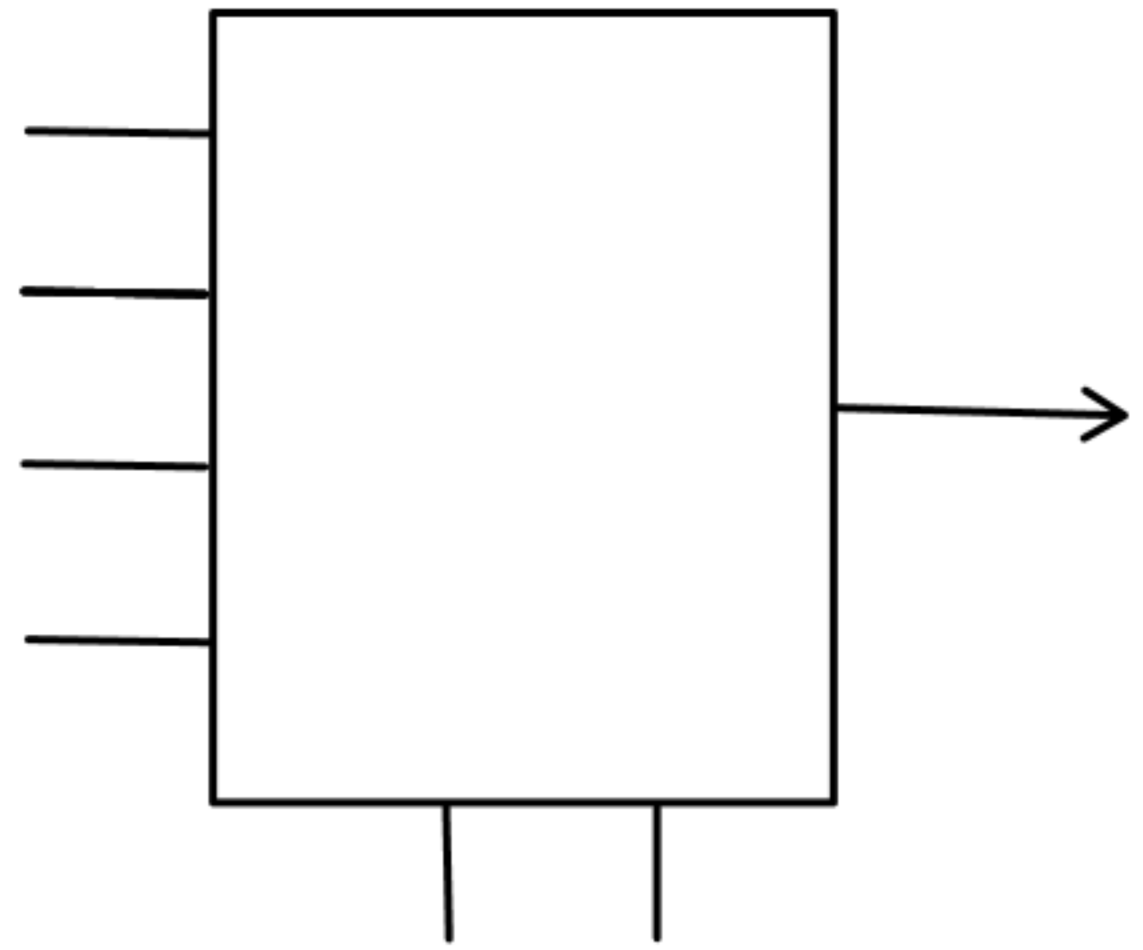
IMPLEMENTATION OF BOOLEAN FUNCTIONS USING MULTIPLEXER



IMPLEMENTATION OF BOOLEAN FUNCTIONS USING MULTIPLEXER

- Set the Number of variables (n) = Number of select lines
- All variables are connected to the select lines

- The desired output is connected to the input line in the same order of MSB to LSB

Example: Implement $f(A,B) = \sum(1,3)$ using MUX

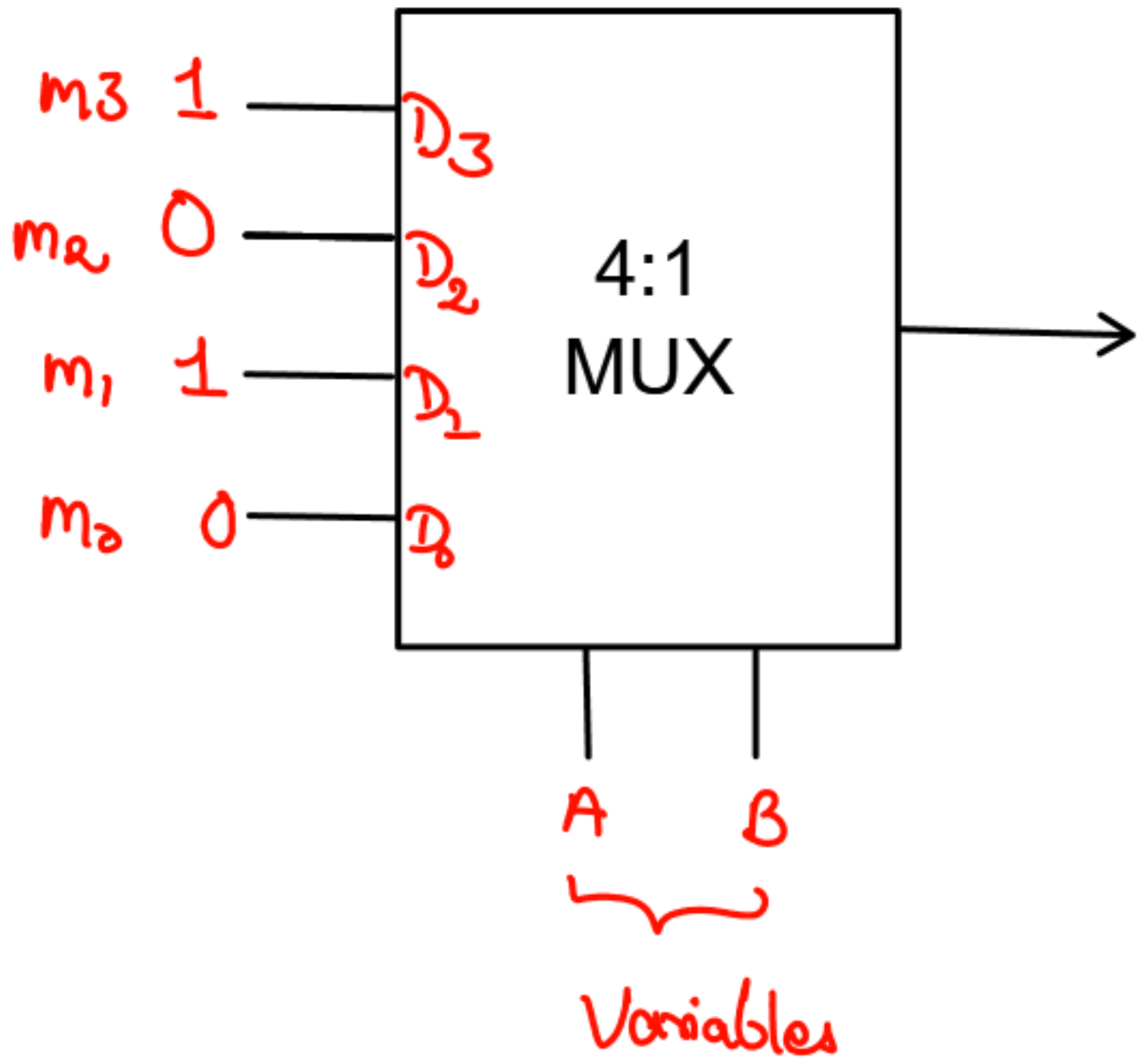
MSB LSB

	A	B	Y
m_0	0	0	0
m_1	0	1	1
m_2	1	0	0
m_3	1	1	1

method 1 :

↓

less effective



IMPLEMENTING FUNCTION KEEPING N-1 VARIABLES AS SELECT LINE MUX

- STEP 1 - Draw the truth table for given number of variable function
- STEP 2 - Consider one variable as input and another variables as select lines
  - From the Implementation table where select lines of mux are columns and one input variable and its components are rows.
- STEP 3 - Find AND between rows on the basis of the truth table.
- STEP 4 - So the outcome are considered as input lines for MUX.

