

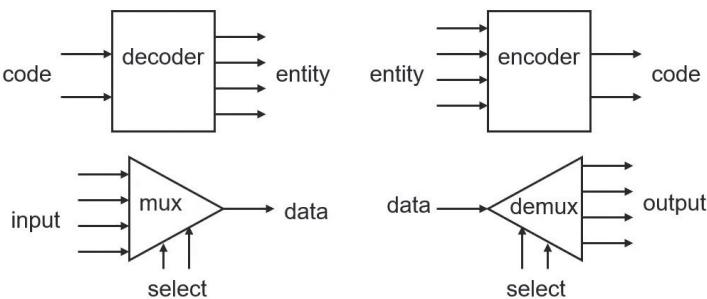
MSI Components

· **Integrated circuit (IC)** - a set of electronic circuits on one small flat piece of semiconductor material.

· **Scale of integration** - the number of components fitted into a standard size IC.

Name	Signification	Year	#transistors	#logic gates
SSI	Small-scale integration	1964	1 to 10	1 to 12
MSI	Medium-scale integration	1968	10 to 500	13 to 99
LSI	Large-scale integration	1971	500 to 20000	100 to 9999
VLSI	Very large-scale integration	1980	20k to 1m	10k to 99999
ULSI	Ultra-large-scale integration	1984	1m and more	100k and more

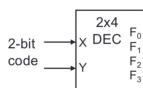
· Common MSI Circuits:



Decoders

- Codes are frequently used to represent entities
- Codes can be identified (or decoded) using a decoder
(Given a code, identify the entity.)
- Convert binary information from n input lines to (a maximum of) 2^n output lines.
- Known as n -to- m -line decoder, or $n:m$ or $n \times m$ ($m \leq 2^n$)
- May be used to generate 2^n minterms of n input variables.

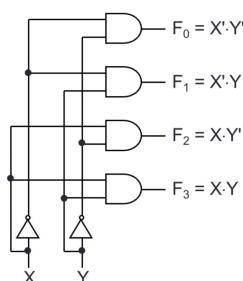
- Example: If codes 00, 01, 10, 11 are used to identify four light bulbs, we may use a 2-bit decoder.



X	Y	F_0	F_1	F_2	F_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

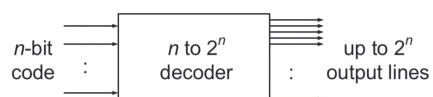
- From truth table, circuit for 2x4 decoder is:

- Note: Each output is a minterm ($X'Y'$, $X'Y$, XY' or XY) of a 2-variable function



In general:

for a n -bit code, a decoder can select up to 2^n lines



Direct application of decoders - Implementing Functions.

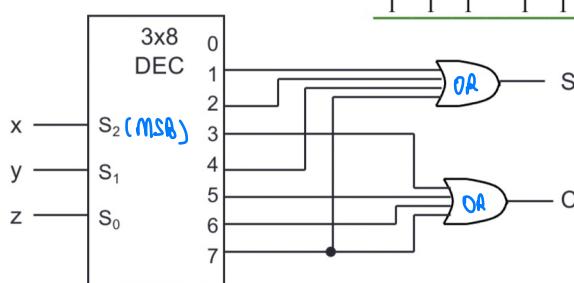
- A boolean function, in **sum-of-minterms** form,
 - decoder to generate the minterms, and
 - an OR gate to form the sum.
- Any combinational circuit with n inputs and m outputs can be implemented with an $n \cdot 2^n$ decoder with m OR gates.
- Good when circuits has many outputs, and each function is expressed with a few minterms.

Example: Full adder

$$S(x, y, z) = \sum m(1, 2, 4, 7)$$

$$C(x, y, z) = \sum m(3, 5, 6, 7)$$

x	y	z	C	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



Enable

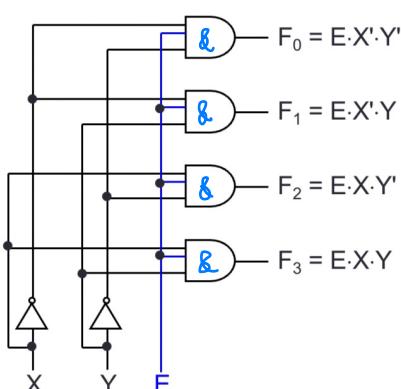
- Decoders often comes with an **enable control** signal, so that the device is only activated when the enable, $E = 1$. (**one-enable**)

