**MULTIPLEXER**

It is a combinational circuit that selects binary information from one of many inputs and directs that to the single output.

2x1 MUX (multiplexer)

I want only one of the inputs to be at the output.

A picture containing hanger

Description automatically generated A picture containing whiteboard

Description automatically generated

If S is 1, output is I0 ; if S is 0, output is I1.

4x1 MUX

Diagram

Description automatically generatedBackground pattern

Description automatically generated with low confidenceDiagram

Description automatically generated

In the truth table, we only write control inputs. You don’t need to write the different variations of the inputs.

By using a decoder, we can combine any kind of canonical circuits bc each output of a decoder includes minterm.

By using a MUX, we can also design a circuit.

*🡪 : number of data input , n: control input*

Using MUX for Digital Design

We have n variables for the equation.

Use n-1 variable for the control input and the remaining variable for the output.

Example:

F = (1, 2, 6, 7) -----> how many variables do we need : 3

F(x, y, z) = (1, 2, 6, 7)

Diagram, schematic

Description automatically generatedI will use 2 variables (x and y) for the control inputs, and 1 variable (z) for the data input.

very similar to z so we can say it is z.   
If control inputs are 0 0, output is going to be z



0

z’

1



|  |  |  |  |
| --- | --- | --- | --- |
| x | y | z | F |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Blue : 0 0 / 0 0 case ( / diğer satıra geç demek)

Green : 0 1 / 0 1 case

Red : 1 0 / 1 0 case

Orange : 1 1 / 1 1 case

Calendar

Description automatically generated with medium confidence

Example:

F(A, B, C, D) = (1, 3, 4, 11, 12, 13, 14, 15)

Diagram

Description automatically generatedWe have 4 variables and 4 inputs. We need 8x1 MUX bc we are gonna use 3 of the inputs as a control inputs and 1 as a data input. That means we can represent = 8 different outputs. So we will need 8x1 MUX.

D

1

D

D’

0

0

D

1



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A (s2) | B (s1) | C (s0) | D | F |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Table

Description automatically generated

We use same MUX everytime, but inputs are changing according to function.