Module 5 : Decoder & Encoder

Learning Objectives :

1. Understand the working principles of decoders and encoders.

2. Being able to distinguish between a decoder and an encoder, as well as knowing when to use each one

- Theory
- Video

Theory

1. Decoder

In general, decoder is a combinatorial device that has n input ports and m output ports such that `n <= m <= 2n`. A decoder is commonly prefixed with the phrase `n-to-m`, where n and m are the number of input and output ports respectively.
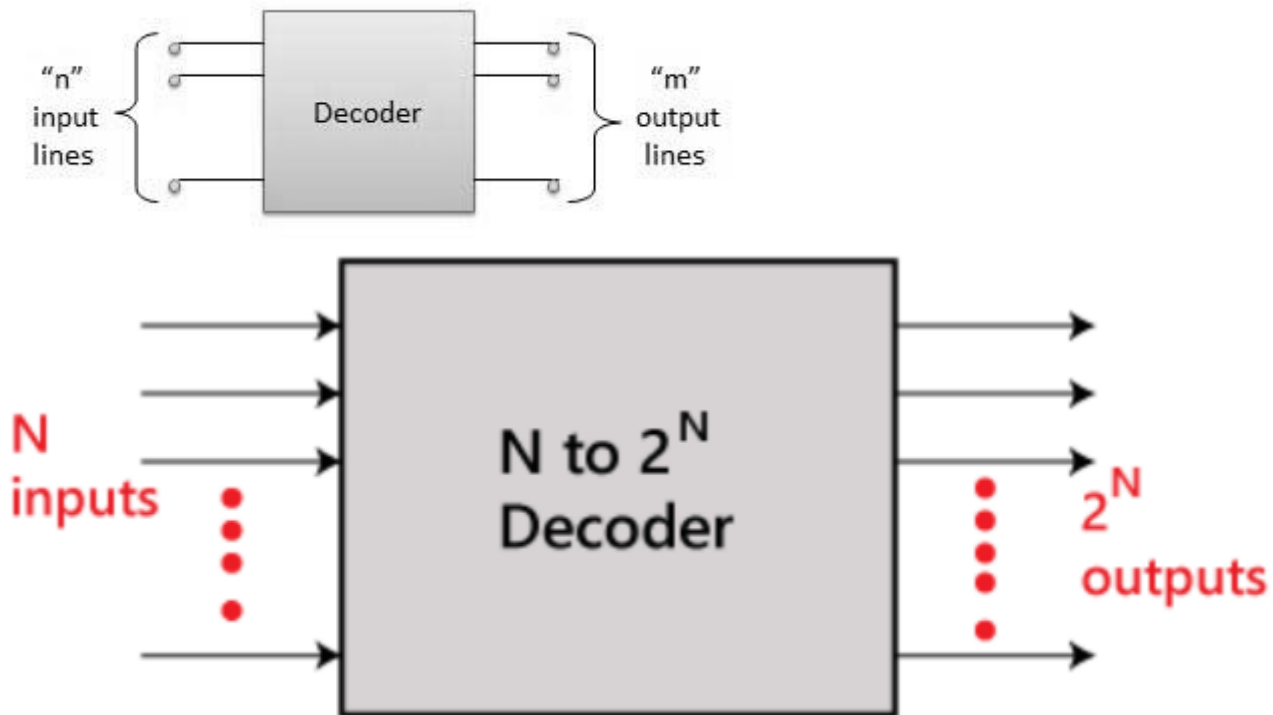

Figure 1-1 General Shape of a Decoder

Decoders are used for decoding, which is a process of converting or "translating" an input code which is hard to deal with, to an output code which is easier to deal with. For example, a binary decoder is a type of decoder that converts an n-bit binary number to its corresponding minterm index or a one-hot representation of the number. So, a 3-to-8 binary decoder would have the truth table:

Input (A)			Output (D)							
A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Explanation : If we're looking for the number 5 (101), we just need to check port D5. If D5 is 0, then the number A is definitely not 5. If, however, D5 is 1, then it is certain that the number A is 5. Checking the output port D5 is easier than checking straight from the undecoded input. We would need to check if A2=1 AND A1=0 AND A0=1, which is tedious, especially for larger numbers with more bits.