Example: Implement $f(A,B,C) = \sum(0,1,4,6,7)$ using 4:1 MUX

4:1 MUX → 2 Select lines

TRUTH TABLE

	A	B	C	Y
0	0	0	0	1
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

let A&B are the Select line and c is Input Variable

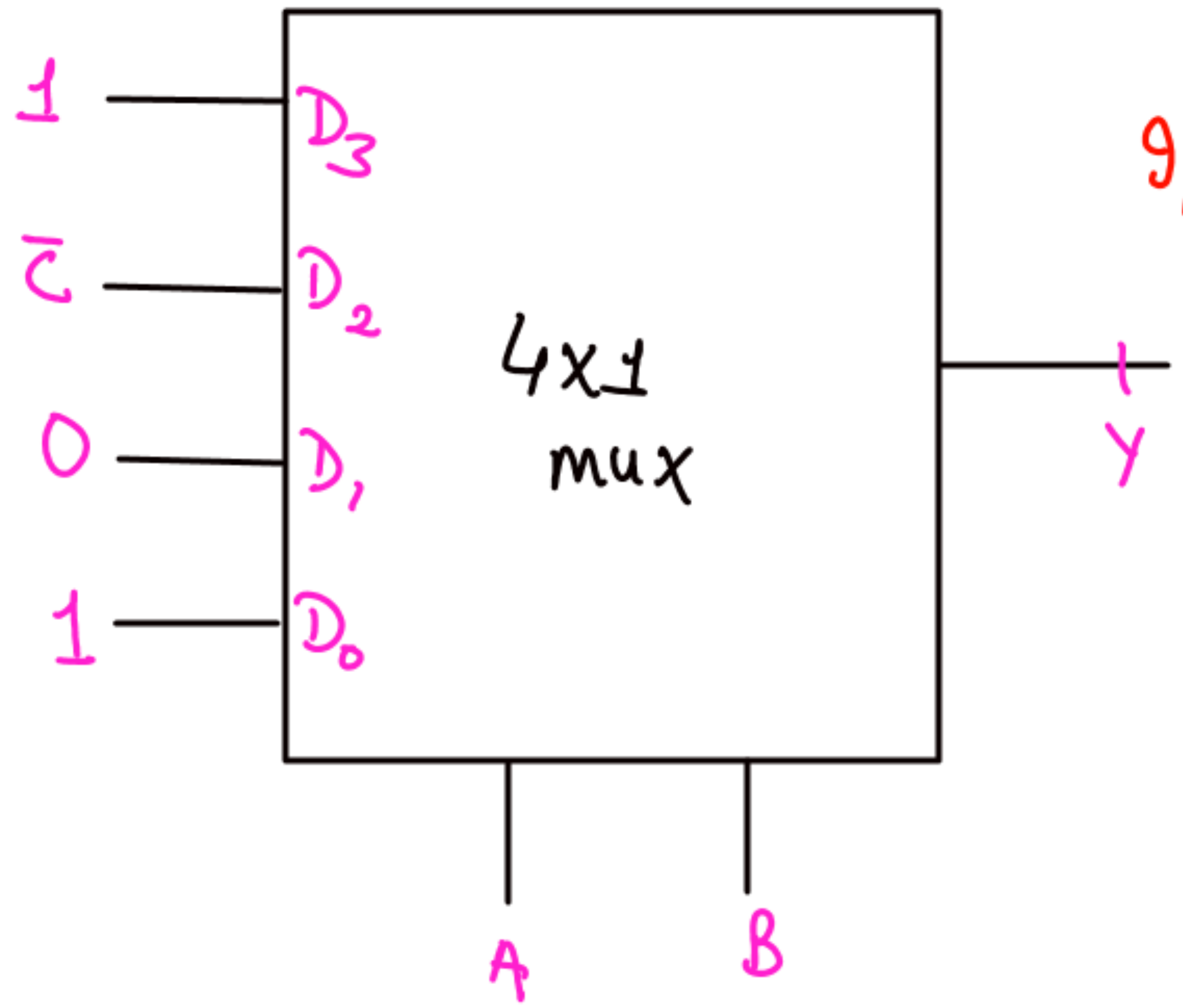
* Maintain the sequence of input Variable with Select line

Implementation Table

not Gray Code

Decimal

C ↓ / AB →	00	01	10	11
$\bar{C}$ 0	0 (ABC 000)	2 (ABC 010)	4 (ABC 100)	6 (ABC 110)
C 1	1 (ABC 001)	3 (ABC 011)	5 (ABC 101)	7 (ABC 111)
Input →	1	0	$\bar{C}$	1
	D_0	D_1	D_2	D_3



* Implementation table is NOT a K-map.

taking A as input line and BC as a select line.

