Demultiplexer

Silicon Community - Session 1

Demultiplexers - Introduction (1)

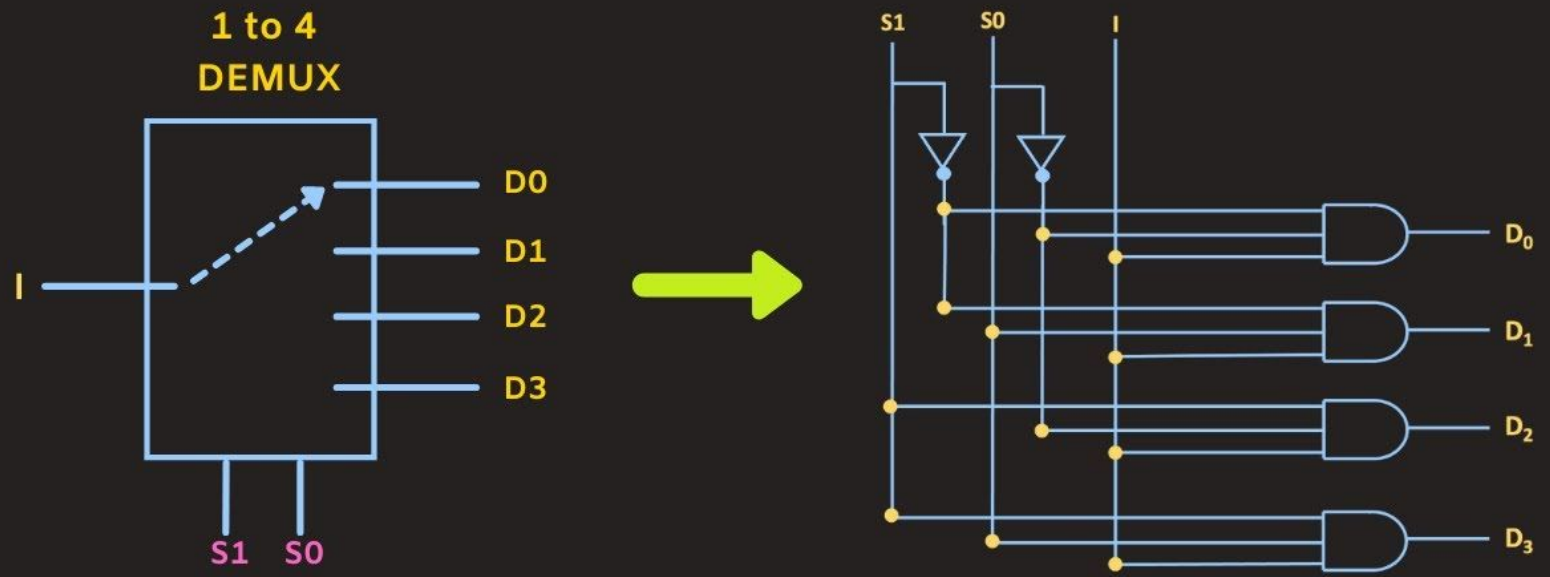
In digital logic and design, a demultiplexer (often abbreviated as DEMUX) is a combinational logic circuit that takes a single input and distributes it to one of multiple outputs based on the control signals. It performs the reverse operation of a multiplexer (MUX).

A demultiplexer has one input line, multiple output lines, and control inputs that determine which output line the input will be routed to. The number of output lines in a demultiplexer is determined by the number of control inputs (or selectors), often represented as 2^n, where n is the number of control inputs.

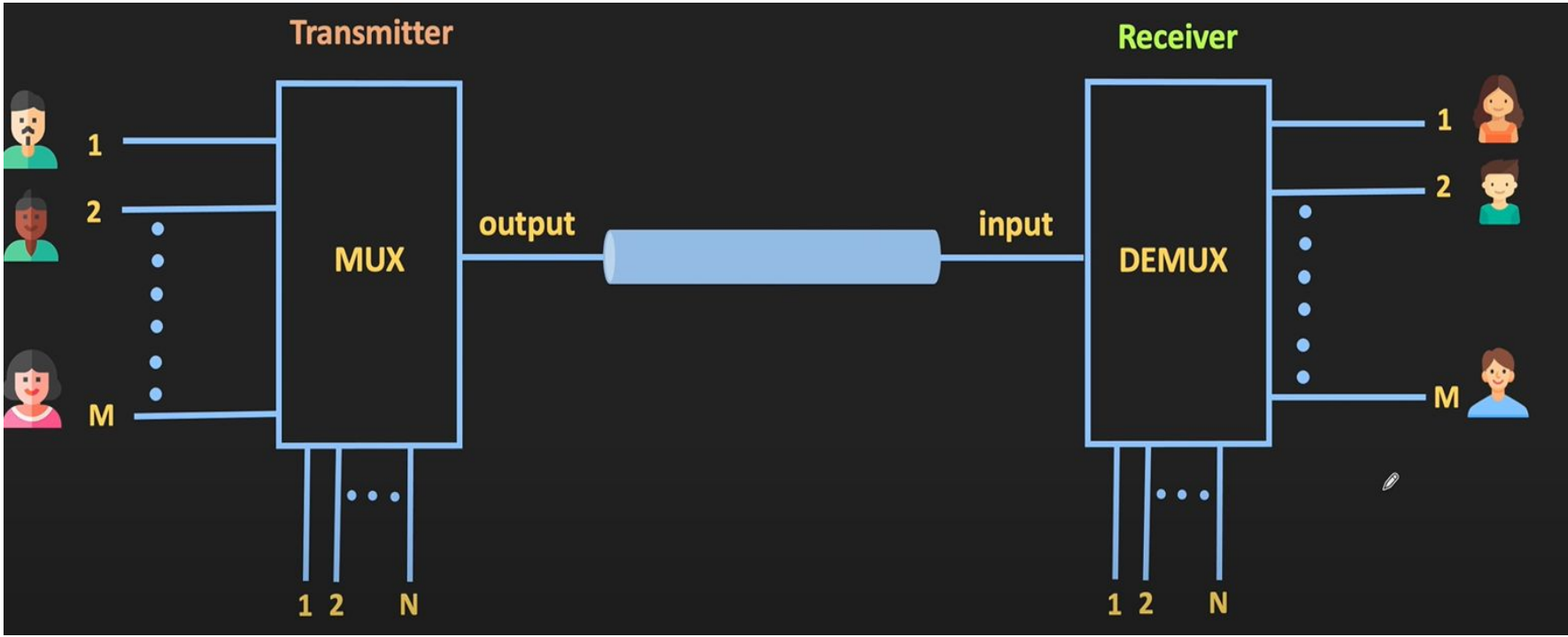
Demultiplexers - Introduction (2)

The main purpose of a demultiplexer is to take a single input signal and distribute it to a specific output line based on the control inputs. Each control input combination corresponds to a specific output line, and the input signal is forwarded only to the selected output line while being disabled on the remaining output lines.

