Decoder circuit design in VHDL

DR. FATMA ELFOULY

Decoder

- 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.

2-4 Decoder

a1	a0	y3	y2	y1	y0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0
d	d	0	0	0	0

2-4 Decoder

- library ieee;
- use ieee.std_logic_1164.all;
- use ieee.std_logic_arith.all;
- entity decoder2_4 is
- port (a: in std_logic_vector(1 downto 0);
- y: out std_logic_vector(3 downto 0));
- end entity decoder2_4;
- architecture rtl of decoder2_4 is
- Begin
- Process(a)
- begin
- If (a = “00”) then y <= “0001” ;
- elsif (a = “01”) then Y<=“0010” ;
- Elsif (a = “10”) then Y<= “0100” ;
- Else Y<= “1000”;
- End if ;
- End process;
- end architecture rtl;

2-4 Decoder

en	a1	a0	y3	y2	y1	y0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0
0	d	d	0	0	0	0

2-4 Decoder

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
entity decoder2_4 is
port (a: in std_logic_vector(1 downto 0);
      en: in std_logic;
y: out std_logic_vector(3 downto 0));
end entity decoder2_4;
architecture rtl of decoder2_4 is
Begin
Process(a,en)
Begin
If (en = '1') then
If (a = "00" ) then y <= "0001" ;
 elsif (a = "01") then Y<="0010" ;
Elsif (a = "10" ) then Y<= "0100" ;
Else Y<= "1000";
End if ;
Else  y<= "0000";
End if;
End process;
```

3-8 Decoder

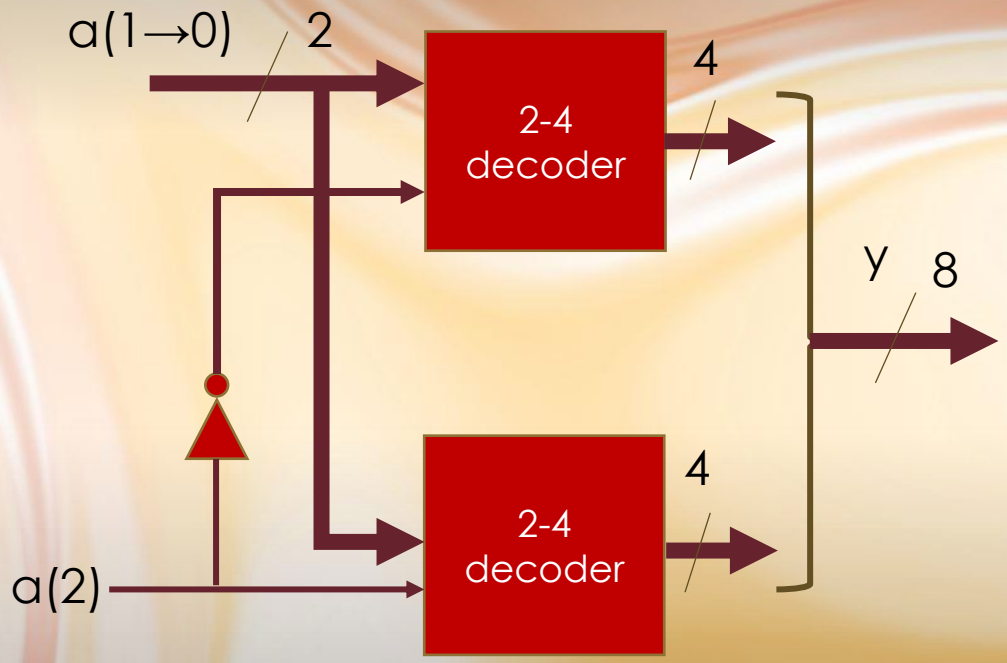
3-8 Decoder

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

Solved Problems

Solved Problems

- *Construct a **3-to-8 decoder** using only **2-to-4 decoders**.*

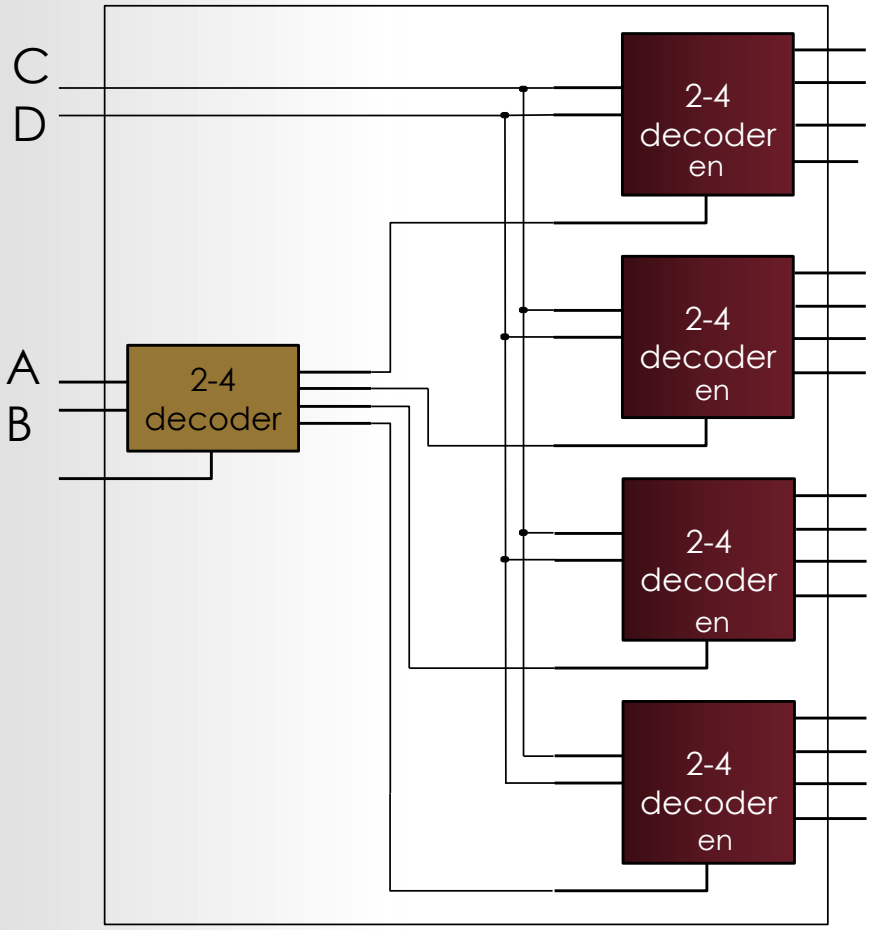


4-16 Decoder

a				Y															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Solved Problems

Construct a 4-to-16 decoder using only 2-to-4 decoders



Solved Problems

Construct a 5-to-32 decoder using only 2-to-4 decoders and 3-to-8 decoders.