

Computer System Architecture
COMP201Th
Lecture: 8
Multiplexer and De-multiplexer

Combinational logic has many uses in electronic systems. It is used to carry out the essential arithmetic in computers, calculators and many other types of automatic machinery. Mathematics and logical decision making also uses combinational logic in the form of comparators. These circuits decide whether one value is the same, larger or smaller than another value. After making a logical decision, data may need to be routed to different parts of the electronic system, this routing is controlled by more combinational logic circuits such as multiplexer and demultiplexers.

In this lecture, we will discuss about one such combinational logic circuit called multiplexer.

- Multiplexer:
 - (Many to one)
 - It is a combinational logic circuit that selects binary information from one of the multiple input lines and directs it to an output line according to a received select code.
 - Multiplexer is also called data selector.

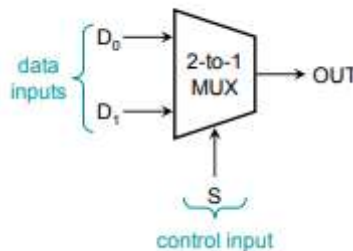
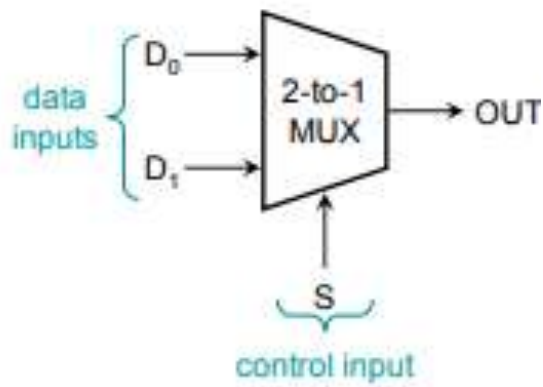
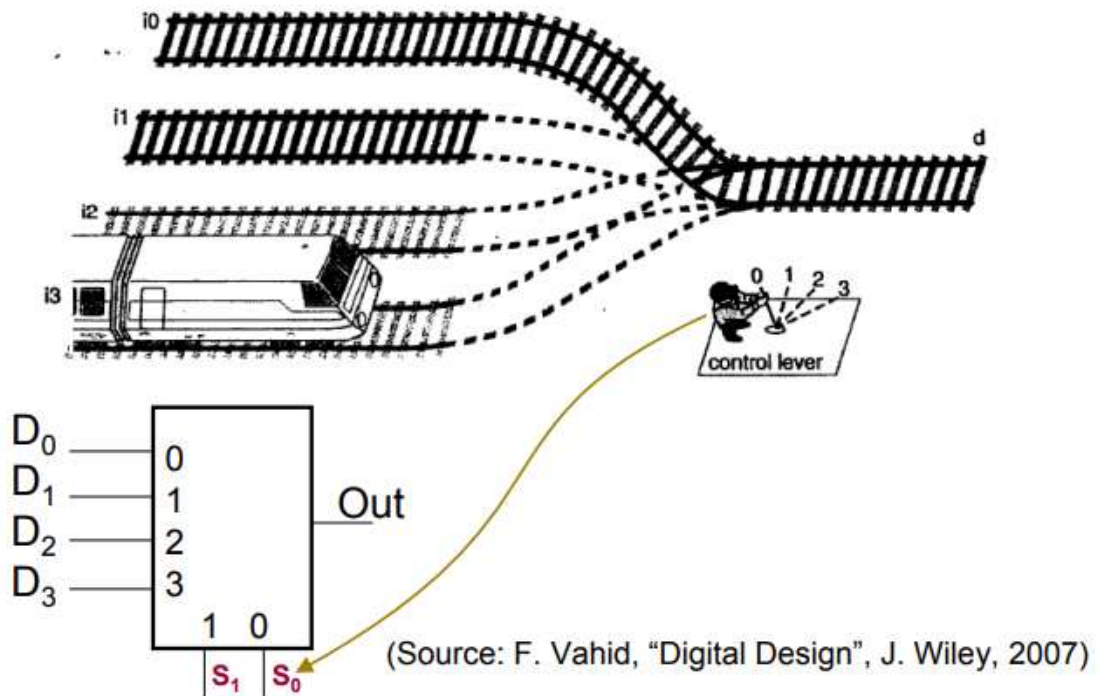


Fig: 2-1 multiplexer

- Multiplexer contains the following:
 - 2^n data inputs
 - n selection inputs
 - a single output
 - selection input determines the input that should be connected to the output.