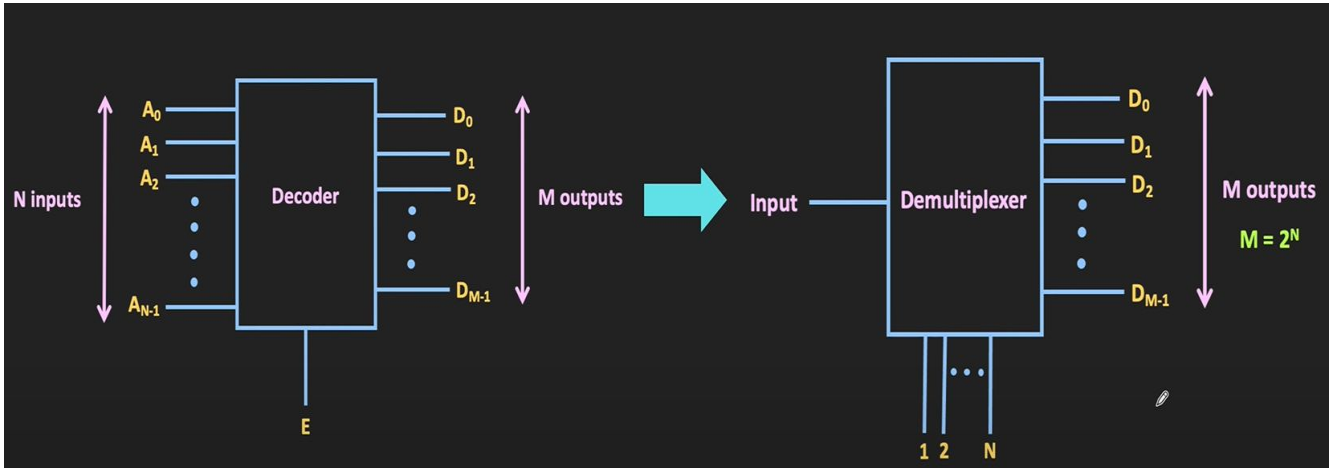
Demultiplexer



Application of demultiplexer

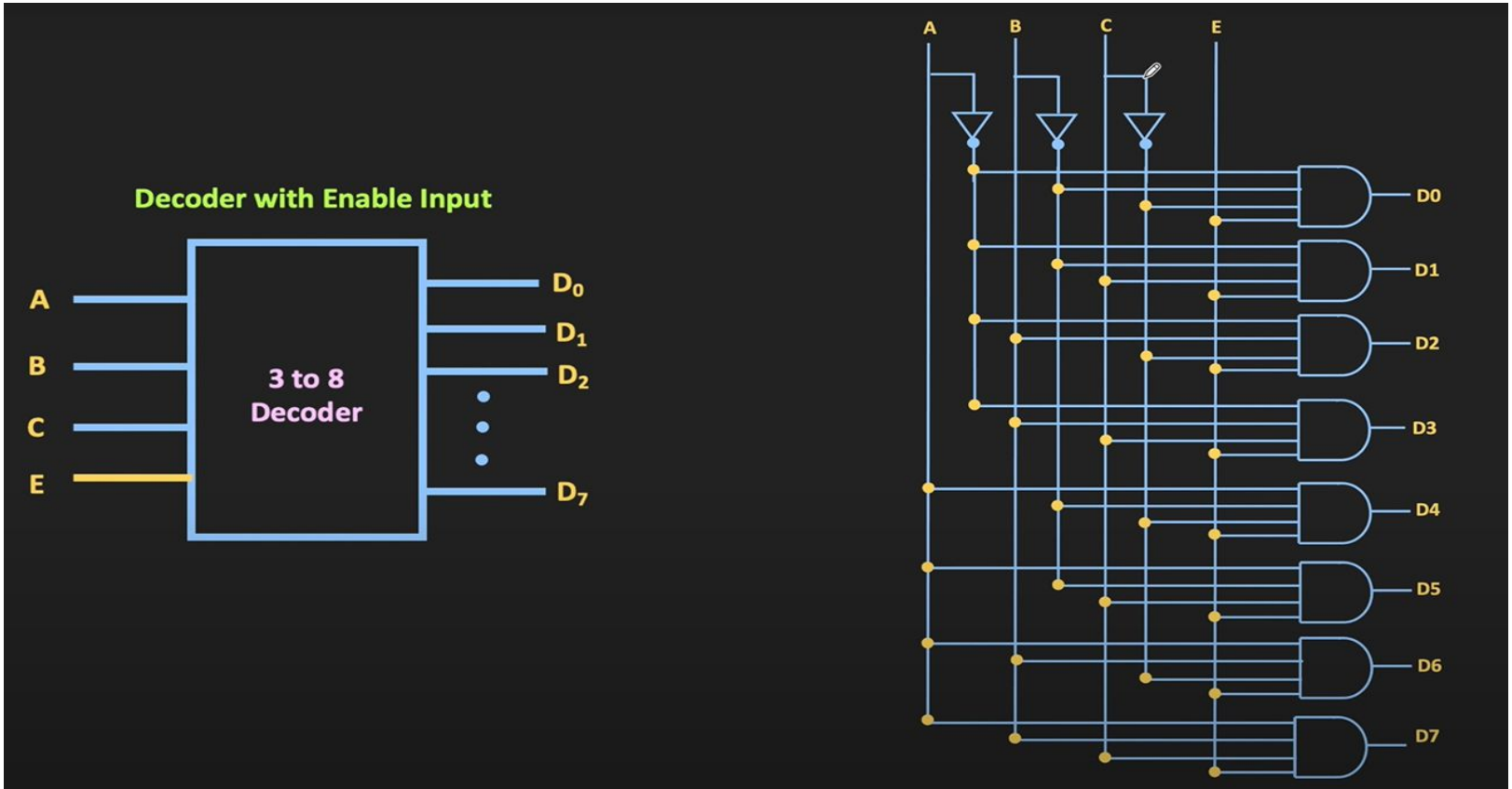


Decoder as Demultiplexer

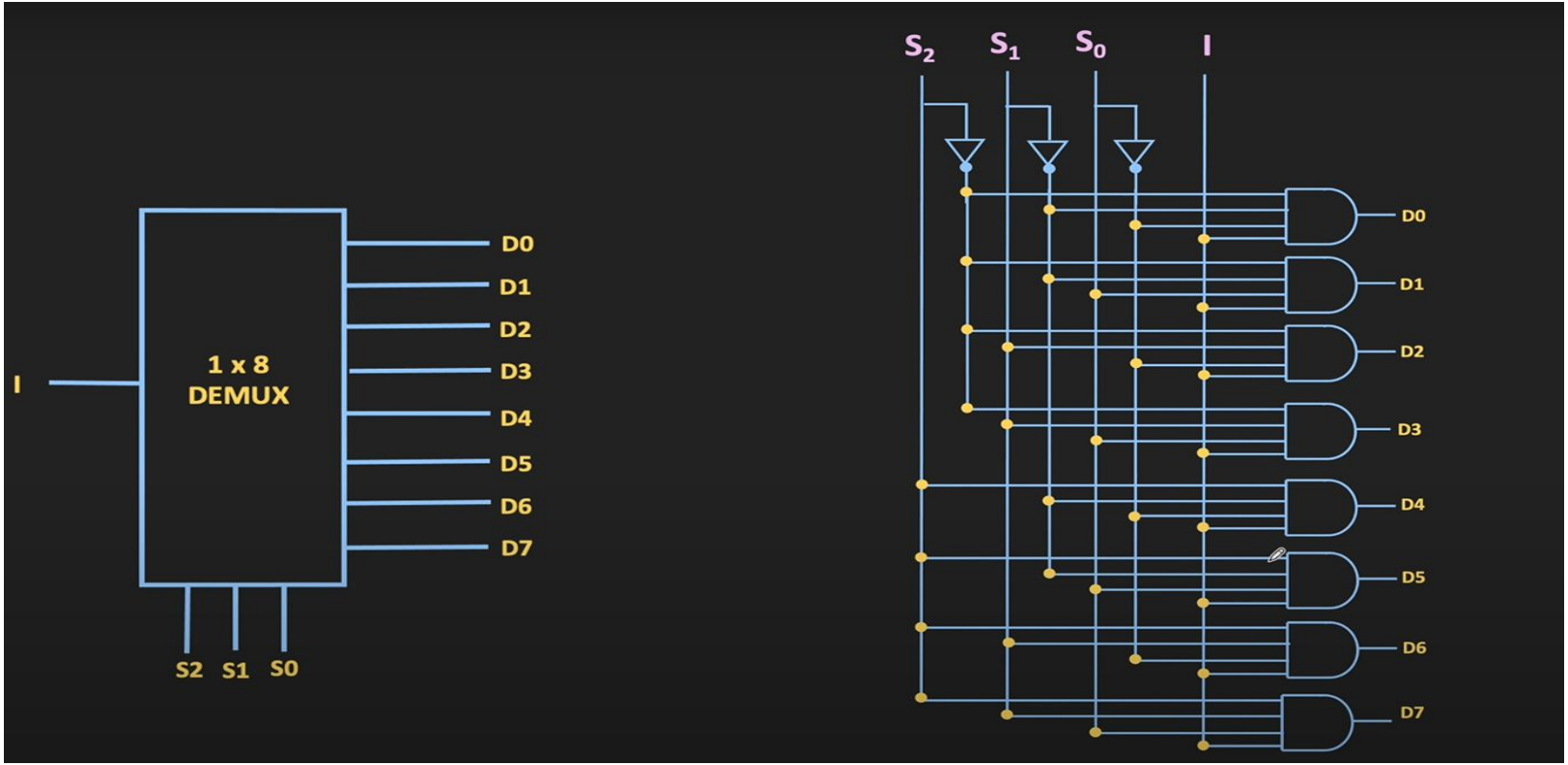


Enable signal in decoder is Input in DEMUX
