7 March

Ex 2: 3-to-8 Decoder

```
Entity decode is
port (A, B, C : in bit;
      F: out bit_vector (7 down to 0));
end;

Architecture flow of decode is
begin
F(0) <= not (A) and not (B) and not (C)
F(1) <= not (A)
  :
```

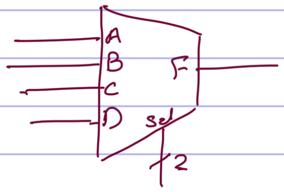
7 March

Sign

لما يكون ال Condition ـه True لـ expression-1 بتعطي لل Signal .

إذا ما تحقق الأول بروح على الثاني .

Mux 4-1

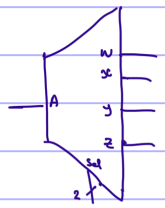


Sel	F
"00"	A
"01"	B
"10"	C
"11"	D

كيف بدنا نبنيه بواسطة الـ logic gates ؟

```
entity mux is
port (A, B, C, D : in bit;
      Sel : in bit_vector (1 down to 0)
      F : out bit) ;
end;

Architecture f1 of mux is
begin
F <= A when sell = "00" else
     B when sell = "01" else
     C when sell = "10" else
     D ;
```

1-to-4 Demultiplexer



Sel	w	x	y	z
"00"	A	0	0	0
"01"	0	A	0	0
"10"	0	0	A	0
"11"	0	0	0	A

W ⇐ A and not Sel (0) and not Sel (1)

x ⇐

y ⇐

z ⇐ S

9-March

Logical operator

Co

* Selected Signal Assignment.

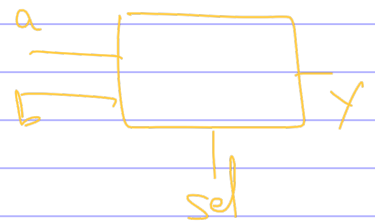
with → بتحددلي ال value اللي بدي أعمل منها Read.

Select → value من بين multiple values
بناءً على شرط معين (when)

Others زي فكرة else . لما ولا شرط يتحقق.

$Y \Leftarrow X;$ → variable.

Y <= a when '0',
b when '1';



ABC

Signal <= A & B & C ⇒ Signal ABC : bit vector (2 down to 0)
MSB LSB

مثل ←
ABC(2) <= A
ABC(1) <= B
ABC(0) <= C

With abc select
F0 <= '1' when "000",
'0' when others;

" أول o/p بالجدول "
(مش متأكد)

⇐ قبل كل جملة شرط (condition) لازم اسبقها بـ with

⇐ متى يستخدم الـ select ؟ لا يوجد أكثر من o/p عند نفس الـ comp.

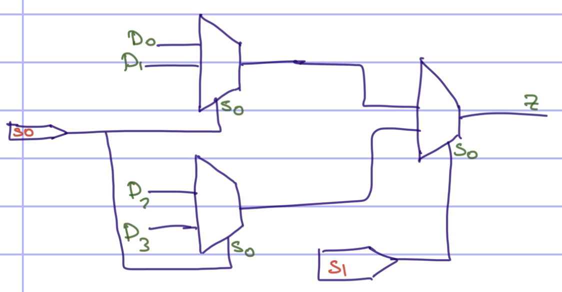
14 March
Tue.

Mux 4 to 1 / Muxes 2 to 1

1st ex: Building (4-1) Mux from 3 (2-1) Muxes

```
entity Mux21 is
port (D0, D1, S0 : in bit;
        O: out bit);
end;

Architecture f1 of Mux21 is
begin
 o <= D0 when S0 = '0' else
      D1 when S0 = '1';
end;


entity Mux41 is
port (a, b, c, d, S0, S1 : in bit)
      Z: out bit);

Architecture f1 of Mux41 is
  component Mux21 is
  port (d0, d1, s0 : in bit
        O ; out bit)
  end;
  begin
    mux1 : Mux21 port map (d0 => a, d1 => b, ~~d2 => sig1~~ s0 => s0, o => Sig1);
    mux2 : Mux21 port map (c, d, S0, Sig2)  ]
    mux3 : Mux21 port map (Sig1, Sig2, S1, Z) ] ?
  end;
```

(*) صار ن مشكلة هون وتعطل فيها شويتنا [30 : 00 تقريباً] .