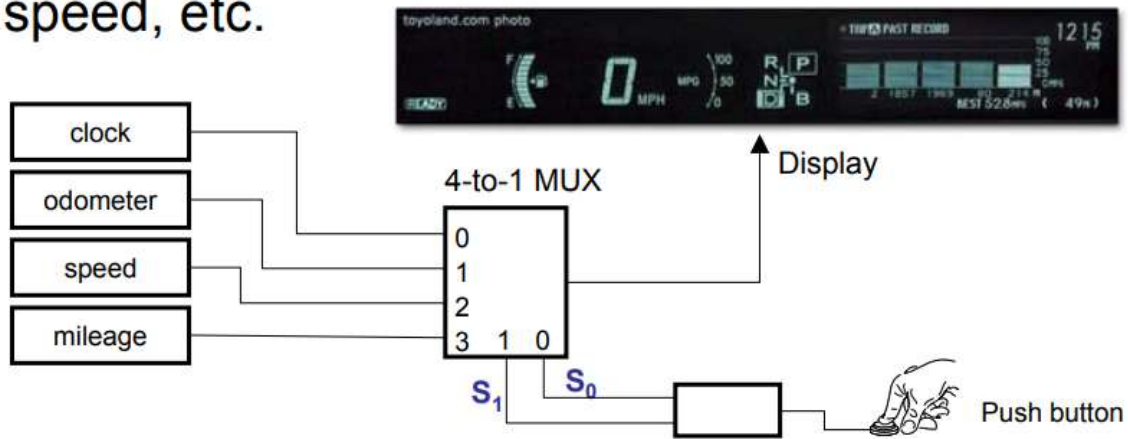
S	OUT
0	D ₀
1	D ₁



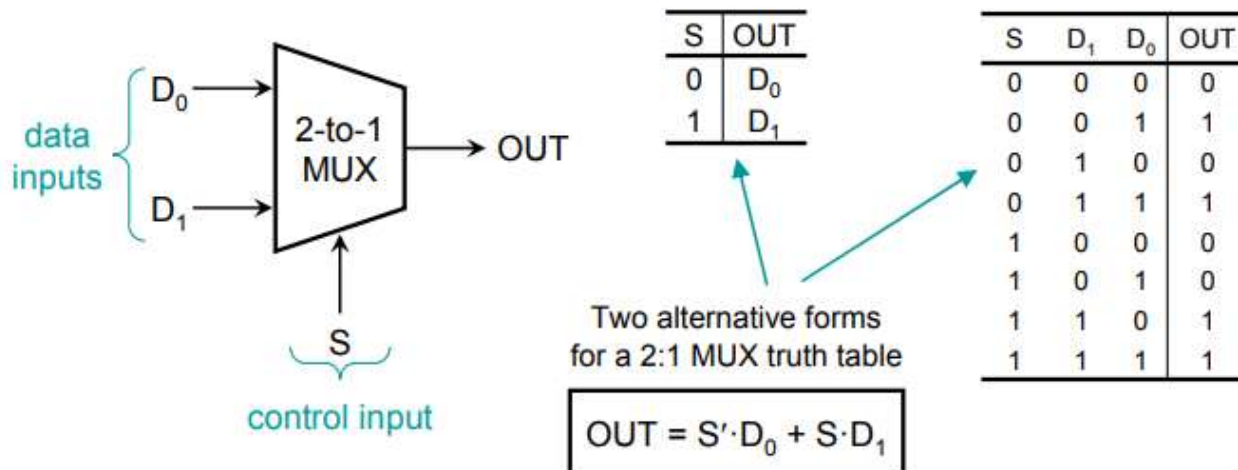
An analogy for Multiplexer is a railway junction. Say there is a junction where 4 railway lines are converging on to a single railway line i.e. rails from six different cities come to the junction and will go to same destination now how these 4 rail lines will be connected by the same one railway line is controlled through control lever(selection lines of mux).

An example of use of MUX is trip controller in cars to display average, time, distance etc.

- Trip controller in a car to display mileage, time, speed, etc.



- A 2-to-1 MUX:



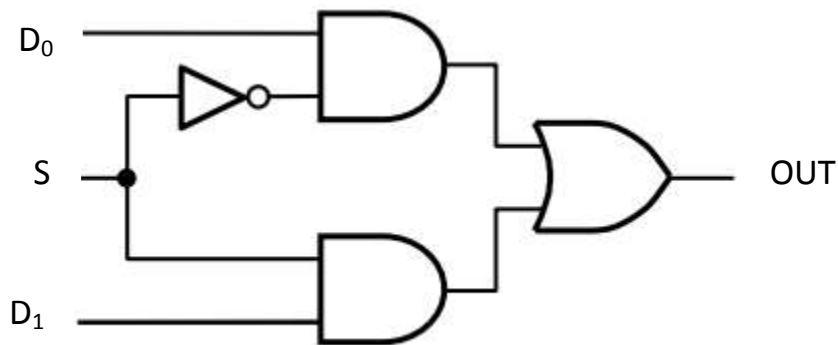
- Fig above indicates the logic symbol and truth table of 2 to 1 multiplexer.
- In this multiplexer:
 - D₀ and D₁ are the data inputs
 - S is the select
 - And OUT is the output of the multiplexer.
- From the truth table, when S is 0 then the data D₀ appears at output OUT and when S =1, the output OUT receives the data D₁.
- In the alternate form of truth table, for S=0; we have 4 possible combinations for the inputs D0 and D1 i.e. 00,01,10,11. Similar is the case for S=1.