N inputs are control signals in DEMUX
Outputs M=2^N in both cases

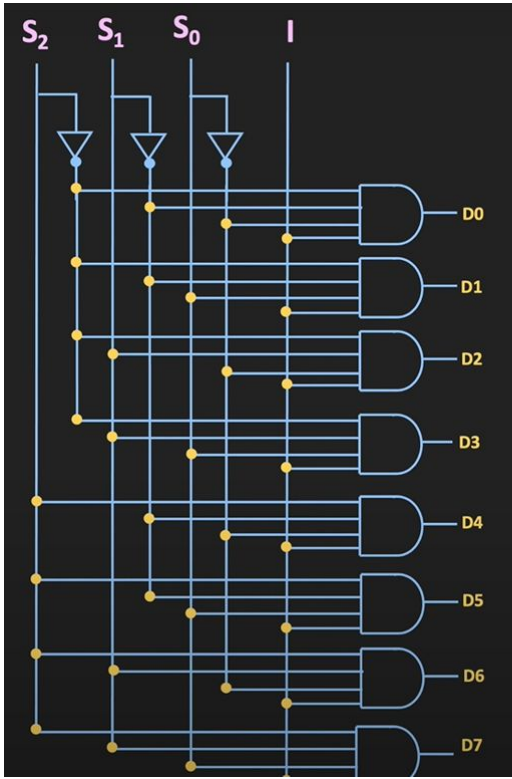
3x8 Decoder


Decoder with Enable Input
A
B
C
E
3 to 8
Decoder
D0
D1
D2
D7
A
B
C
E
D0
D1
D2
D3
D4
D5
D6
D7