Truth table:

E	X	Y	F ₀	F ₁	F ₂	F ₃
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1
0	d	d	0	0	0	0

don't care

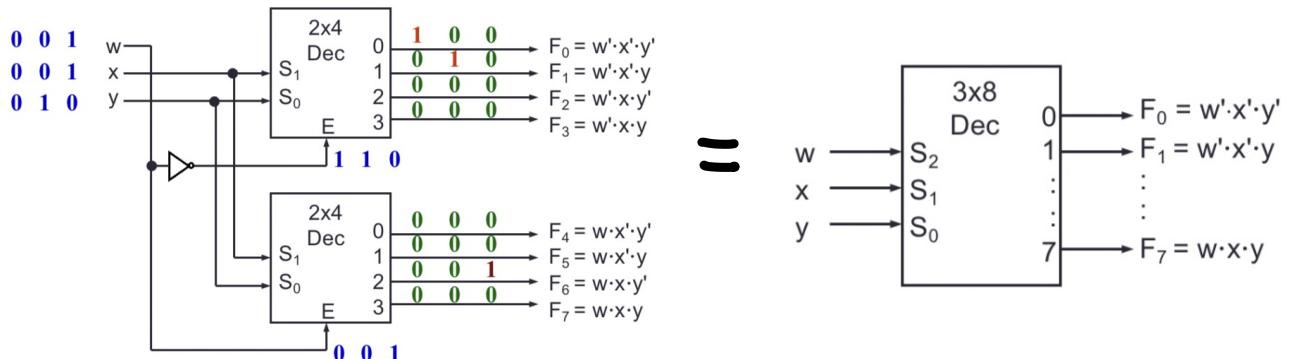
- Circuit of a 2×4 decoder with enable:



- In most MSI decoders, enable signal is **zero-enable**, usually denoted by E' or \bar{E} . The decoder is enabled when the signal is **zero** (low).

Constructing Larger Decoders

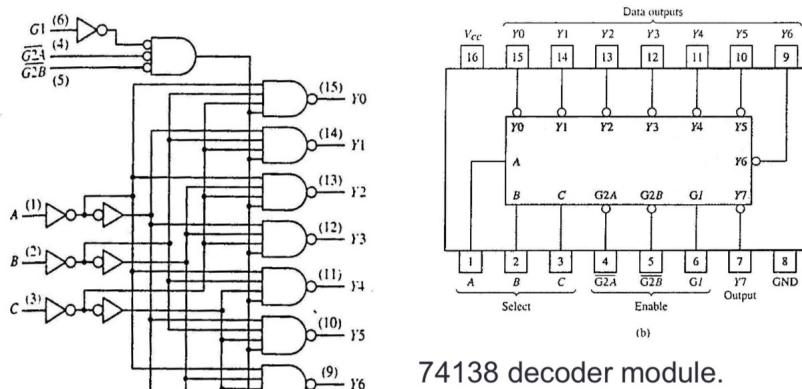
- Larger decoders can be constructed from smaller univ.



- The input **W** and its complement **w'** are used to select either one of the 2 smaller decoders.
- Decoders may also have **negated outputs** (active low outputs)

• Standard MSI Decoder

■ 74138 (3-to-8 decoder)



74138 decoder module.
(a) Logic circuit.
(b) Package pin configuration.

INPUTS		OUTPUTS							
ENABLE	SELECT	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
disabled	X H	X X X	H H H H	H H H H	H H H H	H H H H	H H H H	H H H H	H H H H
	L X	X X X	H H H H	H H H H	H H H H	H H H H	H H H H	H H H H	H H H H
	H L	L L L	L H H H	H H H H	H H H H	H H H H	H H H H	H H H H	H H H H
	H L	L L H	H H H H	H H H H	H H H H	H H H H	H H H H	H H H H	H H H H
	H L	L H H	H H H L	H H H L	H H H L	H H H L	H H H L	H H H L	H H H L
	H L	H L L	H H H H	H H H H	H H H H	H H H H	H H H H	H H H H	H H H H
	H L	H L H	H H H H	H H H H	H H H H	H H H H	H H H H	H H H H	H H H H
	H L	H H H	H H H H	H H H H	H H H H	H H H H	H H H H	H H H H	H H H H
enabled		H L	H H H	H H H H	H H H H	H H H H	H H H H	H H H H	H H H H