** هاد اللي حتذكرة مع الدكتورة .

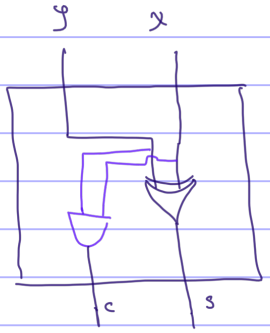
*حل السؤال بطريقة أخرى

```
begin
with So Select
   0 <= do when '0'
      d1 when others.

end.
```

2nd Example: Structural mode of a 4-bit RCA → بالسلايدات
(Ripple Carry Adder)

نفس المثال السابق

16. March.

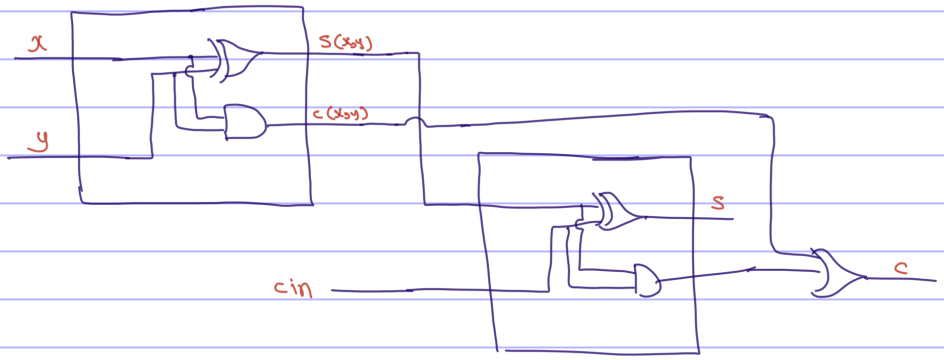


half adder

c	s	x	y
0	0	0	0
0	1	0	1
0	1	1	0
1	1	1	1

S = x xor y

C = x and y



c	s	x	y	cin
0	0	0	0	0
0	1	0	0	1
0	1	0	1	0
1	0	0	1	1
0	1	1	0	0
1	0	1	0	1
1	0	1	1	0
1	1	1	1	1

```
with sel Sel1 ?
Sig <= A & '0' & '0' & '0' when "00"
       '0' & A & '0' & '0' when "01"
```

4 bit

حل آخر لسؤال حليته من قبل بدكم باستخدام with واحدة.

file 1

```
entity HA is
port (in1, in2 : in bit;
      s, c : out bit);
end;

architecture f1 of HA is
begin
s <= in1 xor in2
c <= in1 and in2
end
```

file 2

```
entity Mux21 is
port (in1, in2, in3 : in bit;
      s,c: out bit);
end;

Architecture f1 of FA is
component HA is
port (in1, in2 : in bit;
      s, c : out bit);
End Component;
Signal sig1, sig2, sig3: bit;
begin
HA1: HA port
HA2 : HA port
```

file 3

```
entity RCA is
port (A, B : in bit_vector (3 downto 0);
      s : out bit_vector (3 downto 0);
      c : out bit);
end;
```

21/3/2023

```
begin
  Sign <= in
  out <= Sign
A: process (x,y)
      ⋮
   end;
B: process (x,x)
      ⋮
   end;
end
```

داخل الـ process

If & else (condition) ⟹ if condition then dosomething
end if

موجد بالسيرفيرات

if ... then ...
else dosmthing

Compenational : أي تغيّر في الـ I/P يغيّر الـ o/p

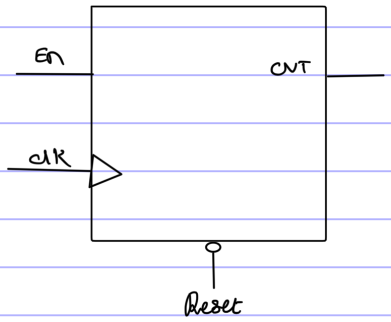
28 March

Ch.7

TB → test bench

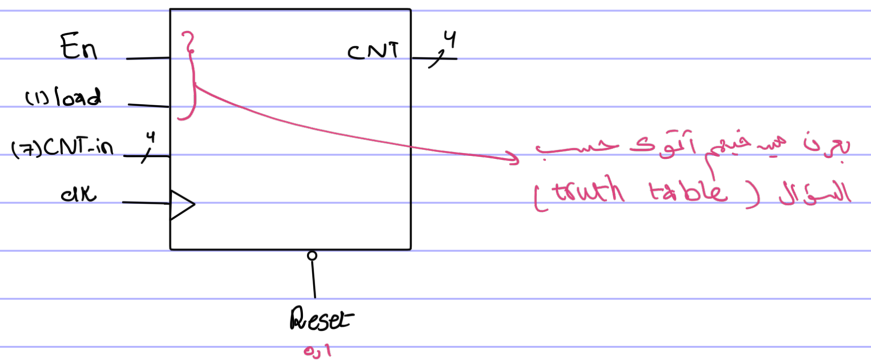
30-4

Synthesis
يفيد لأشياء
على شكل بلوك.
process → X




```
Library IEEE
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;
  entity Counter is
    port (En, clk, R: in std_logic;
          CNT: out std_logic_vector (3 down to 0));
  Architecture A1 of counter is
    Signal S: integer range 0 to 15;
    begin
    proc: process (clk, R)
      if (R='0') then S=0;
      elif (rising_edge(clk)) then
          if (En='1') then
            if (S=15) then S=0;
            else S=S+1
      end if; end if; end if;
  end proc;

  CNT <= std_logic_vector(to_unsigned(S,4));
end;
```