1x8 DEMUX (3 control lines) using logical gates



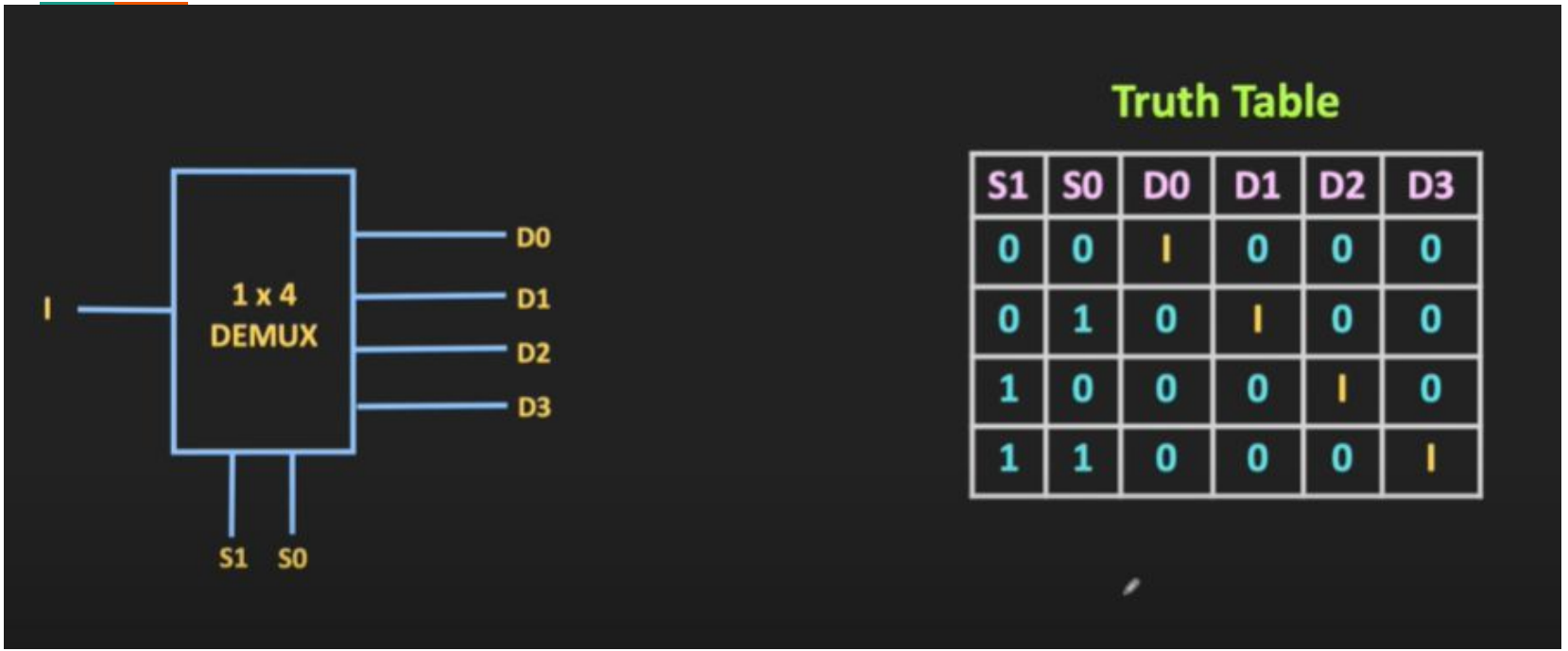
1x8 Demux Truth Table



Truth Table

S2	S1	S0	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	I	0	0	0	0	0	0	0
0	0	1	0	I	0	0	0	0	0	0
0	1	0	0	0	I	0	0	0	0	0
0	1	1	0	0	0	I	0	0	0	0
1	0	0	0	0	0	0	I	0	0	0
1	0	1	0	0	0	0	0	I	0	0
1	1	0	0	0	0	0	0	0	I	0
1	1	1	0	0	0	0	0	0	0	I