$$f(ABC) = \sum m(0,1,4,6,7)$$

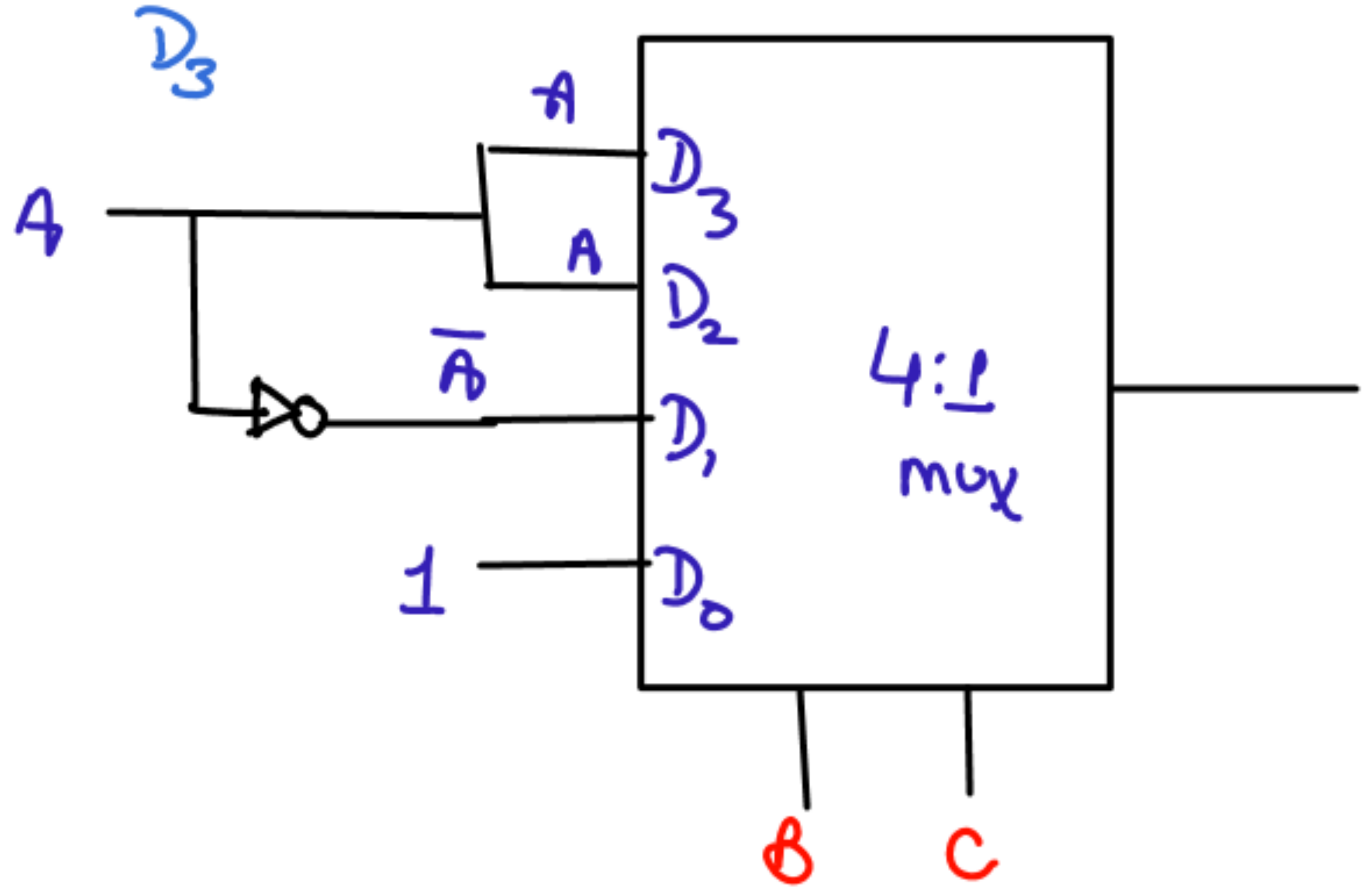
SL → BC→

I/P → A↓

A \ BC	00	01	10	11
$\bar{A}$ 0	0 (000)	1 (001)	2 (010)	3 (011)
A 1	4 (100)	5 (101)	6 (110)	7 (111)
	1 D_0	$\bar{A}$ D_1	A D_2	A D_3

Straight Binary Code

B	C	Y	
0	0	1	D_0
0	1	$\bar{A}$	D_1
1	0	A	D_2
1	1	A	D_3



$f(ABC) = \sum m(0,1,4,6,7)$

$\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + A\underbrace{B\bar{C}} + AB\underbrace{C}$ ← SOP form

m_0 m_4

$\bar{B}\bar{C}(A+\bar{A})$ $\bar{A}\cdot\bar{B}C$ $AB\bar{C}$ $A\cdot BC$

$\bar{B}\bar{C}\cdot 1$ $\bar{A}$ A A

$\Rightarrow 1$ $\bar{A}$ A A

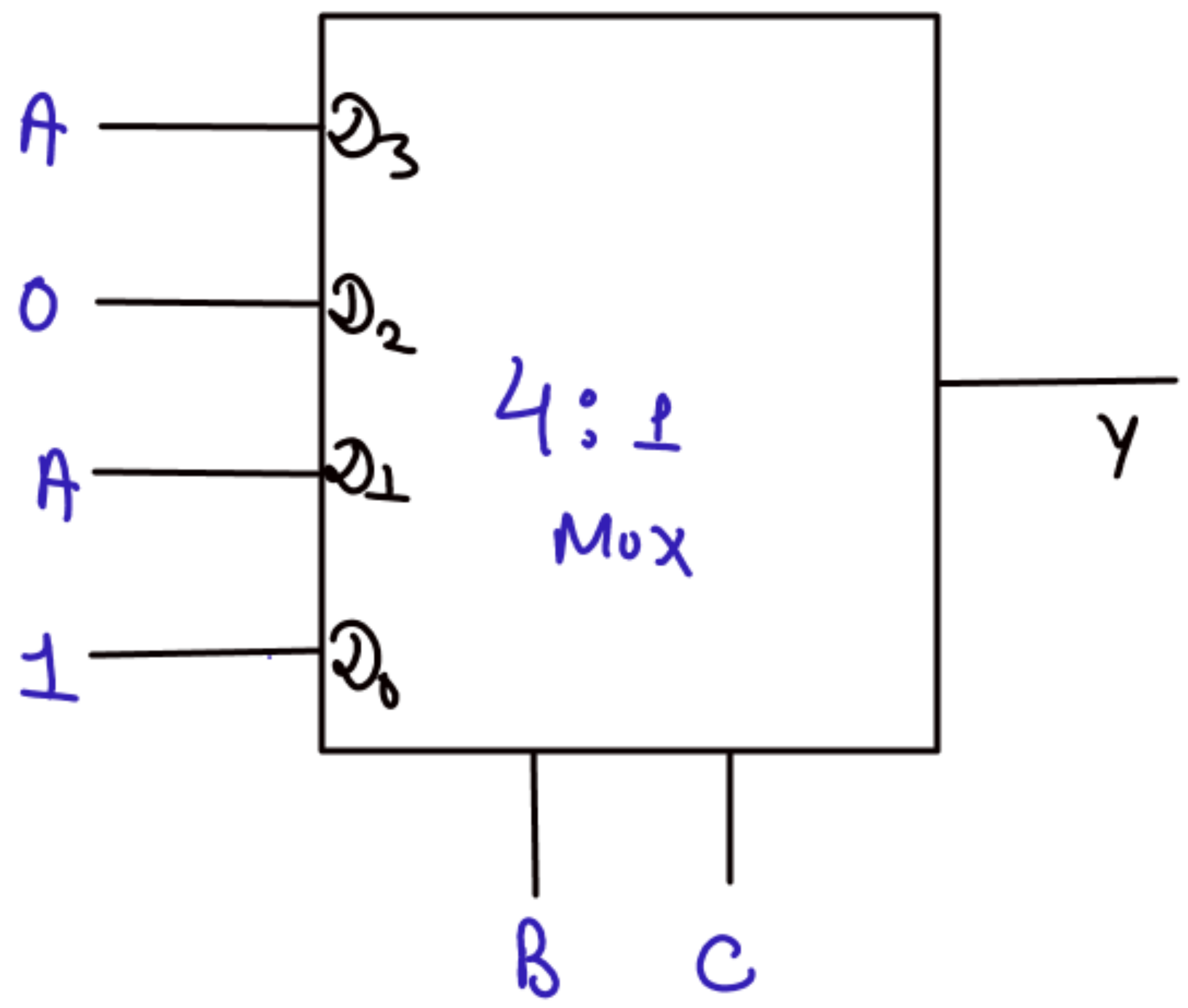
D_0 D_1 D_2 D_3

STEPS FOR IMPLEMENTING FUNCTION USING MUX

Implement $f(A,B,C) = \sum(0,1,4,7)$ using 4:1 MUX

Let A be the input line, B&C be the select line

BC→		00	01	10	11
$\bar{A}$	0	0 (000)	1 (001)	2 (010)	3 (011)
A	1	4 (100)	5 (101)	6 (110)	7 (111)
		1	$\bar{A}$	0	A
		D_0	D_1	D_2	D_3

