

تصميم رقمي متقدم Advanced Digital Design

Dr.-Eng. Samer Sulaiman

2021-2022

- أساسيات التصميم الرقمي
- عناصر وتقنيات التصميم الرقمي التوافقي والتعاقبي (المتسلسل)
- **نمذجة التصميم الرقمي باستعمال لغة توصيف الكيان الصلب VHDL**
- المحاكاة الوظيفية والزمنية للأنظمة الرقمية

تصميم الأنظمة الرقمية باستخدام VHDL



• أساسيات البرمجة بلغة ال VHDL:

• طريقة كتابة شيفرة (كود) ال VHDL:

- في لغة VHDL إذا وضع (- -) قبل سطر التعليمة فهذا يدل على أن السطر هو تعليقات أو ملاحظات.
- يجب أن تنتهي جميع أسطر التعليمات والتصريحات بفاصلة منقوطة (;)
- عند التصريح عن أكثر من متحول تستخدم الفاصلة للفصل بينهم (,)
- إسناد قيمة ما إلى إشارة رمزها (<=)، بينما إسناد قيمة ما إلى متحول والتصريح عن قيمة ثابتة ولتعيين قيمة ابتدائية رمزها (:=)

يمكن استخدام الأرقام والحروف و() للتصريح عن أسماء المتحولات والإشارات

- لا يمكن أن يبدأ الاسم برقم ، وكذلك لا يمكن استخدام كلمة محجوزة في الاسم .

• لغة VHDL غير حساسة لحالة الحروف (صغيرة أو كبيرة)

- خلافاً لبرامج الحاسوب العادية التي تكون متسلسلة التنفيذ فإن لغة ال VHDL تنفذ التعليمات على التفرع لذلك نقول VHDL code بدلا من VHDL program

• العمليات الأساسية المتوفرة في لغة VHDL

	Operator Class	Operator
Highest precedence	Miscellaneous	**, ABS, NOT
	Multiplying	*, /, MOD, REM
	Sign	+, -
	Adding	+, -, &
	Shift	SLL, SRL, SLA, SRA, ROL, ROR
	Relational	=, /=, <, <=, >, >=
Lowest precedence	Logical	AND, OR, NAND, NOR, XOR, XNOR

تصميم الأنظمة الرقمية باستخدام VHDL



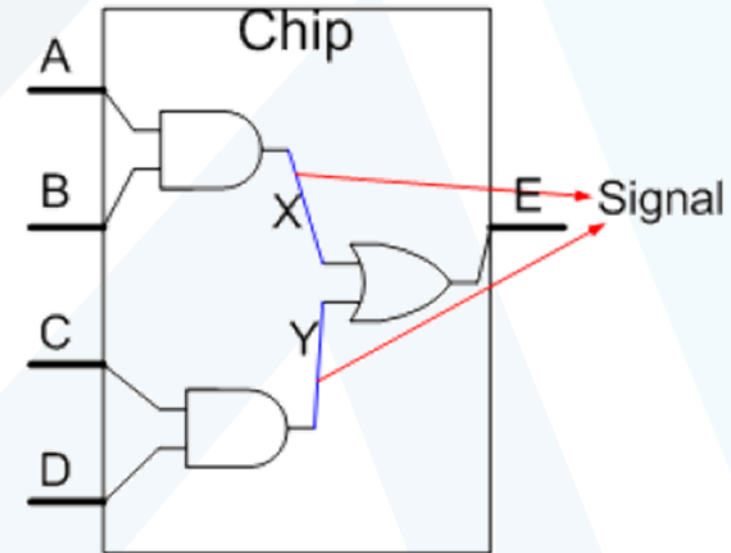
- أساسيات البرمجة بلغة ال VHDL:
- طريقة كتابة شيفرة (كود) ال VHDL:
- التصريح عن الإشارات ضمن لغة ال VHDL:
- الإشارات عبارة عن المتحولات الداخلية التي تستخدم لإيصال خرج أحد المكونات الداخلية إلى دخل مكون داخلي آخر
- يتم ذلك كما يلي:
- أمثلة عن التصريح عن إشارات مختلفة:
- ```
SIGNAL a : STD_LOGIC;
SIGNAL b : STD_LOGIC_VECTOR(7 DOWNTO 0);
```
- ```
SIGNAL a: STD_LOGIC;  
SIGNAL b: STD_LOGIC_VECTOR(3 DOWNTO 0);  
SIGNAL c: STD_LOGIC_VECTOR(3 DOWNTO 0);  
SIGNAL d: STD_LOGIC_VECTOR(15 DOWNTO 0);  
SIGNAL e: STD_LOGIC_VECTOR(8 DOWNTO 0);  
a <= '1';  
b <= "0000"; -- Binary base assumed by default  
c <= B"0000"; -- Binary base explicitly specified  
d <= X"AF67"; -- Hexadecimal base  
e <= O"723"; -- Octal base
```