هيئ متأكد من هيك هيك



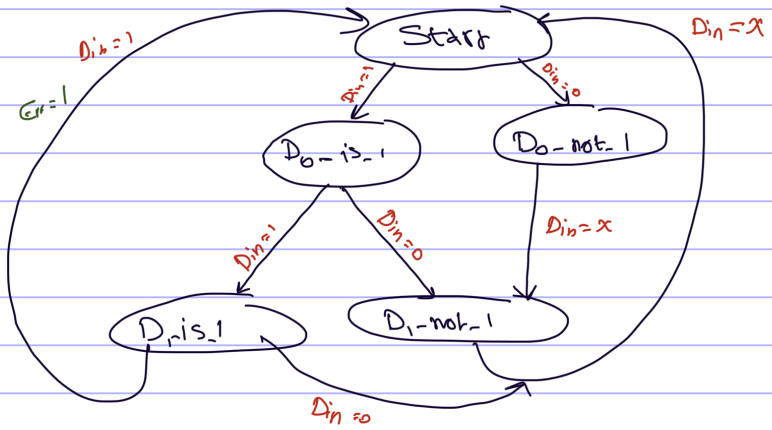
بدن هيك فيهم أكتب حسب السؤال (truth table)

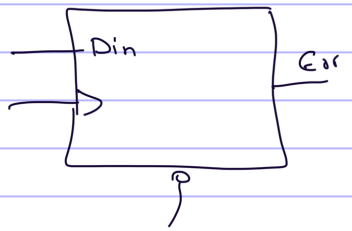
تطبيق على counter الـ

Finite State Machine الـ

```
Library IEEE
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;
   entity Counter is
     port (En, clk, R: in std_logic;
          CNT: out std_logic_vector (3 down to 0);
          CNT: in std_logic_vector (3 down to 1));
   Architecture A1 of counter is
     Signal S: integer range 0 to 15;
     begin
     proc: process (clk, R)
       if (R = '0') then S=0;
       elif (rising_edge (clk)) then
            if (L = '1') then
            S <= toInteger(Unsigned(CNT_INT));
            elif (En = '1') then
               if (S=15) then S=0;
               else S=S+1
       end if; end if; end if;
     end proc;

     CNT <= std_logic_vector(toUnsigned(S,4)) ;
     end;
```

14-May






```
entity
  :
end entity
```

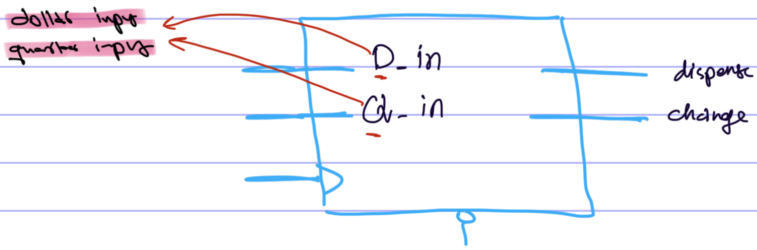
```
Architecture A1 of bit_detector is
type State is
  :
begin
memory_state : process(clk, R)
begin
  if (R='0') then Err <= '0';
  elsif (rising_edge(clk)) then
     current_state = next_state;
  end if;
  end process;

  next_State_logic : process (current_state, Din)
  begin
  case (current_state) is
  when state => if (Din='1') then
                  next_state <= D0_is_1
                  else next_state <= D0_not_1
                  end if;
when D0_not_1 => next_state <= D1_not_1
when D1_not_1 => next_state <= state
when D0_is_1  => if (Din='1') then
                   next_state <= D1_is_1
                   else next_state <= D1_not_1
                   end if
```

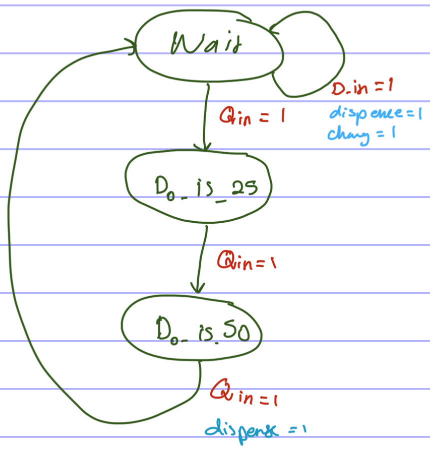
```
when D_is_1 => next_state <= State
when other => next_state <= Start
end case;
end process;

output_logic : process (current_state , Din)
begin
Case ( current_State ) is
   when D_is_1 => if (Din = 1) then
                     Err = '1';
                  else Err = '0';
                  end if
 when others => Err <= '0'
 end case;
end process;
end;
```

* عدد mini
max

* عدد ال States
ربط ال states

Ex: vinding machine





* حل الكتاب

ال simplest case.