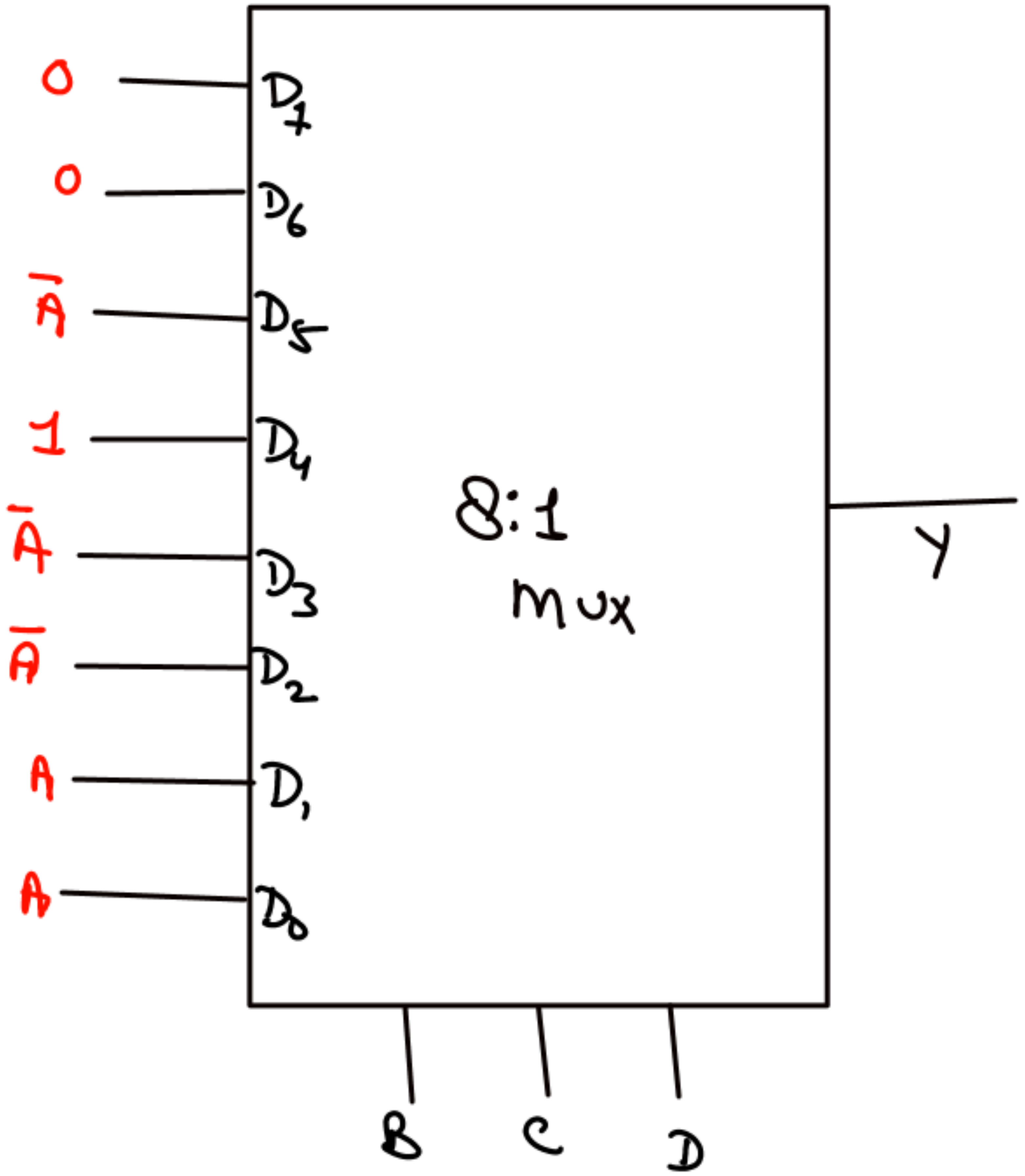


STEPS FOR IMPLEMENTING FUNCTION USING MUX

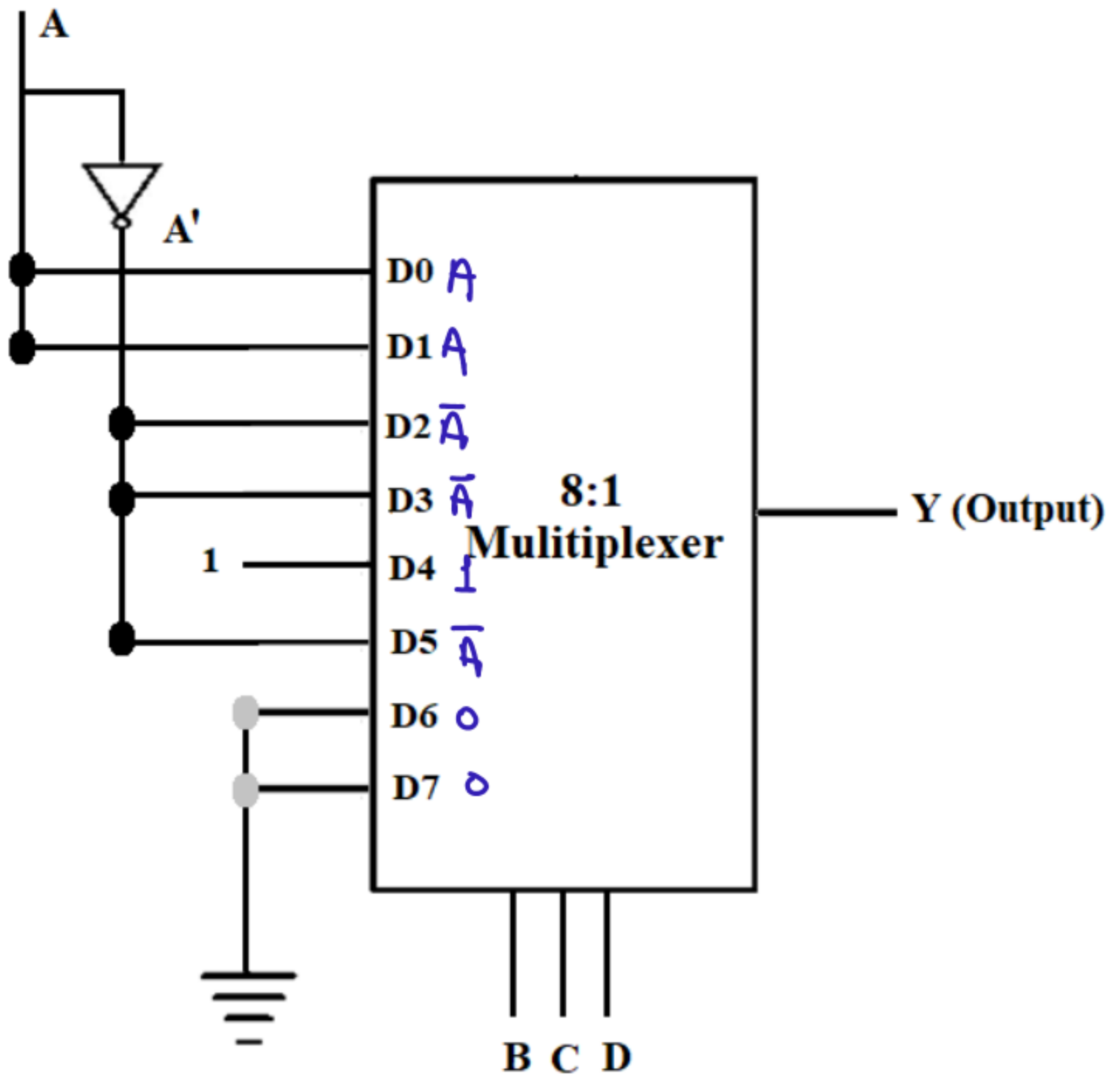
Implement $f(ABCD)=\sum m(2,3,4,5,8,9,12)$ using 8:1 MUX

Let A be the Input line & BCD be the Select line

BCD→	000	001	010	011	100	101	110	111
$\bar{A}$ 0	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111
A 1	8 1000	9 1001	10 1010	11 1011	12 1100	13 1101	14 1110	15 1111
	A	A	$\bar{A}$	$\bar{A}$	1	$\bar{A}$	0	0
	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7



