ACTIVITY 3: DECODER

Jason Harvey Lorenzo, March 4, 2024
ENGG 125.03, Department of Electronics, Computer, and Communications Engineering
School of Science and Engineering, Ateneo de Manila University
Quezon City, 1108, Philippines
jason.lorenzo@obf.ateneo.edu

***Abstract*—This paper details the creation of a decoder circuit designed to mimic the behavior of a 74XX138 IC. By analyzing its truth table and utilizing logic gates, the decoder links its outputs (Y0' to Y7') with specific input combinations of C, B, A, and their complements (as maxterms). Special attention is given to enable inputs G2B', G2A', and G1, which directly influence the outputs, particularly in detecting the input pattern 001. The implemented HDL code efficiently captures this logic, and simulation results validate its functionality.**

***Index Terms*—Verilog HDL, decoder**

This activity involves the construction of a decoder, specifically the task is to imitate the function of a 74XX138 IC as outlined in Fig. 1.

IN	IN	IN	IN	IN	IN	OUT	OUT	OUT	OUT	OUT	OUT	OUT	OUT
G2B'	**G2A'**	**G1**	**C**	**B**	**A**	**Y0'**	**Y1'**	**Y2'**	**Y3'**	**Y4'**	**Y5'**	**Y6'**	**Y7'**
X	X	0	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
1	X	X	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

Note: "X" means "don't care",i.e., may take the values of 0 or 1.

Fig. 1. 74XX138 Truth Table

As depicted in Fig. 1, the output Y0' corresponds precisely to a single maxterm, denoted as M_0, while Y1' aligns with M_1, Y2' with M_2, and so forth. Essentially, these outputs are OR combinations of the variables C, B, A, and their respective negations, as demonstrated in Table I. This comprehensive representation effectively encompasses the outputs (Y0' to Y7') within the enabled/functional segment of the truth table.

Additionally, as evident from Fig. 1, it becomes apparent that unless the enable inputs G2B', G2A', and G1 are set to 0, 0, and 1, respectively, all outputs will default to 1. This underscores the direct influence of these enable inputs on the ultimate outputs. Specifically, the input pattern 001 can be easily identified by utilizing a triple-input OR gate with the least significant bit negated for $G1 = 1$. Referred to as 'enable' henceforth, the output of this gate will register zero only if the input corresponds to 001; otherwise, it will output 1. In the absence of enable, signifying the un-enabled section of the truth table, all outputs (Y0' to Y7') will uniformly assume a value of 1. This segment can be readily OR-operated with the enabled/functional part linked to the maxterms to yield the desired output. When the enable input is not 001, it will default to 1, rendering the final output independent of the functional part's value (X+1=1). Conversely, if the enable is 0, the final output is dependent on the output of the functional part.

TABLE I
OUTPUT AND THEIR SINGLE MAXTERMS EQUIVALENT

OUT	Maxterm	As Single Sum
Y0'	M_0	C + B + A
Y1'	M_1	C + B + A'
Y2'	M_2	C + B' + A
Y3'	M_3	C + B' + A'
Y4'	M_4	C' + B + A
Y5'	M_5	C' + B + A'
Y6'	M_6	C' + B' + A
Y7'	M_7	C' + B' + A'

I. HDL CODE

Listing 1 shows the HDL code implementation of the function defined above.

Listing 1. 74XX138 Decoder HDL Code

```
module decoder(
        input G2B, G2A, G1, C, B, A,
        output Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7
);

wire enable;

assign enable = G2B | G2A | ~G1;

assign Y0 = enable | (C | B | A);
assign Y1 = enable | (C | B | ~A);
assign Y2 = enable | (C | ~B | A);
assign Y3 = enable | (C | ~B | ~A);
assign Y4 = enable | (~C | B | A);
assign Y5 = enable | (~C | B | ~A);
assign Y6 = enable | (~C | ~B | A);
assign Y7 = enable | (~C | ~B | ~A);

endmodule
```

Since decoders are combinational circuit, behavioral representation with the 'assign' keyword was used for the wire and

outputs. Note that enable is created as a wire to be reused for all outputs. This wire is then combined with the functional part (maxterms) of the circuit using the OR operator to obtain the final output.

II. Output Circuit

Fig. 2 shows the output circuit of the code above.

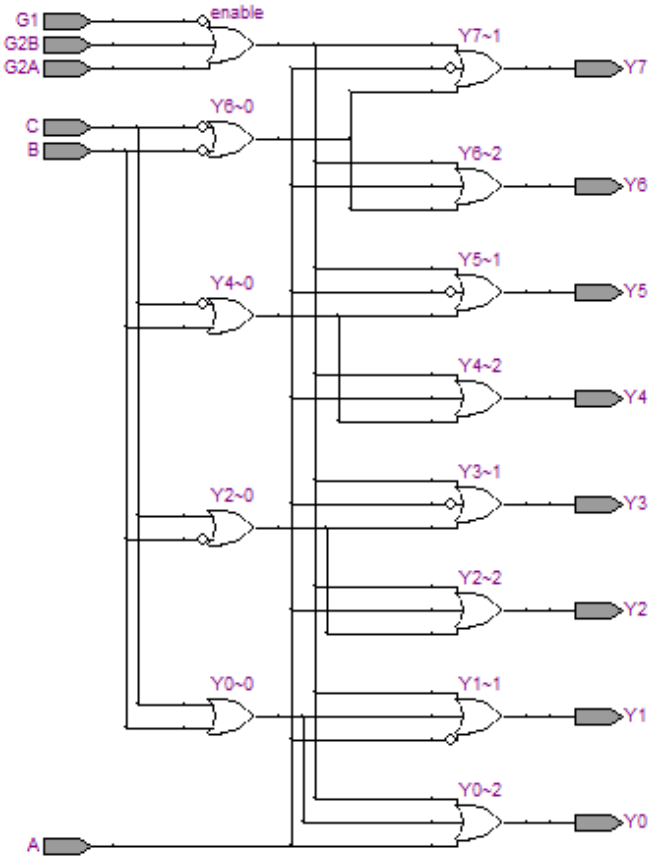

Fig. 2. 74XX138 Decoder Circuit Implementation RTL Netlist

As described previously, the enable inputs are first combined with the respective maxterms for each corresponding output using an OR operator. This combined result is then further combined with an OR operator. This solution ensures both resource and speed efficiency.

III. Simulation

To streamline the simulation of the decoder circuit, G2B' and G2A' will be set to 0. It's worth noting that if either of these becomes 1, it implies that all outputs will be 1. Figure 3 illustrates the simulated waveforms for all possible input combinations of G1, C, B, and A inputs.

As seen, this aligns well with the output observed in Fig. 1, particularly with the functional part. However, transient glitches may occur due to the non-simultaneous switching of bits, resulting in brief undesired states.

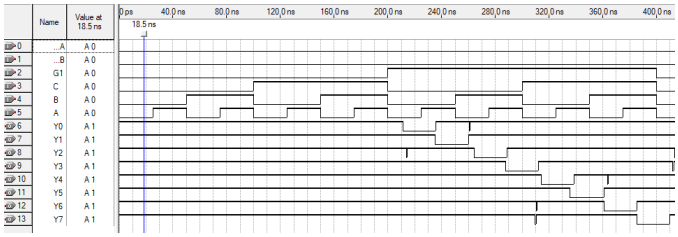

Fig. 3. 74XX138 Decoder Implementation Simulation Waveforms