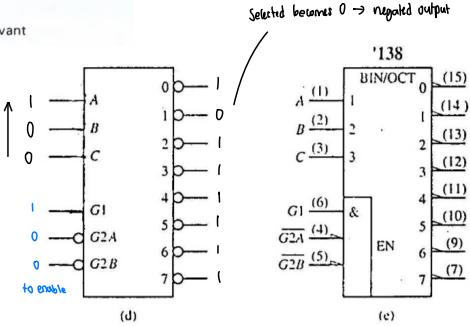
* $\bar{G}_2 = \bar{G}_{2A} + \bar{G}_{2B}$
H = high level, L = low level, X = irrelevant

(c)

74138 decoder module.
(c) Function table.

74138 decoder module.
(d) Generic symbol.
(e) IEEE standard logic symbol.

Source: The Data Book Volume 2, Texas Instruments Inc., 1985



Implementing Functions

- 3x8 Decoder

$$f(Q, X, P) = \sum m(0, 1, 4, 6, 7) = \prod M(2, 3, 5)$$

- function implementation:
- active-high decoder w OR gate

$$f = m_0 + m_1 + m_4 + m_6 + m_7$$

- active-low decoder w NAND gate

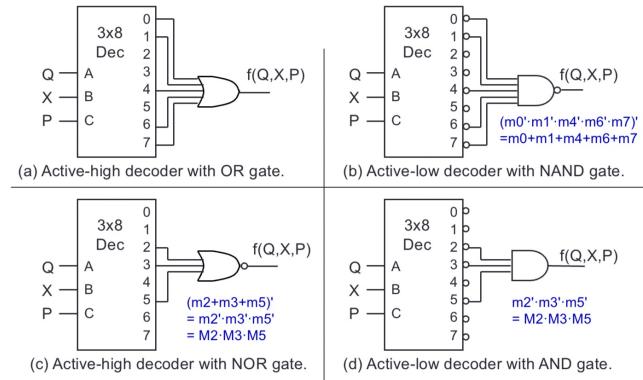
$$f = (m_0' \cdot m_1' \cdot m_4' \cdot m_6' \cdot m_7')'$$

- active-high decoder w NOR gate

$$f = (m_2 + m_3 + m_5)' [= M_2 \cdot M_3 \cdot M_5]$$

- active-low decoder w NAND gate

$$f = m_2' \cdot m_3' \cdot m_5'$$



Encoders

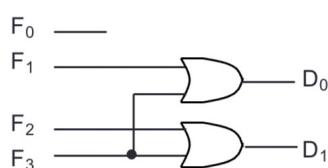
- Encoding is the converse of decoding.
- Given a set of input lines, of which exactly one is high and the rest are low, the encoder provides a code that corresponds to that high input line.
- Contains 2^n (or fewer) input lines and n output lines.
- Implemented with OR gates

- Truth table:

- With K-map, we obtain:

- $D_0 = F_1 + F_3$
- $D_1 = F_2 + F_3$

- Circuit:



Simple 4-to-2 encoder

F_0	F_1	F_2	F_3	D_1	D_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1
0	0	0	0	X	X
0	0	1	1	X	X
0	1	0	1	X	X
0	1	1	0	X	X
0	1	1	1	X	X
1	0	0	1	X	X
1	0	1	0	X	X
1	0	1	1	X	X
1	1	0	0	X	X
1	1	0	1	X	X
1	1	1	0	X	X
1	1	1	1	X	X

• At any one time, only one input line of an encoder has a value of 1 (high), the rest are zeroes (low)

• To allow for more than one input line to carry a 1, we need priority encoder.

invalid.

Priority Encoder

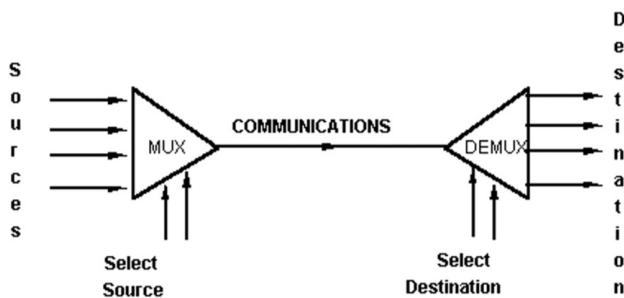
- If 2 or more inputs is equal to 1, the input with the highest priority takes precedence.
- If all inputs are 0, this input combination is considered invalid.

eg.