تصميم الأنظمة الرقمية باستخدام VHDL



- أساسيات البرمجة بلغة ال VHDL:
- طريقة كتابة شيفرة (كود) ال VHDL:
- التصريح عن الإشارات ضمن لغة ال VHDL:
- مثال أكتب شيفرة ال VHDL للدارة المنطقية التالية:

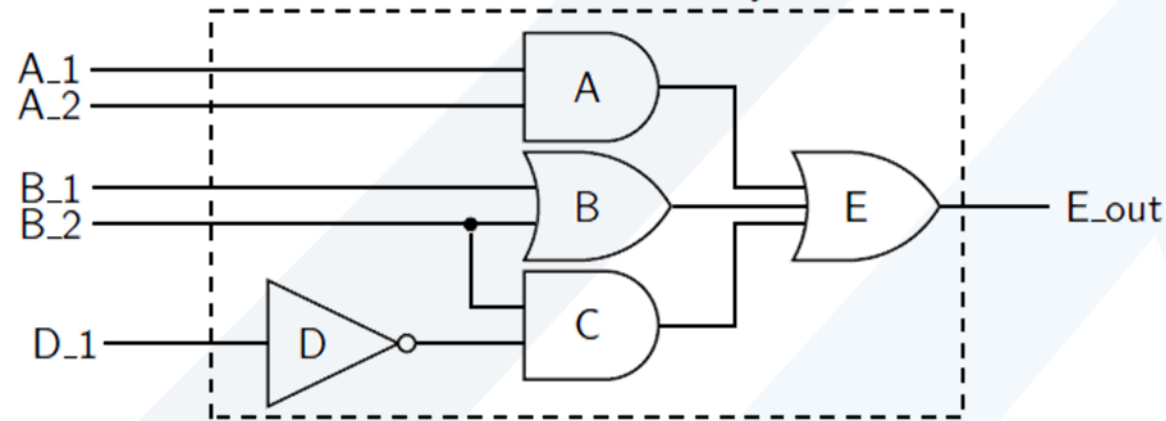
```
• LIBRARY IEEE;
  USE IEEE.STD_LOGIC_1164.ALL;
  ENTITY TEST IS
    PORT (A,B,C,D : IN STD_LOGIC;
          E : OUT STD_LOGIC);
  END TEST;
  ARCHITECTURE BEHAVIOR OF TEST IS
    SIGNAL X,Y : STD_LOGIC;
  BEGIN
    X <= (not A) AND B;
    Y <= C AND D;
    E <= X OR Y;
  END BEHAVIOR;
```



تصميم الأنظمة الرقمية باستخدام VHDL



- أساسيات البرمجة بلغة الـ VHDL:
- طريقة كتابة شيفرة (كود) الـ VHDL:
- التصريح عن الإشارات ضمن لغة الـ VHDL:
- مثال أكتب شيفرة الـ VHDL للدارة المنطقية التالية:



تصميم الأنظمة الرقمية باستخدام VHDL



- أساسيات البرمجة بلغة ال VHDL:
- طريقة كتابة شيفرة (كود) ال VHDL:
- التصريح عن الإشارات ضمن لغة ال VHDL:
- مثال أكتب شيفرة ال VHDL للدائرة المنطقية التالية:

```
entity my_circuit is
    port (
        A_1,A_2,B_1,B_2,D_1 : in std_logic;
        E_out : out std_logic);
end my_circuit;
architecture my_circuit_arc of my_circuit is
    signal A_out, B_out, C_out : std_logic;
begin
    A_out <= A_1 and A_2;
    B_out <= B_1 or B_2;
    C_out <= (not D_1) and B_2;
    E_out <= A_out or B_out or C_out;
end my_circuit_arc;
```

يمكن كتابتها ايضاً بالشكل التالي بدون الحاجة
إلى التصريح عن الاشارات:

```
E_out <= (A_1 and A_2) or (B_1 or B_2)
or ((not D_1) and B_2);
```

تصميم الأنظمة الرقمية باستخدام VHDL



- أساسيات البرمجة بلغة ال VHDL:

- طريقة كتابة شيفرة (كود) ال VHDL:

- مثال: أكتب شيفرة ال VHDL لدارة جامع كامل بخانة واحدة

- المرحلة الأولى:

- توصيف النظام على أنه صندوق أسود يمتلك عدداً من المداخل والمخارج بمواصفات محددة .