Another type of decoder is the 7-segment decoder. It converts a one-digit BCD (4-bit) input to a 7-bit output. Each output line correspond to a segment on a 7-segment display and when connected would result in a digit showing up in the display. For example, when the input is the number 5 (0101 in BCD), the output would light up appropriate segments of the display so that the actual "5" appears in it. Two such 7-segment decoders are IC 7447 and IC 7448

Figure 1-2 A 7-Segment Display

Figure 1-3 Patterns on a 7-segment display could display all 10 decimal digits

TRUTH TABLE:

BCD Inputs				Output Logic Levels from IC 7447 to 7-segments							Decimal number display
D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0	2
0	0	1	1	0	0	0	0	1	1	0	3
0	1	0	0	1	0	0	1	1	0	0	4
0	1	0	1	0	1	0	0	1	0	0	5
0	1	1	0	1	1	0	0	0	0	0	6
0	1	1	1	0	0	0	1	1	1	1	7
1	0	0	0	0	0	0	0	0	0	0	8

Feature	IC 7447	IC 7448
Display Type	Common Anode	Common Cathode
Output Polarity	Active Low	Active High
Display Control	Turns on segments with logic 0	Turns on segments with logic 1
Usage	For displays with connected anodes	For displays with connected cathodes

- [74LS47 Datasheet](#)
- [74LS48 Datasheet](#)

2. Encoder

Encoder is the counterpart to the decoder; it does the inverse of what the decoder does, which is encoding. An encoder has m input lines and n output lines $n <= m <= 2n$. Like the decoder, encoder is usually prefixed with the phrase $m\text{-to-}n$, where m and n are the number of input and output ports respectively.
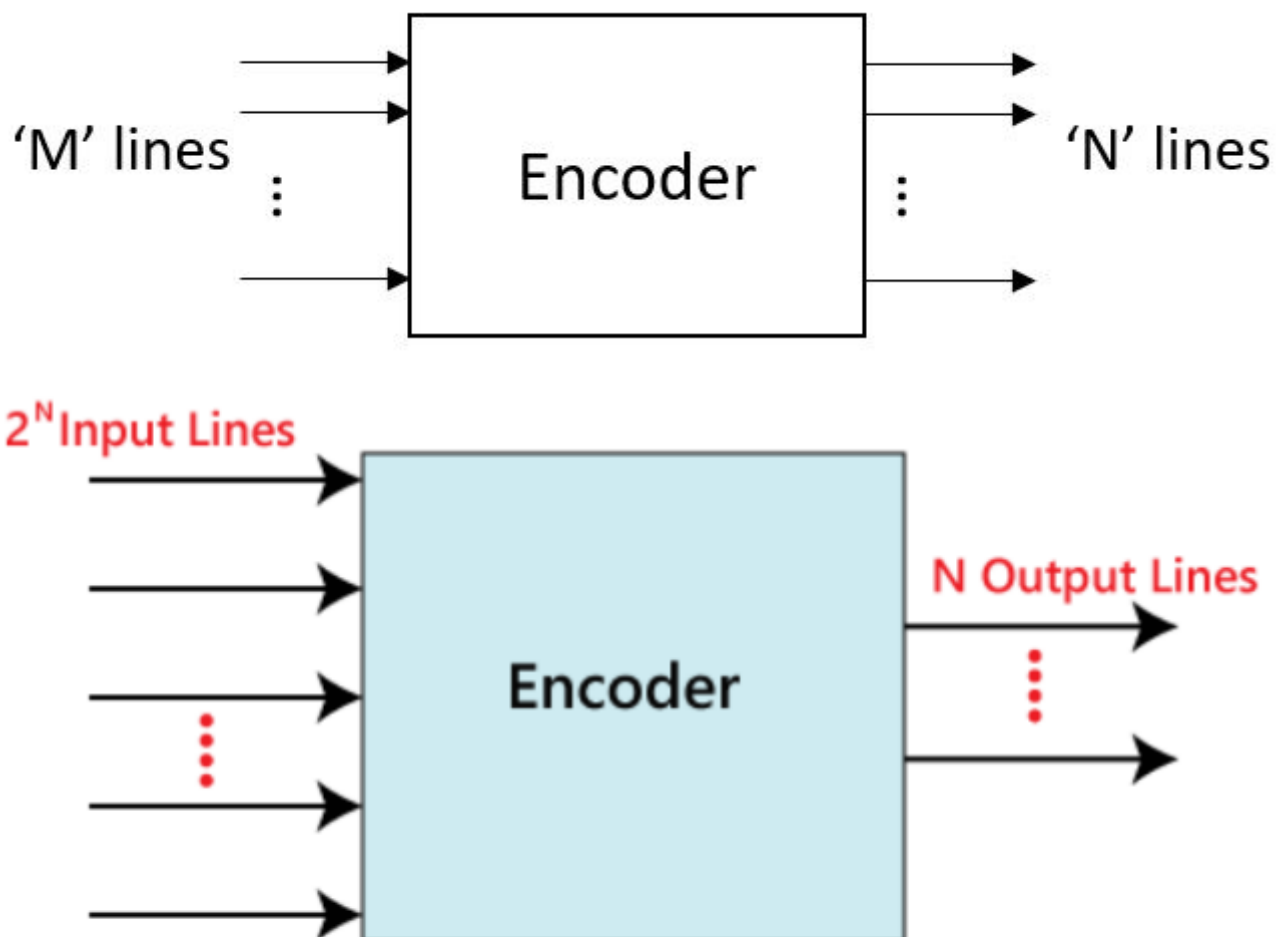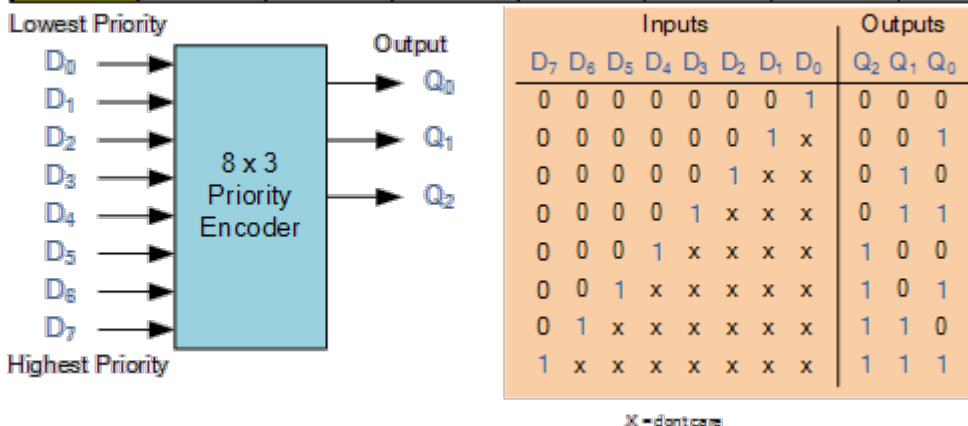