Inputs				Outputs		
D_0	D_1	D_2	D_3	f	g	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

compact truth table { indicate 1 if valid

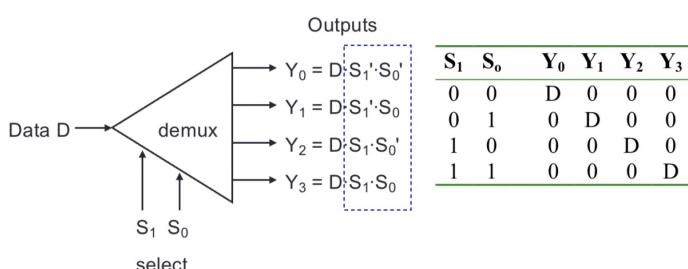
Multiplexers and Demultiplexers



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

Demultiplexers

- Given an input line and a set of selection lines, a demultiplexer directs data from the input to one selected output line.

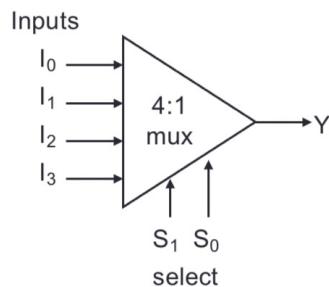
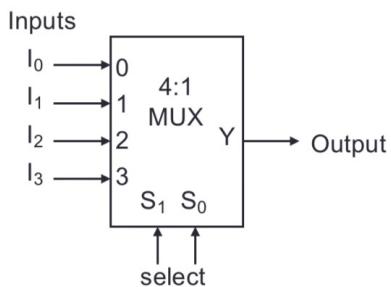


Multiplexers

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

I₀	I₁	I₂	I₃	S₁	S₀	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

S₁	S₀	Y
0	0	I ₀
0	1	I ₁
1	0	I ₂
1	1	I ₃



- Output of multiplexer is

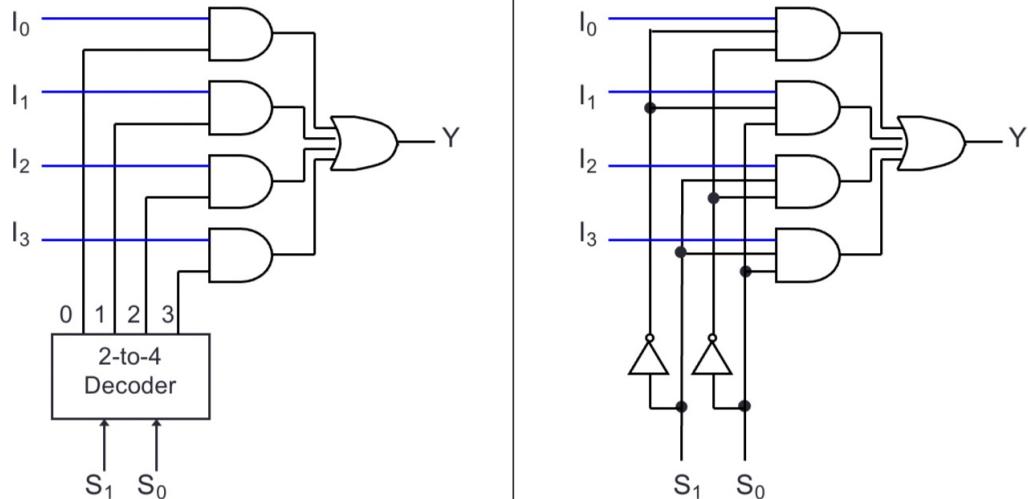
"sum of the (product of data lines and selection lines)"

eg.
$$Y = I_0 \cdot (S_1 \cdot S_0') + I_1 \cdot (S_1' \cdot S_0') + I_2 \cdot (S_1 \cdot S_0) + I_3 \cdot (S_1' \cdot S_0)$$

