

Decoders

Decoders are used to decode data that has been previously encoded using a binary, or possibly other, type of coded format. An n -bit code can represent up to 2^n distinct bits of coded information, so a decoder with n inputs can decode up to 2^n outputs. Various models of a 3-8 binary decoder are included in example 3.3.

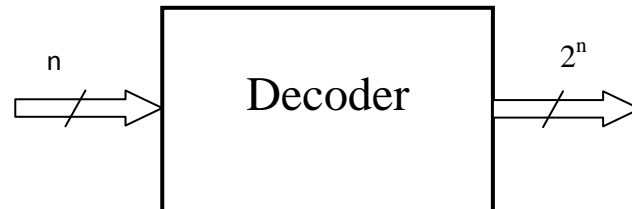


Figure 1 Decoder

Table 1 Truth table of a 3-8 binary decoder

inputs			outputs							
A2	A1	A0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
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

Example 1 Modeling of a 3-8 binary decoder

The models of the 3-8 binary decoders in this example conform to the truth table in Table 1. Different model versions using conditional, selected signal assignments along with typical **if**, and **case** statements. Again, there is no right or wrong modeling technique. The **case** statement is commonly used because of its clarity, and the fact it is not a continuous assignment and so may simulate faster.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
entity decoder3_8 is
port (a: in std_logic_vector(2 downto 0);
y: out std_logic_vector(7 downto 0));
end entity decoder3_8;
architecture first of decoder3_8 is
begin
y <= "00000001" when a = "000" else
"00000010" when a = "001" else
```

```

"00000100" when a = "010" else
"00001000" when a = "011" else
"00010000" when a = "100" else
"00100000" when a = "101" else
"01000000" when a = "110" else
"10000000" when a = "111" else
"00000001";
end architecture first;
architecture second of decoder3_8 is
begin
with a select
y <= "00000001" when "000",
"00000010" when "001",
"00000100" when "010",
"00001000" when "011",
"00010000" when "100",
"00100000" when "101",
"01000000" when "110",
"10000000" when "111",
"00000001" when others;
end architecture second;
architecture third of decoder3_8 is
begin
process (a)
begin
if (a = "000") then y <= "00000001";
elsif (a = "001") then y <= "00000010";
elsif (a = "010") then y <= "00000100";
elsif (a = "011") then y <= "00001000";
elsif (a = "100") then y <= "00010000";
elsif (a = "101") then y <= "00100000";
elsif (a = "110") then y <= "01000000";
elsif (a = "111") then y <= "10000000";
else y <= "00000001";
end if;
end process;
end architecture third;
architecture fourth of decoder3_8 is
begin
process (a)
begin
case a is
when "000" => y <= "00000001";
when "001" => y <= "00000010";
when "010" => y <= "00000100";
when "011" => y <= "00001000";
when "100" => y <= "00010000";
when "101" => y <= "00100000";
when "110" => y <= "01000000";
when "111" => y <= "10000000";
when others => y <= "00000001";

```

```
end case;  
end process;  
end architecture fourth;
```