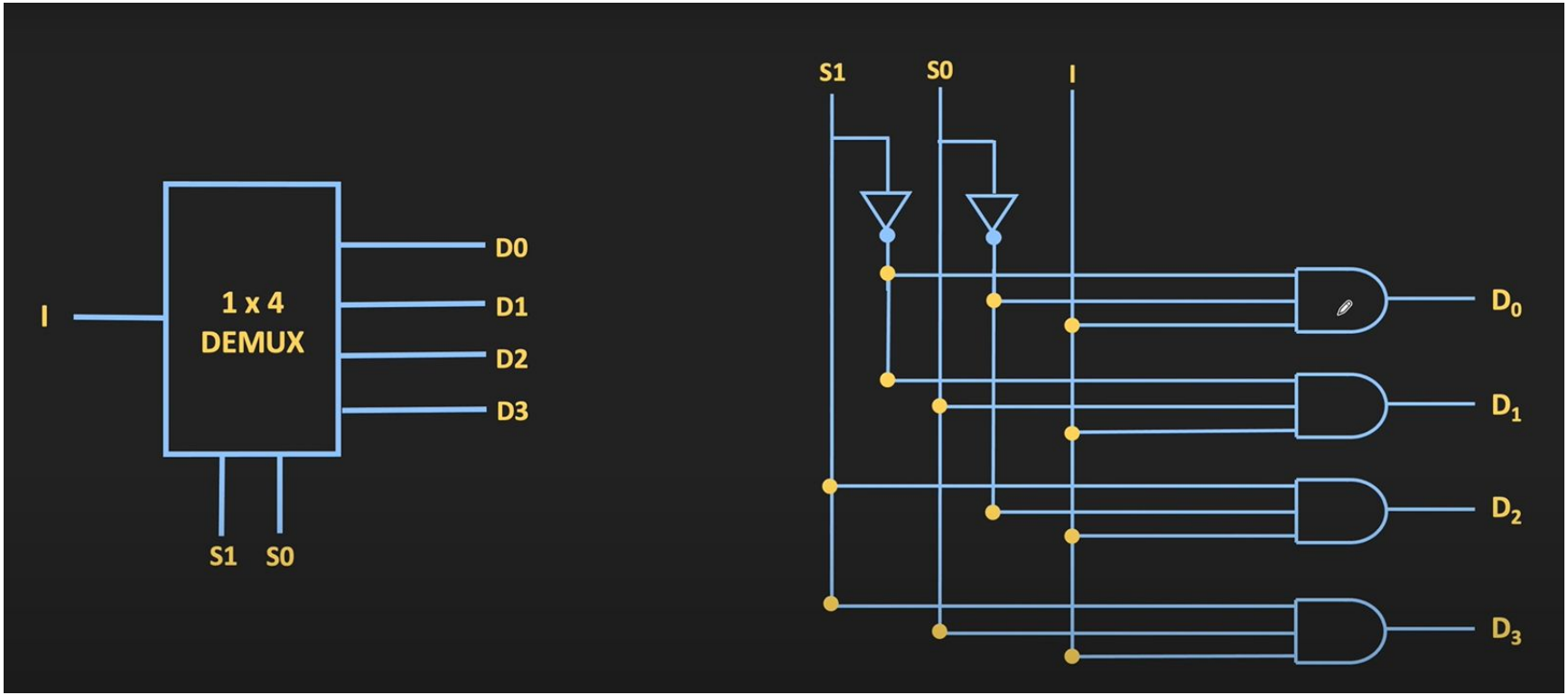
1x4 Demux Truth Table



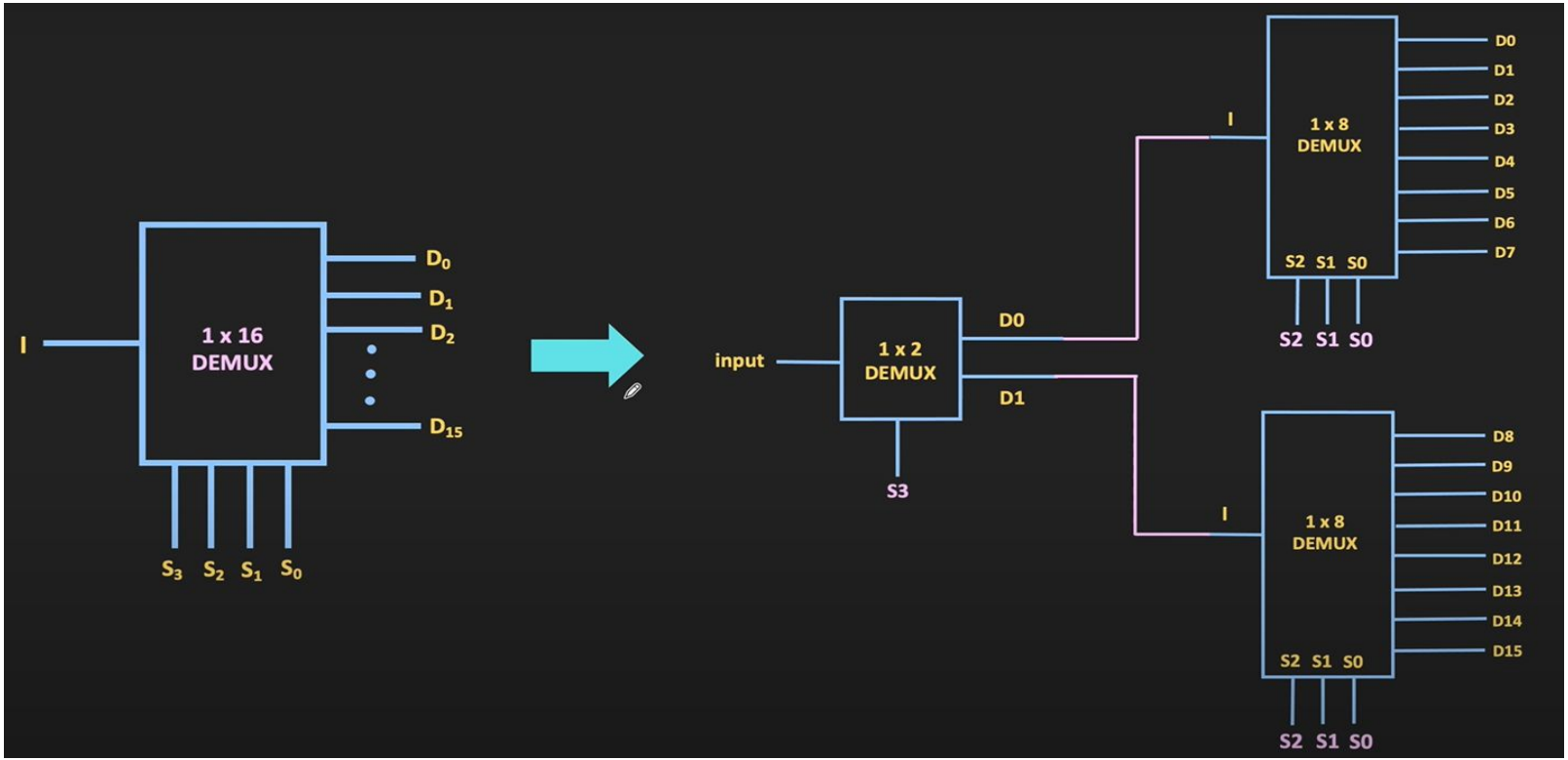
Truth Table

S1	S0	D0	D1	D2	D3
0	0	I	0	0	0
0	1	0	I	0	0
1	0	0	0	I	0
1	1	0	0	0	I

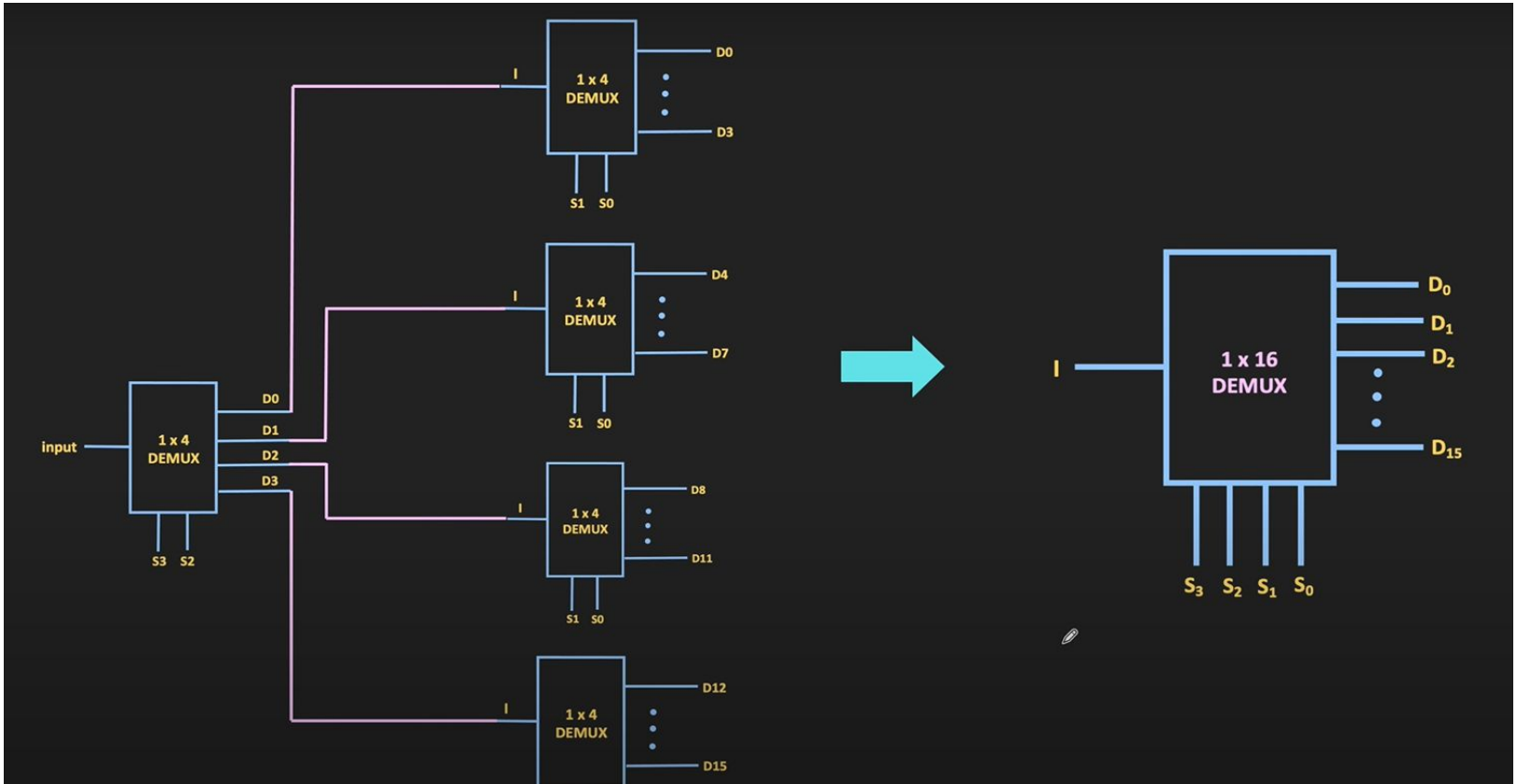
1x4 DEMUX using Logical Gates



