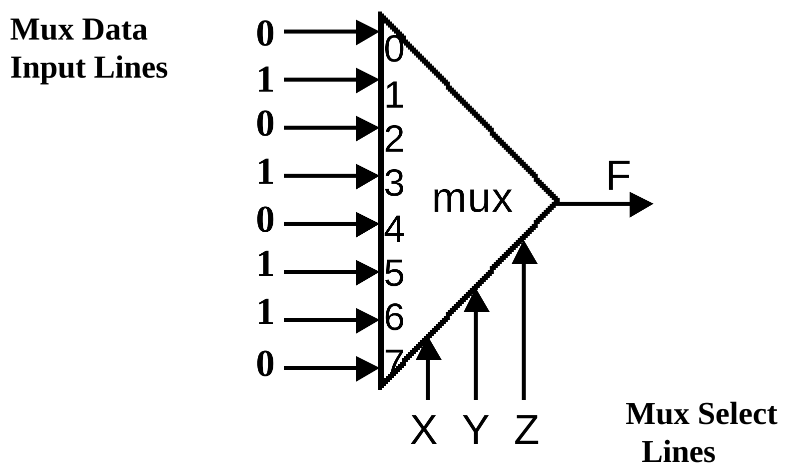
Implementing n-variable Functions Using 2^n-to-1 Multiplexers

- **Any n-variable logic function, in canonical sum-of-minterms form can be implemented using a single 2^n-to-1 multiplexer:**
  - **The n input variables are connected to the mux select lines.**
  - **For each mux data input line I_i (0 £ i £** $2^n - 1$):
    - **Connect 1 to mux input line I_i if i is a minterm of the function.**
    - **Otherwise, connect 0 to mux input line I_i (because i is not a minterm of the function thus the selected input should be 0).**

Example: 3-variable Function Using 8-to-1 mux

- **Implement the function F(X,Y,Z) = S(1,3,5,6) using an 8-to-1 mux.**
  - **Connect the input variables X, Y, Z to mux select lines.**
  - **Mux data input lines 1, 3, 5, 6 that correspond to function minterms are connected to 1.**
  - **The remaining mux data input lines 0, 2, 4, 7 are connected to 0.**



Implementing n-variable Functions Using 2^{n-1}-to-1 Multiplexers

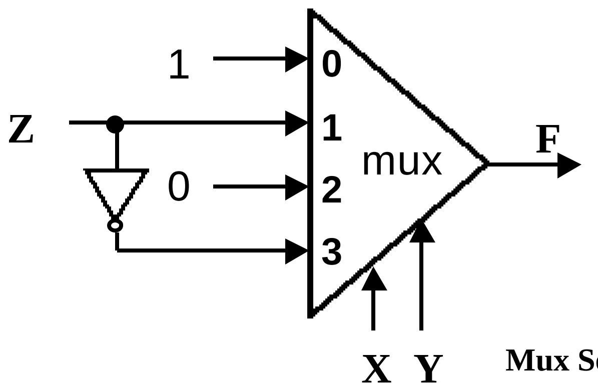
- **Any n-variable logic function can be implemented using a smaller 2^{n-1}-to-1 multiplexer and a single inverter (e.g 4-to-1 mux to implement 3 variable functions) as follows:**
  - **Express function in canonical sum-of-minterms form.**
  - **Choose n-1 variables as inputs to mux select lines.**
  - **Construct the truth table for the function, but grouping inputs by selection line values (i.e select lines as most significant inputs).**
  - **Determine multiplexer input line i values by comparing the remaining input variable and the function F for the corresponding selection lines value i:**
    - **Four possible mux input line i values:**
      - **Connect to 0 if the function is 0 for both values of remaining variable.**
      - **Connect to 1 if the function is 1 for both values of remaining variable.**
      - **Connect to remaining variable if function is equal to the remaining variable.**
      - **Connect to the inverted remaining variable if the function is equal to the remaining variable inverted.**