- Now, K map for this 2-to-1 multiplexer is of three variables as shown below:

		$D_1 D_0$			
		00	01	11	10
S	0		1	1	
	1			1	1

K-map for 2-to-1 MUX

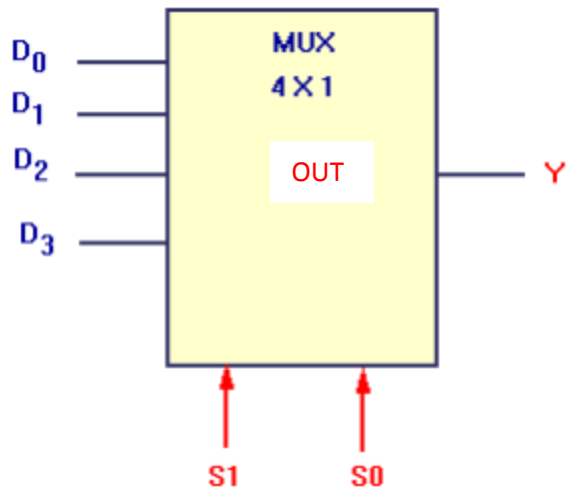
- After simplifying the K-map: $OUT = S'D_0 + SD_1$
- The 2 to 1 multiplexer can be implemented using two AND gates and one OR gate. The NOT gate provides complemented and un-complemented form of S i.e. select line.



Implementation of 2 to 1 multiplexer

- **4 to 1 Multiplexer:**

- In this, there are 4 data inputs namely D_0, D_1, D_2, D_3 and single output OUT .
- For 4 data inputs, there are two select lines namely S_0 and S_1 . The 2 select input provide 4 input combinations for selecting the proper data input at the output OUT .
- The output OUT receives D_0 only when $S_1=0$ and $S_0=0$.
- Similarly, OUT receives D_1 only when $S_1S_0=01$, OUT receives D_2 only when $S_1S_0=10$ and OUT receives only D_3 when $S_1S_0=11$

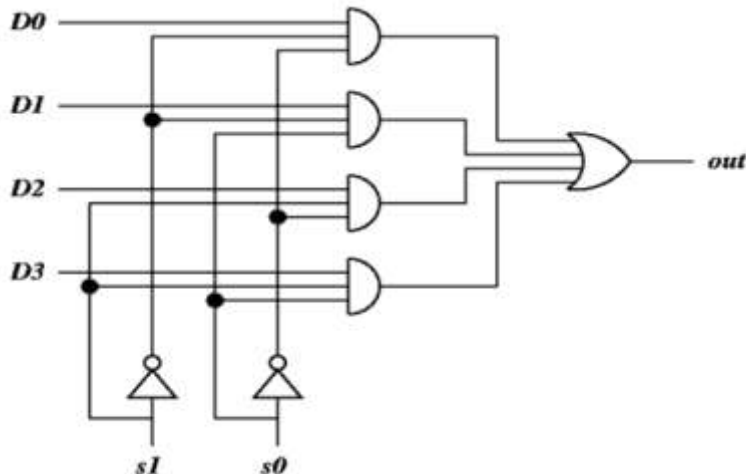


■ Expression for OUT:

$$\text{OUT} = \underbrace{S_1' \cdot S_0'}_{M_0} \cdot D_0 + \underbrace{S_1' \cdot S_0}_{M_1} \cdot D_1 + \underbrace{S_1 \cdot S_0'}_{M_2} \cdot D_2 + \underbrace{S_1 \cdot S_0}_{M_3} \cdot D_3$$

S ₁	S ₀	OUT
0	0	D ₀
0	1	D ₁
1	0	D ₂
1	1	D ₃

- From the Boolean expression we can draw the logic circuit consisting of an OR gate with 4 inputs. Each product term is represented by three input AND gate. One of the input of the AND gate is the respective data input. The Select lines S1 and S0 along with inverter provide select input either in un-complemented or complemented form.



Implementation of 4 to 1 MUX