1x16 DEMUX using 1x8 DEMUX



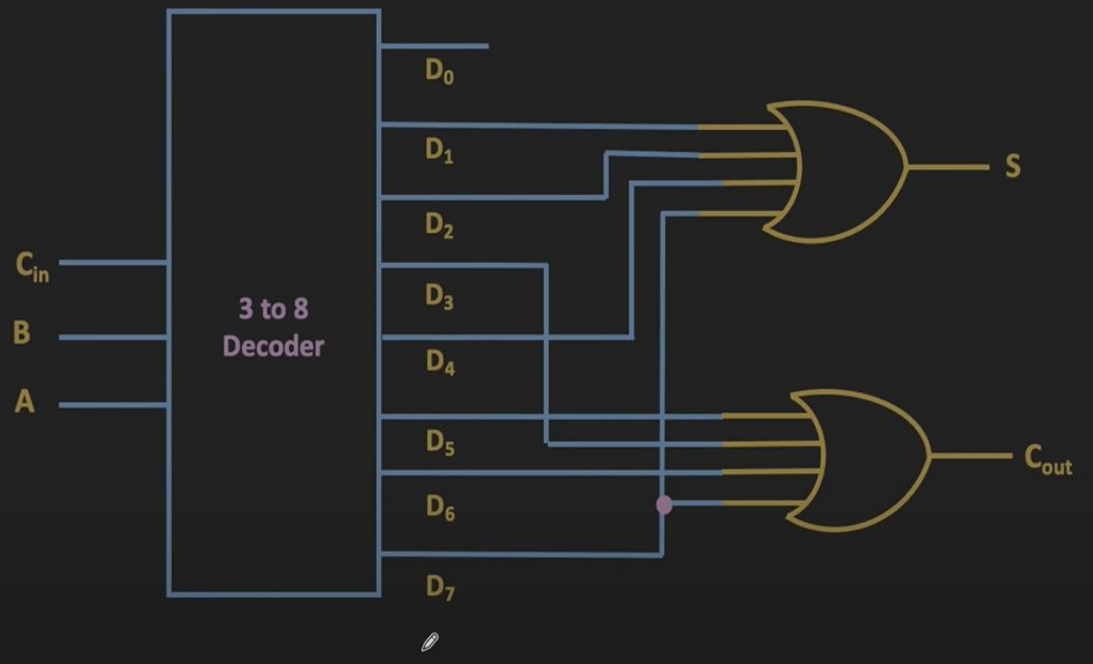
1x16 DEMUX using 1x4 DEMUX



Full adder using Decoder

Full Adder

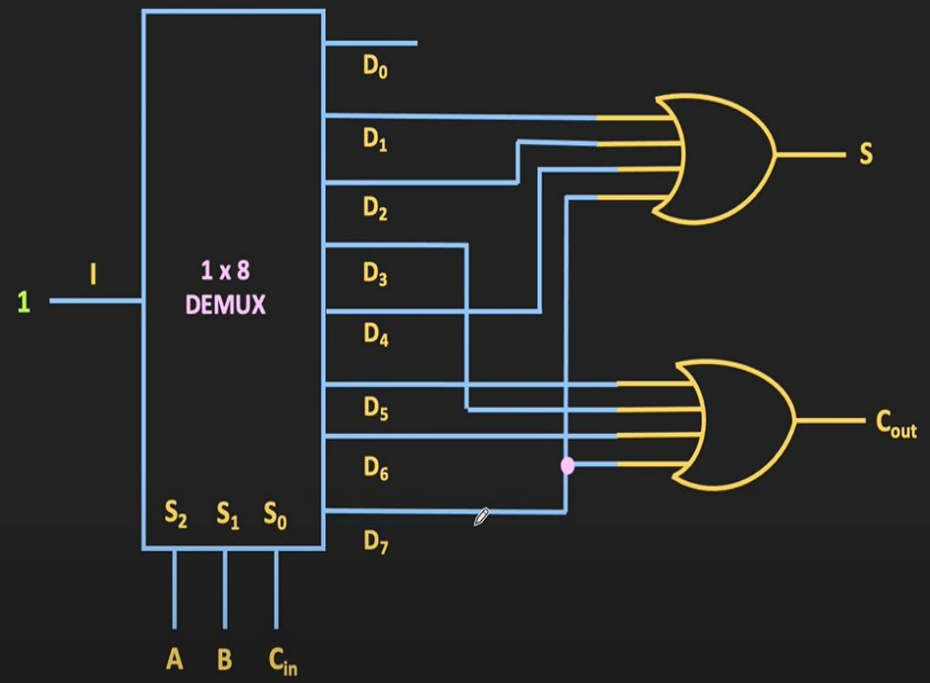
A	B	C_{in}	Sum	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Full adder using DEMUX

