INSTITUTE OF ENGINEERING AND TECHNOLOGY, INDORE

EIR7E1: Circuit Design Using HDL

"Designing 8:3 encoder on VHDL"

Submitted to:

Dr. Vaibhav Neema

Submitted By:

Parag Nainani

17E7029

8:3 Encoder

The 8 to 3 Encoder or octal to Binary encoder consists of 8 inputs: Y7 to Y0 and 3 outputs: A2, A1 & A0. Each input line corresponds to each octal digit and three outputs generate corresponding binary code.

Truth Table for 8:3 encoder

INPUTS								OUTPUTS		
Y7	**Y6**	**Y5**	**Y4**	**Y3**	**Y2**	**Y1**	**Y0**	**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

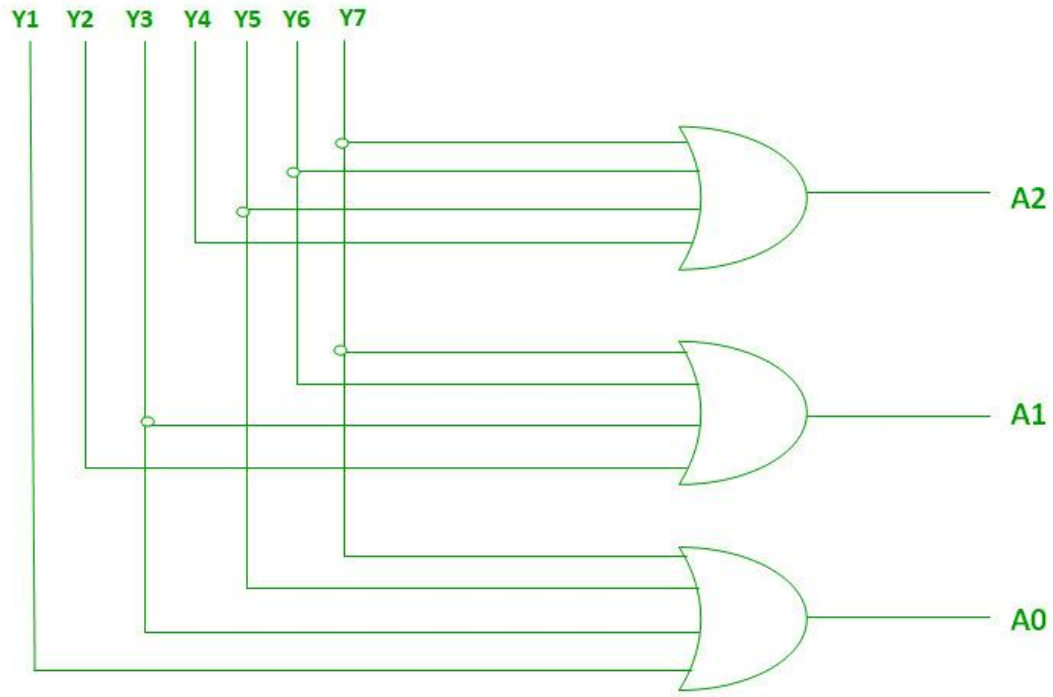
Boolean Equation for the outputs A0, A1 and A2 can be given as:

A2 = Y7 + Y6 + Y5 + Y4

A1 = Y7 + Y6 + Y3 + Y1

A0 = Y7 + Y5 + Y3 + Y1

These Equations can be implemented using 4-input OR gates



VHDL Code to implement 8:3 Encoder using Dataflow Method

DESIGN CODE

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity ENCODER_SOURCE is
    Port (Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7 : in  STD_LOGIC;
            A0,A1,A2 : out  STD_LOGIC);
end ENCODER_SOURCE;

architecture dataflow of ENCODER_SOURCE is
begin

A2 <= Y7 OR Y6 OR Y5 OR Y4;
A1 <= Y7 OR Y6 OR Y3 OR Y2;
A0 <= Y7 OR Y5 OR Y3 OR Y1;