Figure 2-1 General Shape of an Encoder

Generally, to encode means to take an input which has more lines/bits, to an output which has less lines/bits. This is done so that the resulting output could be transferred or transmitted to another location easily because it's more packed and require less bits. For example, a binary encoder would output a binary number that correspond to the minterm index or one-hot representation of the input. So, an 8-to-3 binary encoder would have the truth table :

Input (D)								Output (A)		
D7	D6	D5	D4	D3	D2	D1	D0	A2	A1	A0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

But, there's a problem: not every input combination is defined in the truth table. For example, what would the output of the encoder be if 00000011 is given as the input? To solve that issue, we introduce a special type of encoder, called the priority encoder. It has the truth table :

Input (D)								Output (A)		
D7	D6	D5	D4	D3	D2	D1	D0	A2	A1	A0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	X	0	0	1
0	0	0	0	0	1	X	X	0	1	0
0	0	0	0	1	X	X	X	0	1	1
0	0	0	1	X	X	X	X	1	0	0
0	0	1	X	X	X	X	X	1	0	1
0	1	X	X	X	X	X	X	1	1	0
1	X	X	X	X	X	X	X	1	1	1

A priority encoder "prioritizes" input lines so that one line takes precedence over another. In this case, the input line with a bigger index takes precedence over the ones with a smaller index. If we take the example from before, when we give the priority encoder 00000011 as input, it "ignores" D0 and outputs 001. If we take another number: 10101010, then it would only "care" about D7 and ignore the rest. The resulting output would be 111.

Priority encoders are available in standard IC form and the TTL 74LS148 is an 8-to-3 bit priority encoder which has eight active LOW (logic "0") inputs and provides a 3-bit code of the highest ranked input at its output.

- 74LS148 Datasheet

Additional Explanation

In addition to standard encoders like the priority encoder or decoders such as the BCD-to-7-segment decoder, we can create **custom encoders and decoders** based on specific application needs. These custom designs allow us to define unique encoding or decoding schemes that go beyond the limitations of previous standard examples & components (IC such as 7447,7448,74148) offering flexibility for various digital systems. Your Case Study or Additional Task and Final Project of this subject (Digital Systems Basics) will most likely need you to design these other than using the examples of decoder & encoder mentioned above.

Video

https://www.youtube.com/embed/2iSSgRxi3NA