Full Adder

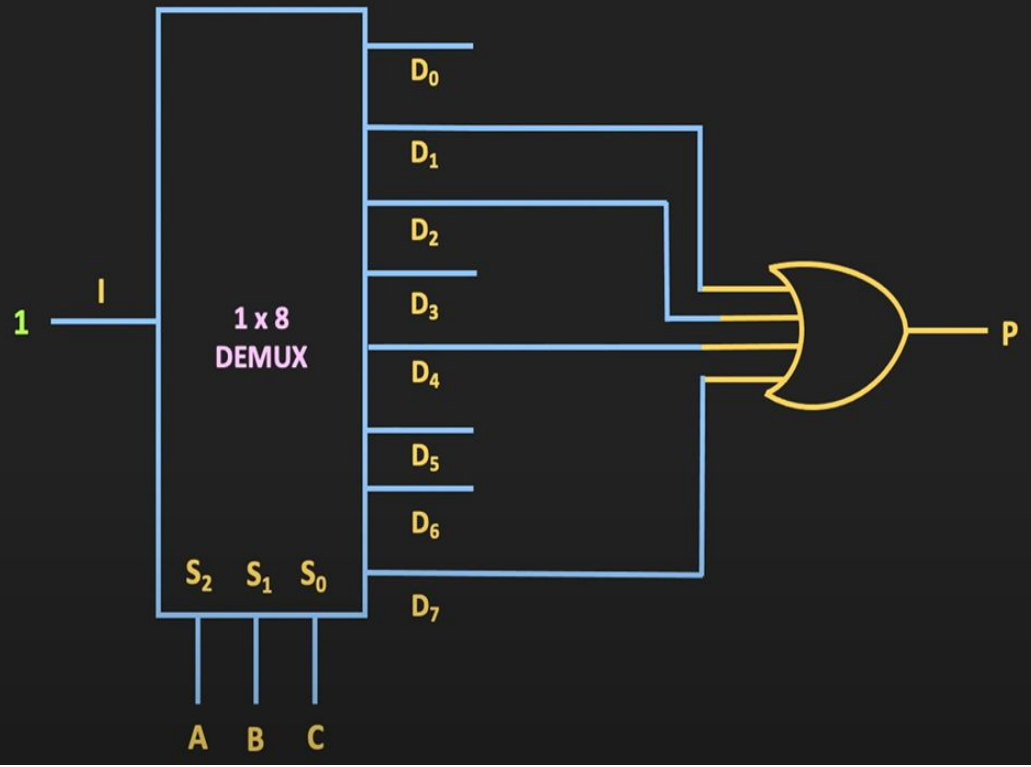
A	B	C_{in}	Sum	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Even Parity using Demux (1x8)

Even Parity Generator

A	B	C	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



FAQs in interviews (1)

1. **What is the difference between a demultiplexer and a decoder?**

The main difference between a demultiplexer and a decoder lies in their applications. A demultiplexer is primarily used for data routing and channel selection purposes, whereas a decoder is typically used to convert coded inputs into corresponding output lines.

Question: **What is the demultiplexing ratio, and how does it impact the performance of a demultiplexer?**

Answer: The demultiplexing ratio refers to the number of output lines in a demultiplexer. It represents the number of possible output options or channels available for routing the input signal. The demultiplexing ratio affects the granularity and flexibility of data routing. A higher demultiplexing ratio allows for more output options but may result in increased complexity and signal degradation due to increased circuitry and potential data skew issues.

FAQs in interviews (2)

Question: What are the limitations and challenges associated with implementing a high-speed demultiplexer?

Answer:

- **Signal propagation delay:** As the demultiplexer circuitry becomes more complex, the signal propagation delay increases, which can limit the maximum achievable speed.
- **Data skew:** At high speeds, data skew may occur, causing timing discrepancies between different output channels and affecting signal integrity.
- **Power consumption:** High-speed demultiplexers may consume significant power due to increased circuit complexity and switching activities.
- **Signal integrity:** High-speed demultiplexers may be prone to signal degradation, including noise, crosstalk, and reflections, which can impact overall system performance.

FAQs in interviews (3)

Question: What is the concept of "fan-out" in demultiplexers, and why is it significant in digital circuit design?

Answer: Fan-out refers to the maximum number of inputs that a single output of a demultiplexer can drive without signal degradation. It represents the output current or power capability of the demultiplexer. Fan-out is crucial in digital circuit design because it determines the number of subsequent logic gates or devices that can be connected to the demultiplexer output without causing issues such as voltage drop, reduced noise margins, or propagation delays.

Question: Explain how a demultiplexer can be used to implement a binary decoder, and provide an example. → Answer in the above slides

FAQs in interviews (4)

Question: What are the advantages and disadvantages of using a demultiplexer-based design compared to a combinational circuit implementation for a specific application?

Answer:

Advantages:

- **Simplicity:** Demultiplexer-based designs can be simpler to implement compared to complex combinational circuits.
- **Flexibility:** Demultiplexers offer flexibility in data routing, making them suitable for applications such as channel selection, data distribution, and time-division multiplexing.
- **Reduced gate count:** Demultiplexer-based designs can potentially reduce the number of logic gates required compared to complex combinational circuits.

FAQs in interviews (5)

Disadvantages:

- **Limited functionality:** Demultiplexers can only perform data routing based on select lines. They lack the ability to perform complex logical functions like combinational circuits.
- **Signal degradation:** The use of demultiplexers may introduce additional signal propagation delay, data skew, and potential signal degradation, especially at high speeds.
- **Complexity with large inputs/outputs:** As the number of inputs or outputs increases, the complexity of the demultiplexer-based design grows, potentially leading to increased power consumption and signal integrity challenges.