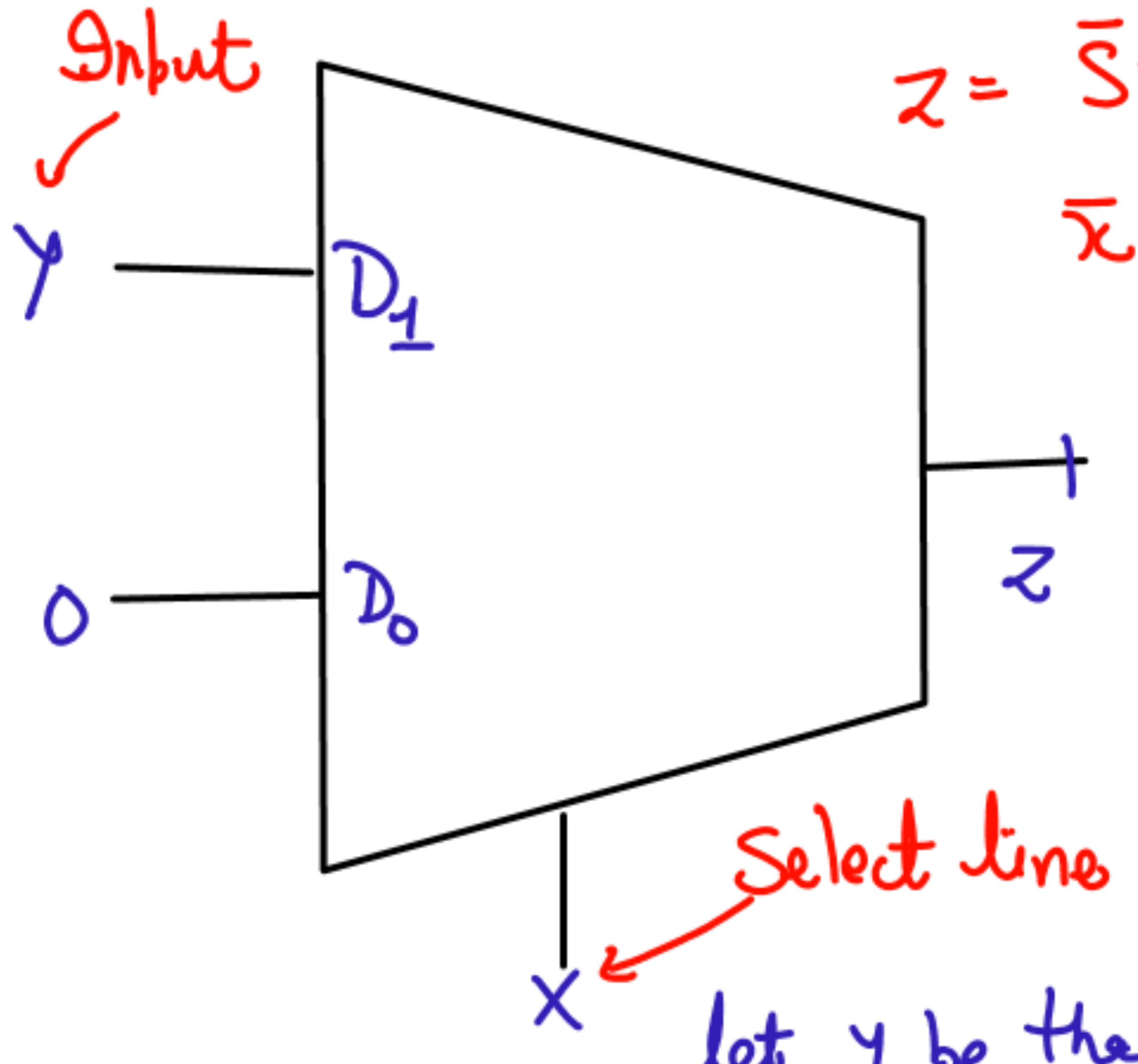
STEPS FOR IMPLEMENTING FUNCTION USING MUX



IMPLEMENTATION OF LOGIC GATES USING MUX

The multiplexer is a universal logic. So we can perform any logical function using the multiplexer