Example: 3-variable Function Using 4-to-1 mux

- **Implement the function F(X,Y,Z) = S(0,1,3,6) using a single 4-to-1 mux and an inverter.**
  - **We choose the two most significant inputs X, Y as mux select lines.**
  - **Construct truth table:**

Select Lines Value i	X	Y	Z	F	Mux Input i
0	0	0	0	1	1
	0	0	1	1	
1	0	1	0	0	Z
	0	1	1	1	
2	1	0	0	0	0
	1	0	1	0	
3	1	1	0	1	Z’
	1	1	1	0	

(X, Y: Select Lines)



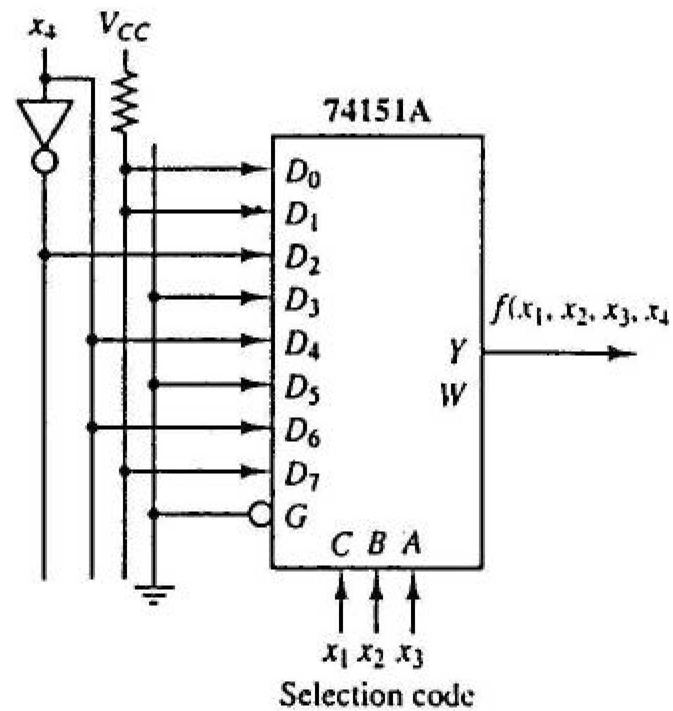
- **We Determine multiplexer input line i values by comparing the remaining input variable Z and the function F for the corresponding selection lines value i:**
  - **when XY=00 the function F=1 (for both Z=0, Z=1) thus mux input0 = 1**
  - **when XY=01 the function F=Z thus mux input1 = Z**
  - **when XY=10 the function F=0 (for both Z=0, Z=1) thus mux input2 = 0**
  - **when XY=11 the function F=Z’ thus mux input3 = Z’**

Example: 4-variable Function Using 8-to-1 mux

- **Implement the function** $F(x_1,x_2,x_3,x_4) =$ å$(0,1,2,3,4,9,13,14,15)$ **using a single 74151A 8-to-1 mux and an inverter.**
  - **We choose the three most significant inputs x_1,x_2,x_3 as mux select lines.**
  - **Construct truth table.**
  - **Determine multiplexer Data input line D_i values.**

	C	B	A				Y
i	X_1	X_2	X_3	X_4	f	f	
0	0	0	0	0	1		
	0	0	0	1	1	1	$D_0 = 1$
1	0	0	1	0	1		
	0	0	1	1	1	1	$D_1 = 1$
2	0	1	0	0	1		
	0	1	0	1	0	$\bar{X}_4$	$D_2 = \bar{X}_4$
3	0	1	1	0	0		
	0	1	1	1	0	0	$D_3 = 0$
4	1	0	0	0	0		
	1	0	0	1	1	X_4	$D_4 = X_4$
5	1	0	1	0	0		
	1	0	1	1	0	0	$D_5 = 0$
6	1	1	0	0	0		
	1	1	0	1	1	X_4	$D_6 = X_4$
7	1	1	1	0	1		
	1	1	1	1	1	1	$D_7 = 1$

(a)



(b)