FAQs in interviews (6)

Question: What is data skew in demultiplexers, and how can it be mitigated?

Answer:

- Data skew refers to the timing discrepancy or variation between different output channels of a demultiplexer.
- It can occur due to differences in signal propagation delay or other factors. Data skew can lead to timing errors and signal integrity issues.
- To mitigate data skew, techniques such as careful layout design, signal buffering, and equalizing trace lengths can be employed.
- Additionally, using high-speed demultiplexers and minimizing signal path differences can help reduce data skew.

FAQs in interviews (7)

Question: What is the role of tri-state outputs in demultiplexers, and how can they be utilized in digital circuit design?

Answer:

- Tri-state outputs in demultiplexers allow for a third state in addition to the standard high and low logic levels. This third state, known as the "high-impedance" or "floating" state, effectively disconnects the output from the circuit.
- Tri-state outputs can be utilized to implement bus-oriented designs, where multiple demultiplexer outputs are connected to a common bus.
- By enabling or disabling specific outputs using control signals, one demultiplexer output can drive the bus while others remain in the high-impedance state, preventing data conflicts.

FAQs in interviews (8)

Question: What are "level-sensitive" and "edge-sensitive" demultiplexers, and how are they applied in synchronous digital systems?

Level-sensitive demultiplexers use the levels (high or low) of the select lines to control the data routing. The output remains stable as long as the select line levels are maintained.

Edge-sensitive demultiplexers, on the other hand, respond to transitions or changes in the select line levels, specifically the rising or falling edges. These demultiplexers typically latch the select line inputs at the occurrence of an edge to determine the data routing.

Level-sensitive demultiplexers are commonly used in combinational circuits, while edge-sensitive demultiplexers find application in synchronous systems where data is synchronized to clock edges.

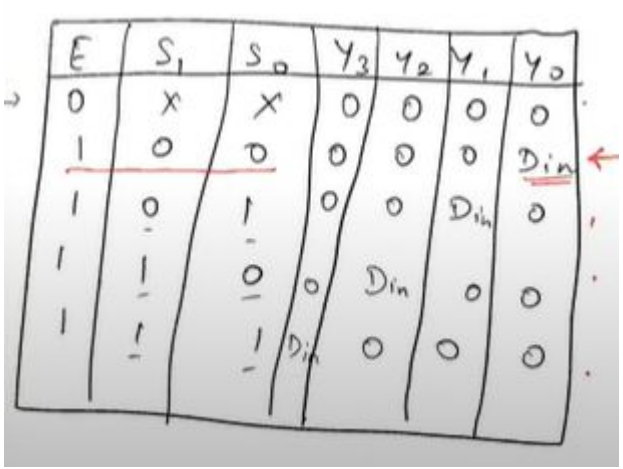
FAQs in interviews (9)

Question: What is a functional hazard in demultiplexer operation, and how can it be eliminated or minimized?

- A functional hazard in demultiplexer operation refers to a temporary discrepancy in the output caused by a momentary change in the select lines, even though the input conditions have not changed.
- It occurs due to variations in signal propagation delay paths within the demultiplexer. Functional hazards can result in incorrect output values and unreliable operation.
- **To eliminate or minimize functional hazards**, hazard detection and hazard cover techniques can be employed. Hazard detection involves identifying the specific input conditions that may cause hazards.
- Hazard cover involves introducing additional logic or delaying circuit elements to ensure that the output stabilizes correctly, preventing hazards from occurring.
- Techniques such as adding delays, using hazard cover logic, or incorporating hazard-free designs can help eliminate or reduce functional hazards.

FAQs in interviews (10)

Question: 1x4 Demux using only NAND logic gates

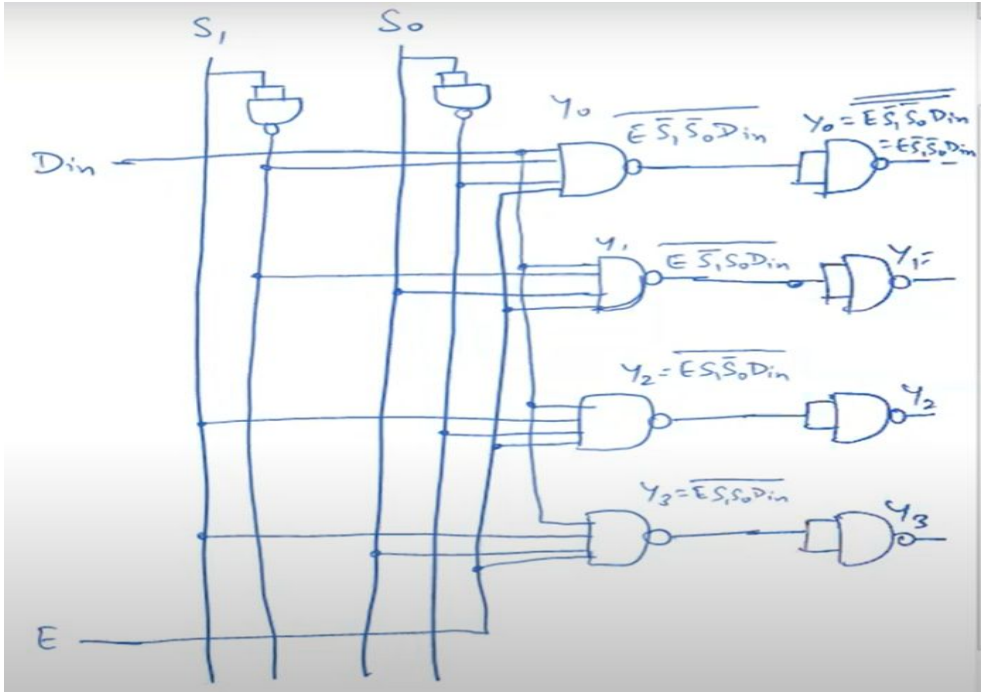


E	S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	X	X	0	0	0	0
1	0	0	0	0	0	Din
1	0	1	0	0	Din	0
1	1	0	0	Din	0	0
1	1	1	Din	0	0	0

Here E is enable signal
Din is the input
Y3, Y2, Y1 and Y0 are the outputs
S0 and S1 are the control/selector Lines

FAQs in interviews (11)

Question: 1x4 Demux using only NAND logic gates



Note: Not gate can be represented by using NAND gate by shorting its inputs