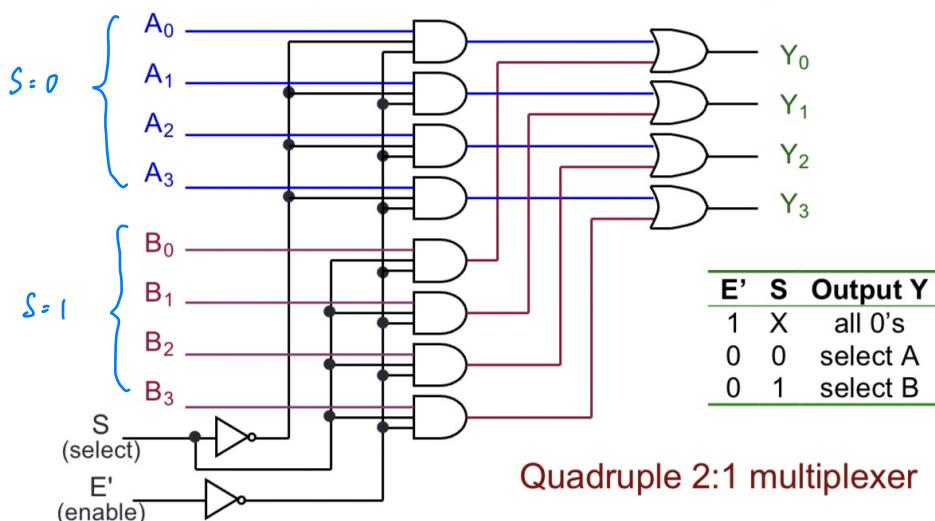
$\text{in minterm notation}$

$$Y = I_0 \cdot m_0 + I_1 \cdot m_1 + I_2 \cdot m_2 + I_3 \cdot m_3$$

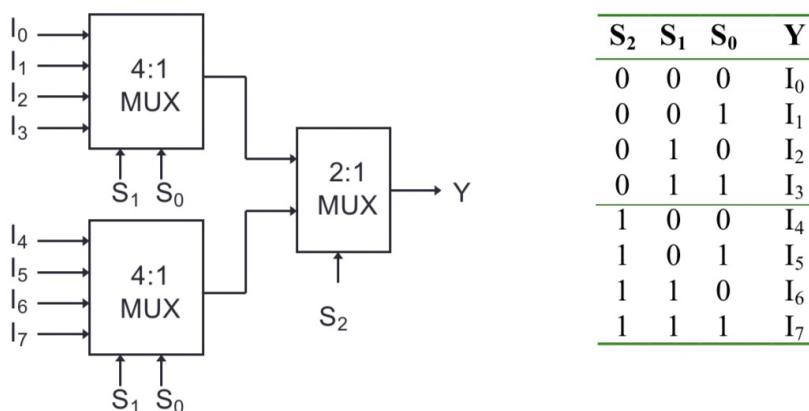
- A 2^n -to-1-line multiplexer, or simply $2^n:1$ MUX, is made from an $n:2^n$ decoder by adding to it 2^n input lines, one to each AND gate.
- A **4:1 multiplexer circuit:**



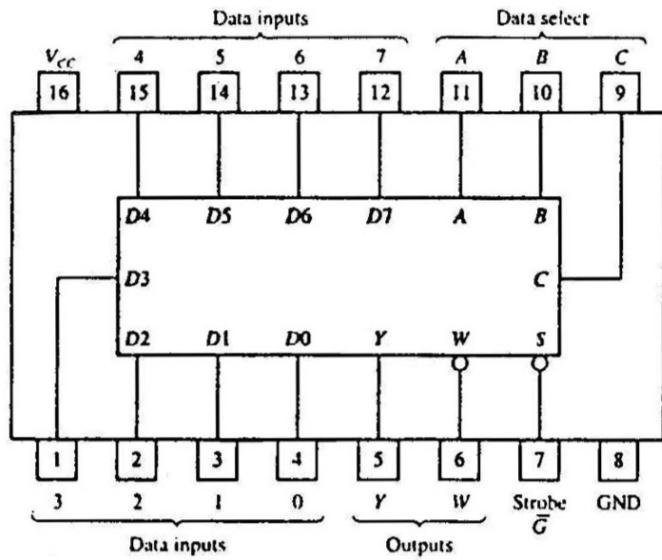
- Some IC packages have a few multiplexers in each package (chip). The selection and enable inputs are common to all multiplexers within the package.



