Q1: Decoders and Encoders  
a) Explain decoder and encoder:

* Decoder: A decoder is a digital circuit that converts binary information from n input lines to a maximum of 2^n unique output lines. It takes an n-bit binary input and activates one of the 2^n output lines based on the input value.
* Encoder: An encoder is a digital circuit that performs the reverse operation of a decoder. It converts 2^n unique input lines into n output lines, effectively converting the physical location of the input into a binary code.

b) Applications of decoders and encoders:

* Decoders are used in applications such as seven-segment displays, memory addressing, and data demultiplexing.
* Encoders are used in applications such as keyboards, priority encoders, and analog-to-digital conversion.

c) Examples:

* Decoder example: A 3-to-8 line decoder takes a 3-bit binary input and activates one of the 8 output lines based on the input value.
* Encoder example: A priority encoder takes 8 input lines and generates a 3-bit binary output that represents the highest priority input line that is active.

Q2: Multiplexers and Demultiplexers  
a) Explain Multiplexers and Demultiplexers:

* Multiplexer (MUX): A multiplexer is a digital circuit that selects one of several analog or digital input signals and forwards the selected input into a single output line.
* Demultiplexer (DEMUX): A demultiplexer is a digital circuit that takes a single input signal and routes it to one of many outputs, based on a set of selection lines.

b) Applications of multiplexers and demultiplexers:

* Multiplexers are used in applications such as data selection, address decoding, and time-division multiplexing.
* Demultiplexers are used in applications such as data distribution, memory addressing, and display driving.

c) Examples:

* Multiplexer example: A 4-to-1 multiplexer takes four input signals and selects one of them to be routed to the output, based on two selection lines.
* Demultiplexer example: A 1-to-4 demultiplexer takes a single input signal and routes it to one of four outputs, based on two selection lines.

Q3: 4-to-1 Multiplexing using Gates  
a) 2-to-1 MUX:

* Truth table:

| **Select** | **Input 0** | **Input 1** | **Output** |
| --- | --- | --- | --- |
| 0 | A | X | A |
| 1 | X | B | B |

* Circuit design:

Copy

Y = (S' \* A) + (S \* B)

b) 4-to-1 MUX:

* Truth table:

| **Select 1** | **Select 0** | **Input 0** | **Input 1** | **Input 2** | **Input 3** | **Output** |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | A | X | X | X | A |
| 0 | 1 | X | B | X | X | B |
| 1 | 0 | X | X | C | X | C |
| 1 | 1 | X | X | X | D | D |

* Circuit design:

Copy

Y = (S1' \* S0' \* A) + (S1' \* S0 \* B) + (S1 \* S0' \* C) + (S1 \* S0 \* D)

Q4: 2-to-4 Decoder  
a) Truth table for a 2-to-4 decoder with enable bit:

| **Enable** | **Select 1** | **Select 0** | **Output 0** | **Output 1** | **Output 2** | **Output 3** |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | X | X | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |

b) Gate diagram for a 2-to-4 decoder:

Copy

EN

|

┌───┐

│ │

┌┴───┴┐

│ S1 │

└────┘

|

┌───┐

│ │

┌┴───┴┐

│ S0 │

└────┘

|

┌───┐ ┌───┐ ┌───┐ ┌───┐

│ O0 │ │ O1 │ │ O2 │ │ O3 │

└───┘ └───┘ └───┘ └───┘

c) Implementation of a 2-to-4 decoder on a trainer:

* Use three-input AND gates with the enable bit as an active-low input.
* Connect the select lines (S1 and S0) to the other two inputs of the AND gates.
* Verify the operation against the truth table.