AND Gate using 2:1 Multiplexer



$Z = \bar{S}D_0 + SD_1$

$\bar{x}\cdot 0 + xY$

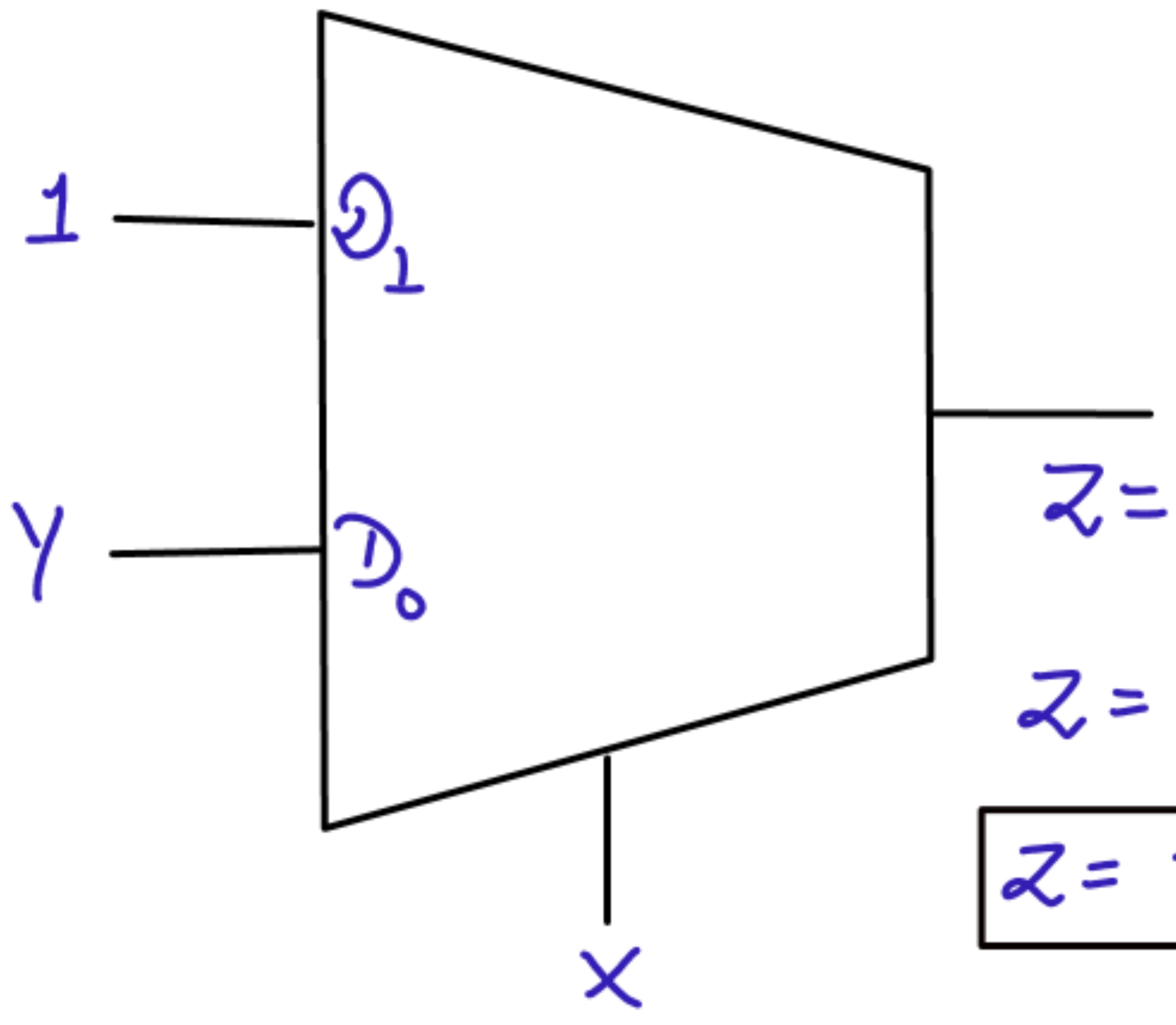
$= xy$

let y be the Input Variable

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1 → m_3

	X = 0	X = 1
$\bar{Y}$ 0	0 (00)	2 (10)
Y 1	1 (01)	3 (11)
	0	Y
	D_0	D_1

OR Gate using 2:1 Multiplexer



$\bar{S}D_0 + SD_1$

$Z = \bar{x}y + x1$

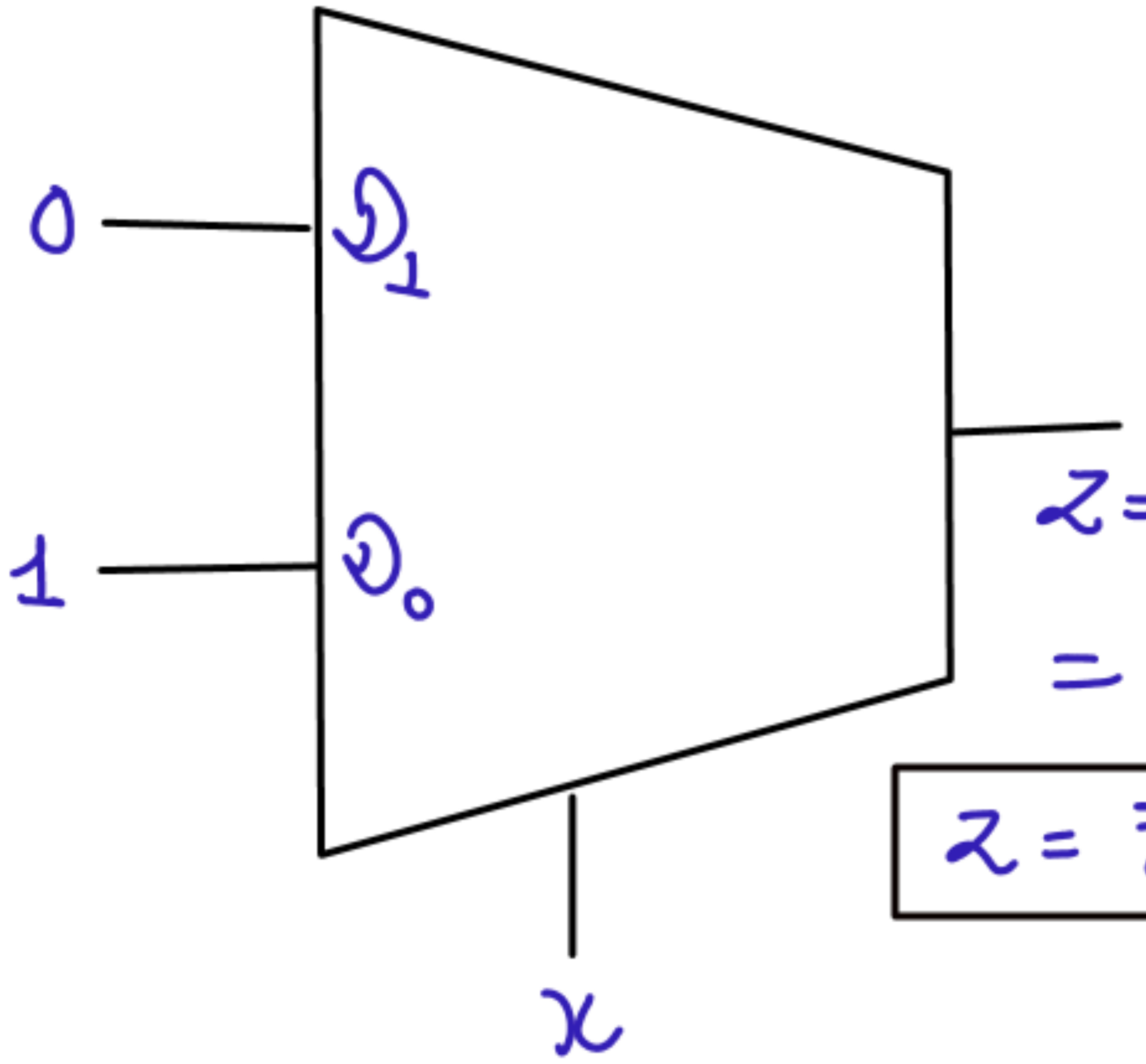
$Z = x + \bar{x}y$

Redundant Literal Rule

$\boxed{Z = x+y}$

IMPLEMENTATION OF LOGIC GATES USING MUX

NOT Gate using 2:1 Multiplexer

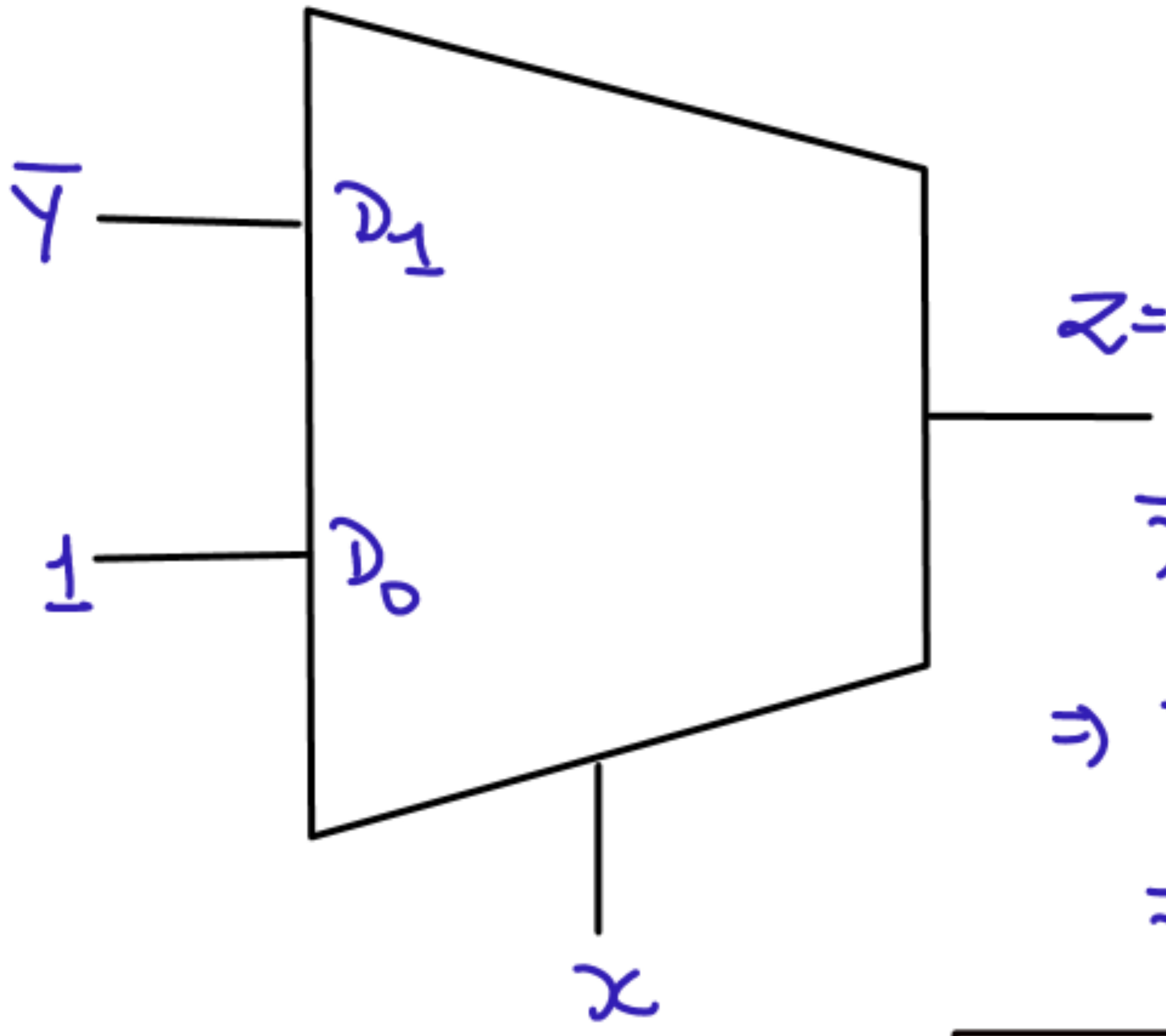


$z = \bar{x} D_0 + x D_1$

$= \bar{x} \cdot 1 + \cancel{x\,0}$

$\boxed{z = \bar{x}}$

NAND Gate using 2:1 Multiplexer



$z = \bar{x} D_0 + x D_1$

$\bar{x} \cdot 1 + x \cdot \bar{y}$

$\Rightarrow \bar{x} + x\bar{y}$ RLR