- Larger multiplexers can be constructed from smaller ones.
- An 8-to-1 multiplexer can be constructed from smaller multiplexers like this (note placement of selector lines):



Standard MSI Multiplexer

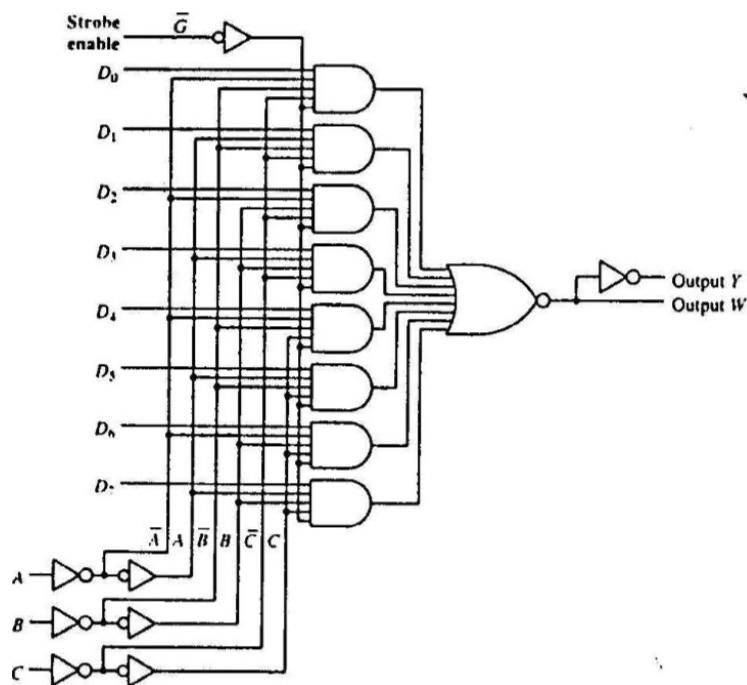


(a)

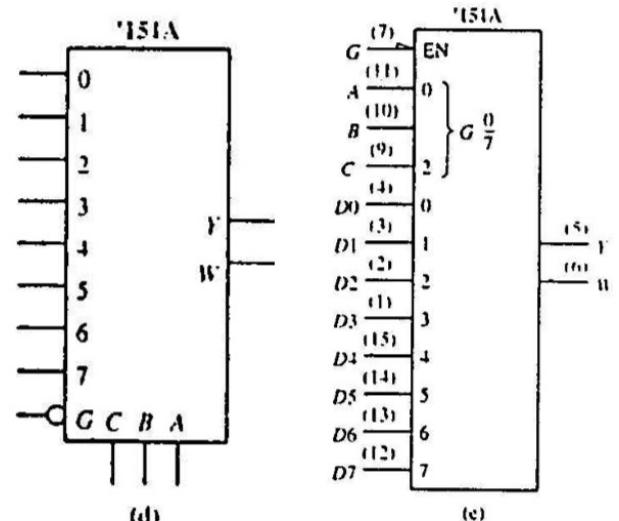
INPUTS			OUTPUTS	
SELECT			STROBE	
C	B	A	\bar{G}	Y W
X	X	X	H	L H
L	L	L	L	D0 \D0
L	L	H	L	D1 \D1
L	H	L	L	D2 \D2
L	H	H	L	D3 \D3
H	L	L	L	D4 \D4
H	L	H	L	D5 \D5
H	H	L	L	D6 \D6
H	H	H	L	D7 \D7

(b)

74151A 8-to-1 multiplexer. (a) Package configuration. (b) Function table.



(c)



(d)

(e)

74151A 8-to-1 multiplexer. (c) Logic diagram. (d) Generic logic symbol. (e) IEEE standard logic symbol.

Boolean functions can be implemented using multiplexers.

- A 2^n -to-1 multiplexer can implement a Boolean function of n input variables:

- Express in sum-of-minterms form

eg. $F(A, B, C) = A'B'C + A'B'C + A'BC + AB'C$
 $= \sum m(1, 3, 5, 6)$

- Connect n variables to the n selection lines

- Put a '1' on a data line if it is a minterm of the function,
 or '0' otherwise.

