & D_0 \overline{S_2}\,\overline{S_1}\,\overline{S_0} + D_1 \overline{S_2}\,\overline{S_1} S_0 + D_2 \overline{S_2} S_1 \overline{S_0} + D_3 \overline{S_2} S_1 S_0 \\ & + D_4 S_2 \overline{S_1}\,\overline{S_0} + D_5 S_2 \overline{S_1} S_0 + D_6 S_2 S_1 \overline{S_0} + D_7 S_2 S_1 S_0 \end{aligned}}$$



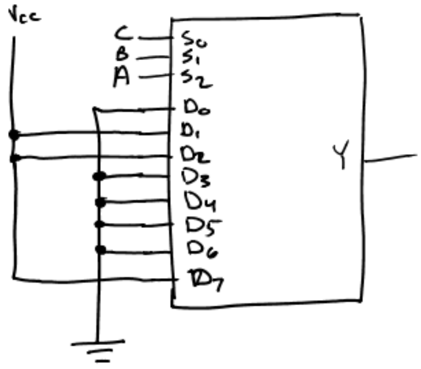
- Can use Multiplexers to implement Standard SOP expressions
  - Inputs: Data Select Bits
  - D's: Activate or deactivate product terms

Implement $X = \underset{\bar{S}_2}{\bar{A}}\,\underset{\bar{S}_1}{\bar{B}}\,\underset{S_0}{C} + \underset{\bar{S}_2}{\bar{A}}\,\underset{S_1}{B}\,\underset{\bar{S}_0}{\bar{C}} + \underset{S_2}{A}\,\underset{S_1}{B}\,\underset{S_0}{C}$

$A = S_2 \quad B = S_1 \quad C = S_0$

Power: D_1, D_2, D_7

Ground: D_0, D_3, D_4, D_5, D_6



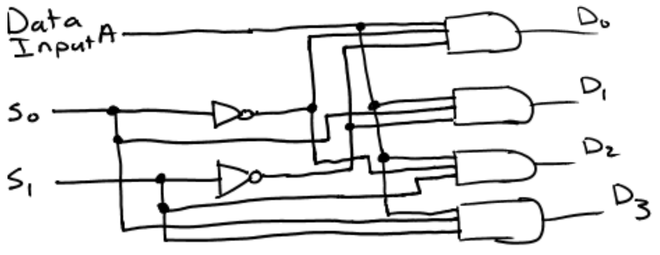
Demultiplexers (DEMUX)

- Serial to Parallel data transfer

1 to 4 lines

Input

S_1	S_0	Output selected	
0	0	D_0	$\bar{S}_1\bar{S}_0A$
0	1	D_1	$\bar{S}_1S_0A$
1	0	D_2	$S_1\bar{S}_0A$
1	1	D_3	S_1S_0A


Data Input A
S0
S1
D0
D1
D2
D3