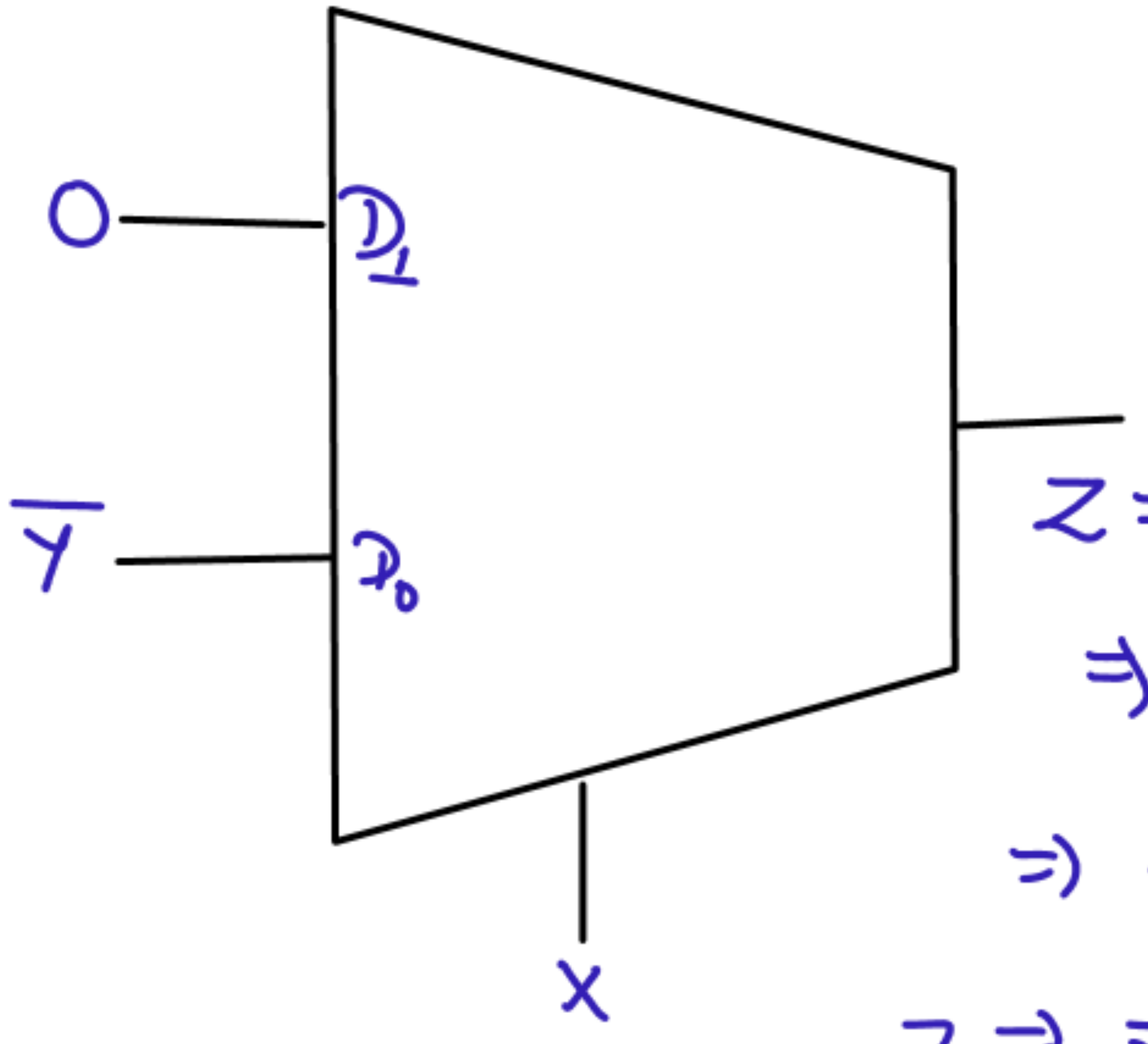
$\bar{x} + \bar{y}$

$\boxed{z = \overline{x \cdot y}}$

IMPLEMENTATION OF LOGIC GATES USING MUX

$\overline{x+y} = \bar{x}\cdot\bar{y}$

NOR Gate using 2:1 Multiplexer



$Z = \bar{x}D_0 + xD_1$

$\Rightarrow \bar{x}I_0 + \cancel{x\cdot 0}$

$\Rightarrow \bar{x}\cdot I_0$

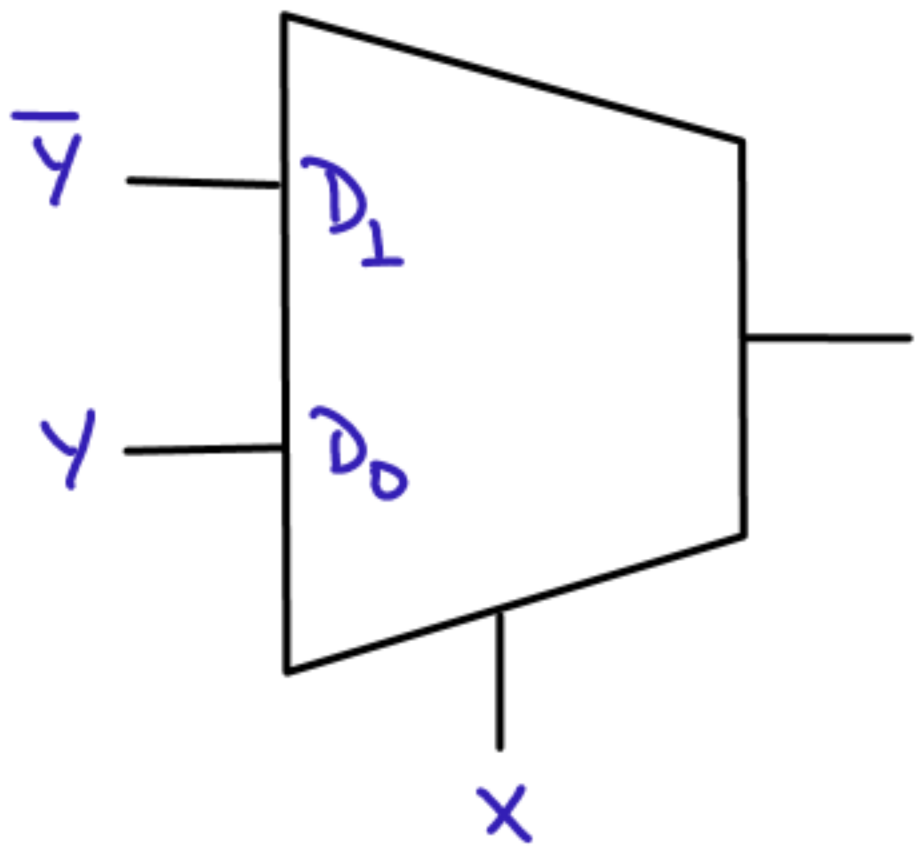
$Z \Rightarrow \bar{x}\cdot\bar{y}$

$\boxed{Z = x+y}$

$\bar{x}y + x\bar{y}$

$\bar{S}D_0 + SD_1$

XOR Gate using 2:1 Multiplexer

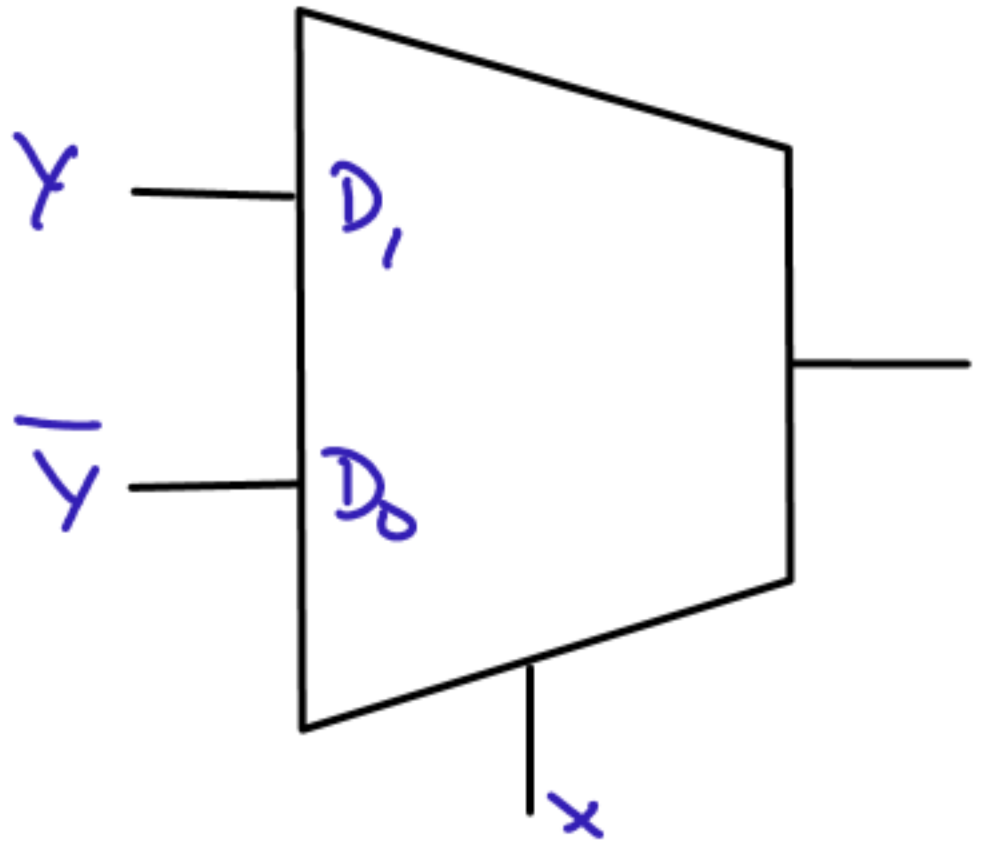


$\boxed{Z = x\bar{Y} + \bar{x}Y}$

$= \bar{x}y + x\bar{y}$

$\bar{x}\bar{y} + xy$

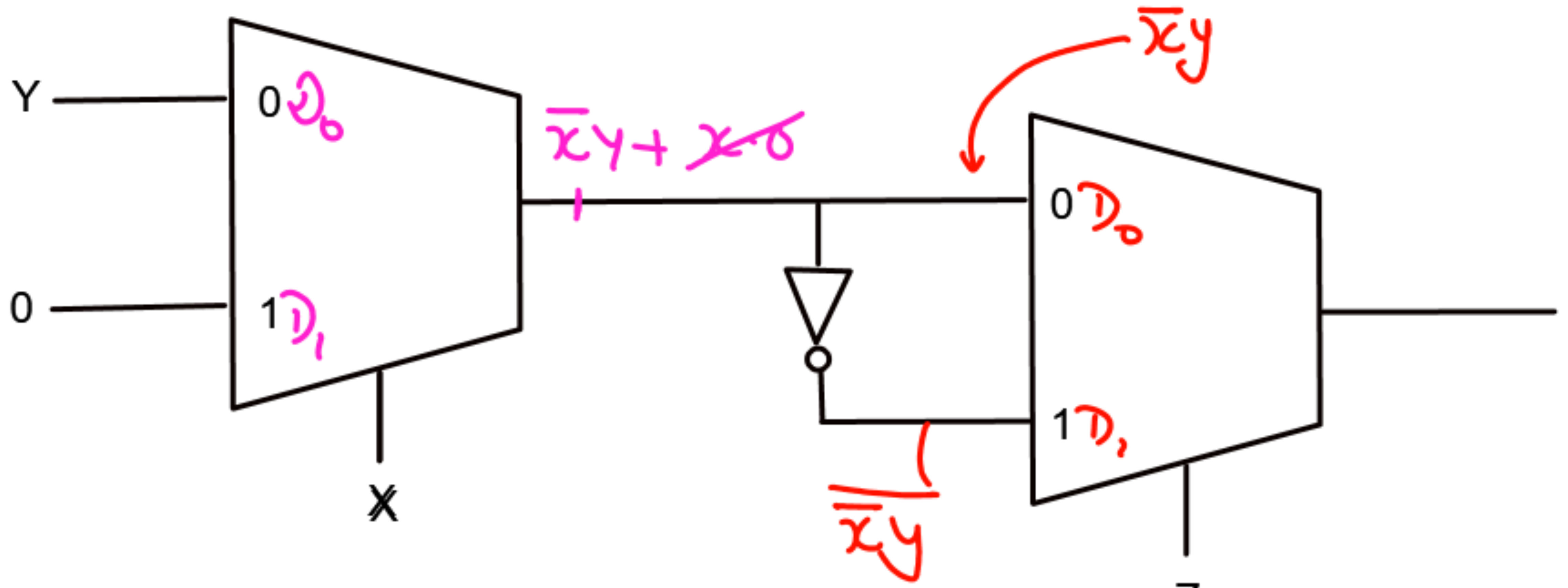
XNOR Gate using 2:1 Multiplexer



$Z = \bar{x}\cdot D_0 + x\cdot D_1$

$= \bar{x}\cdot\bar{y} + x\cdot y$

Consider the circuit shown in the figure.



$\Rightarrow \bar{\bar{x}} + \bar{y} = x + \bar{y}$

$F = \bar{z}\cdot D_0 + z\cdot D_1$

$F = \bar{z}\cdot\bar{x}y + z\cdot(x+\bar{y})$

$= \bar{z}\bar{x}y + zx + z\bar{y}$

$\Rightarrow \bar{x}y\bar{z} + xz + \bar{y}z$

The Boolean expression F implemented by the circuit is

- A $\overline{X}\,\overline{Y}\,\overline{Z} + XY + \overline{Y}Z$
- B $\overline{X}Y\overline{Z} + XZ + \overline{Y}Z$ ✓
- C $\overline{X}Y\overline{Z} + XY + \overline{Y}Z$
- D $\overline{X}\,\overline{Y}\,\overline{Z} + XZ + \overline{Y}Z$

HALF ADDER USING 4:1 MUX

A	B	Sum (S)	Carry (C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

HALF ADDER USING 2:1 MUX

A	B	Sum (S)	Carry (C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

HALF SUBTRACTOR USING 2:1 MUX

A	B	Difference (D)	Borrow (B)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0