- $F(A, B, C) = \sum m(1, 3, 5, 6)$

This method works because:

$$\text{Output} = I_0 \cdot m_0 + I_1 \cdot m_1 + I_2 \cdot m_2 + I_3 \cdot m_3 + I_4 \cdot m_4 + I_5 \cdot m_5 + I_6 \cdot m_6 + I_7 \cdot m_7$$

Supplying '1' to I_1, I_3, I_5, I_6 , and '0' to the rest:

$$\text{Output} = m_1 + m_3 + m_5 + m_6$$

- Example: Use a 74151A to implement

$$f(x_1, x_2, x_3) = \sum m(0, 2, 3, 5)$$

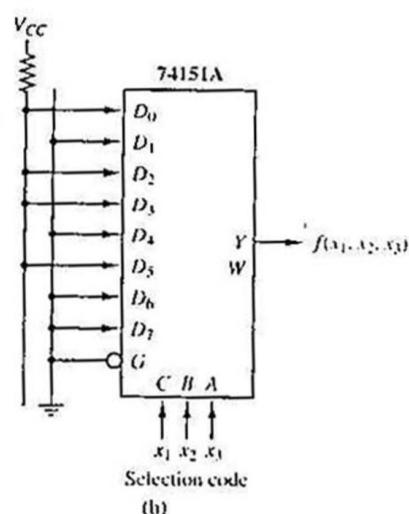
i	C B A			Y
	x_1	x_2	x_3	
0	0	0	0	$f = D_0 = 1$
1	0	0	1	$D_1 = 0$
2	0	1	0	$D_2 = 1$
3	0	1	1	$D_3 = 1$
4	1	0	0	$D_4 = 0$
5	1	0	1	$D_5 = 1$
6	1	1	0	$D_6 = 0$
7	1	1	1	$D_7 = 0$

(a)

Realization of $f(x_1, x_2, x_3) = \sum m(0, 2, 3, 5)$.

(a) Truth table.

(b) Implementation with 74151A.

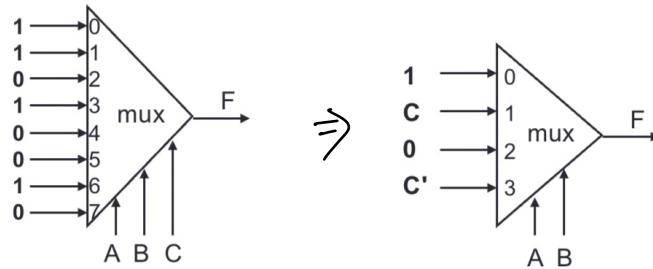


(b)

Using smaller multiplexers

- We can use a smaller single 2^{n-1} to 1 multiplexer to implement a boolean function of n (input) variables.

$$F(A, B, C) = \sum m(0, 1, 3, 6) = A' \cdot B' \cdot C' + A' \cdot B' \cdot C + A' \cdot B \cdot C + A \cdot B \cdot C'$$



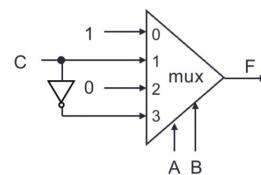
Note: Two of the variables, **A** and **B**, are applied as selection lines of the multiplexer, while the inputs of the multiplexer contain **1**, **C**, **0** and **C'**.

Procedure

- Express Boolean function in sum-of-minterms form.
Example: $F(A, B, C) = \sum m(0, 1, 3, 6)$
- Reserve one variable (in our example, we take the least significant one) for input lines of multiplexer, and use the rest for selection lines.
Example: **C** is for input lines; **A** and **B** for selection lines.
- Draw the truth table for function, by grouping inputs by selection line values, then determine multiplexer inputs by comparing input line (**C**) and function (**F**) for corresponding selection line values.

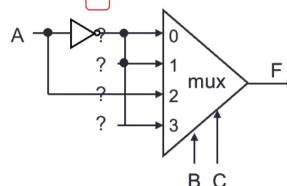
A	B	C	F	MUX input
0	0	0	1	1
0	0	1	1	
0	1	0	0	c
0	1	1	1	
1	0	0	0	0
1	0	1	0	
1	1	0	1	c'
1	1	1	0	

selection lines



Alternative: What if we use **A** for input lines, and **B**, **C** for selector lines?

A	B	C	F	Mux Input
0	0	0	1	1
0	0	1	1	
0	1	0	0	
0	1	1	1	c
1	0	0	0	0
1	0	1	0	
1	1	0	1	c'
1	1	1	0	



A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Annotations:

- A' (when BC = 00)
- A' (when BC = 01)
- A (when BC = 10)
- A' (when BC = 11)