end dataflow;
```

TESTBENCH CODE

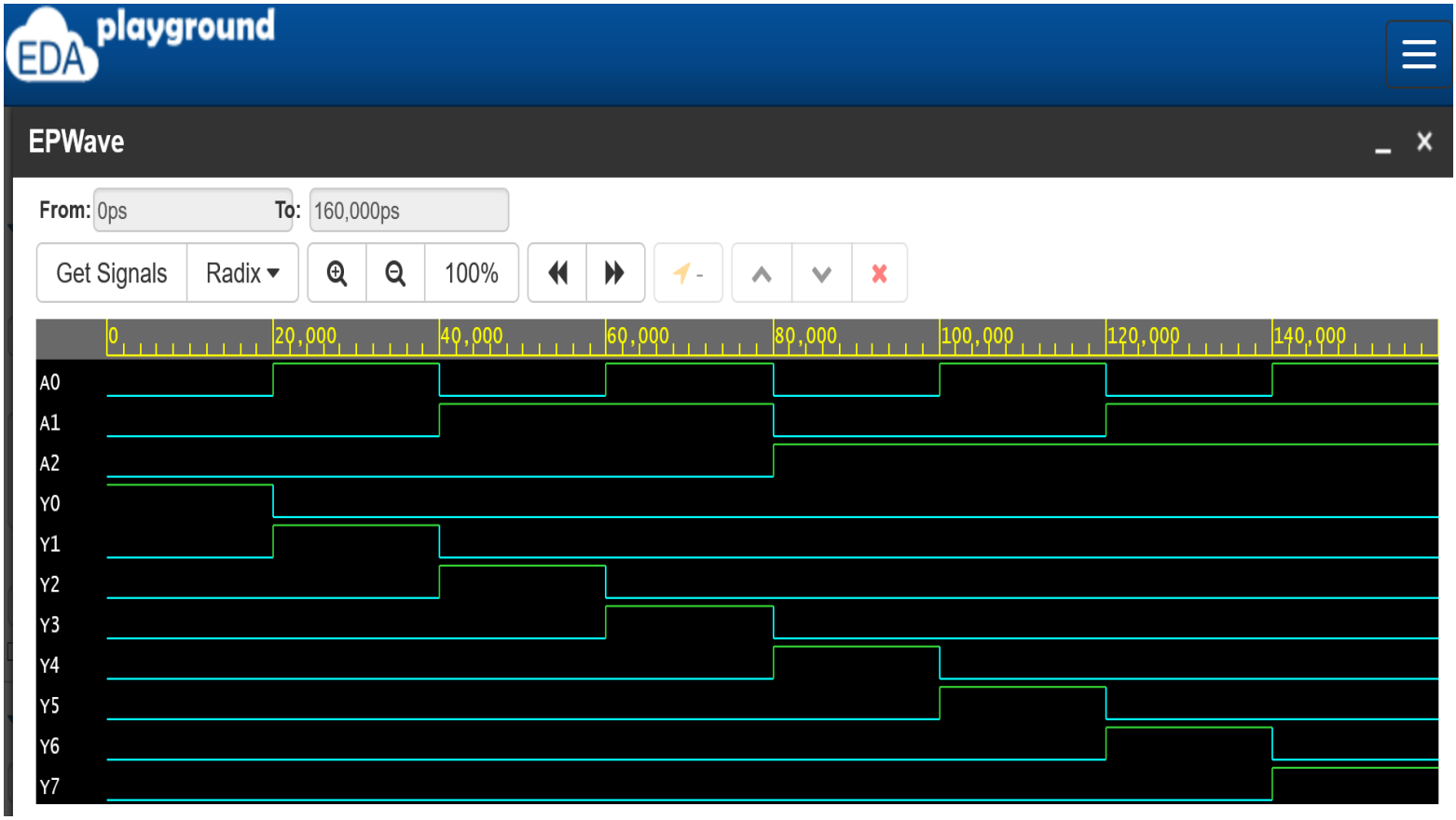
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity encoder_dataflow_tb is
end entity;

architecture encoder_dataflow_tb of encoder_dataflow_tb is
component ENCODER_SOURCE is
Port ( Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7 : in STD_LOGIC;
A0,A1,A2 : out STD_LOGIC);
end component;

signal Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7, A0, A1 ,A2: STD_LOGIC;
begin
uut: ENCODER_SOURCE port map(
Y0 => Y0, Y1 => Y1,
Y2 => Y2, Y3 => Y3,
Y4 => Y4, Y5 => Y5,
Y6 => Y6, Y7 => Y7,
A2 => A2, A1=>A1, A0=> A0);

stim: process
begin
Y7 <= '0';
```

```
Y6 <= '0';
Y5 <= '0';
Y4 <= '0';
Y3 <= '0';
Y2 <= '0';
Y1 <= '0';
Y0 <= '1';
wait for 20 ns;

Y7 <= '0';
Y6 <= '0';
Y5 <= '0';
Y4 <= '0';
Y3 <= '0';
Y2 <= '0';
Y1 <= '1';
Y0 <= '0';
wait for 20 ns;

Y7 <= '0';
Y6 <= '0';
Y5 <= '0';
Y4 <= '0';
Y3 <= '0';
Y2 <= '1';
Y1 <= '0';
Y0 <= '0';
```

```
wait for 20 ns;

Y7 <= '0';
Y6 <= '0';
Y5 <= '0';
Y4 <= '0';
Y3 <= '1';
Y2 <= '0';
Y1 <= '0';
Y0 <= '0';
wait for 20 ns;

Y7 <= '0';
Y6 <= '0';
Y5 <= '0';
Y4 <= '1';
Y3 <= '0';
Y2 <= '0';
Y1 <= '0';
Y0 <= '0';
wait for 20 ns;

Y7 <= '0';
Y6 <= '0';
Y5 <= '1';
Y4 <= '0';
Y3 <= '0';
```

```
Y2 <= '0';
Y1 <= '0';
Y0 <= '0';
wait for 20 ns;

Y7 <= '0';
Y6 <= '1';
Y5 <= '0';
Y4 <= '0';
Y3 <= '0';
Y2 <= '0';
Y1 <= '0';
Y0 <= '0';
wait for 20 ns;

Y7 <= '1';
Y6 <= '0';
Y5 <= '0';
Y4 <= '0';
Y3 <= '0';
Y2 <= '0';
Y1 <= '0';
Y0 <= '0';
wait for 20 ns;
wait;
end process;
end encoder_dataflow_tb;
```

SIMUATION WAVEFORM



You can visit this website to view my code and also run the simulation:

https://www.edaplayground.com/x/MwSg