- إن الدارة الرقمية للجامع الكامل عبارة عن دارة ذات ثلاثة مداخل ومخرجين:

- دخل أول: A، دخل ثاني: B، دخل ثالث: الحمل . Cin

- خرج أول: يمثل المجموع $A + B = \text{Sum}$ ، خرج ثاني: يمثل الحمل الناتج . Cout

- جدول الحقيقة للجامع الكامل:

A	B	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

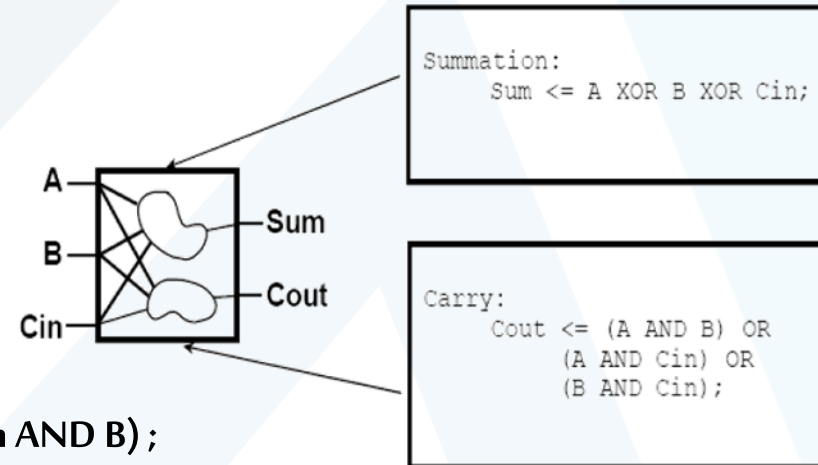
- المرحلة الثانية:

- توصيف النظام من الداخل من خال توصيف العلاقة التي تربط المداخل بالمخارج اعتماداً على العلاقات المنطقية المستنتجة من جدول الحقيقة

تصميم الأنظمة الرقمية باستخدام VHDL



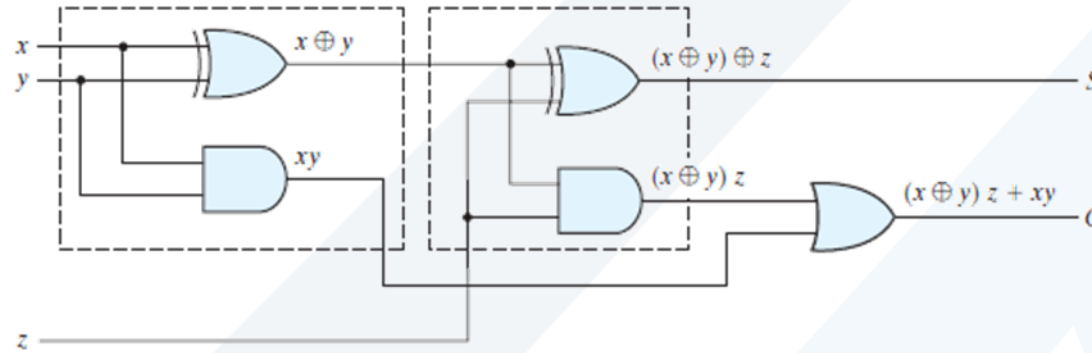
- أساسيات البرمجة بلغة ال VHDL:
 - طريقة كتابة شيفرة (كود) ال VHDL:
 - مثال: أكتب شيفرة ال VHDL لدارة جامع كامل بخانة واحدة
 - المرحلة الثانية:
 - الشيفرة المطلوبة:
- ```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
ENTITY full_adder IS
 PORT (A,B,cin: IN STD_LOGIC ;
 Sum, cout : OUT STD_LOGIC) ;
END full_adder;
ARCHITECTURE dataflow OF full_adder IS
BEGIN
 Sum <= A XOR B XOR cin ;
 cout <= (A AND B) OR (cin AND A) OR (cin AND B);
END dataflow ;
```



# تصميم الأنظمة الرقمية باستخدام VHDL



- أساسيات البرمجة بلغة ال VHDL:
- طريقة كتابة شيفرة (كود) ال VHDL:
- وظيفة: أكتب شيفرة ال VHDL لدارة جامع كامل بخانة واحدة علماً أن الدارة المنطقية المكافئة له موضحة بالشكل التالي:



# تصميم الأنظمة الرقمية باستخدام VHDL



- طريقة التنفيذ في اللغة VHDL:
  - على خلاف لغات البرمجة التقليدية يتم تنفيذ التعليمات في لغة VHDL على التوازي (Concurrent (parallel)
    - يتم تنفيذ الكود البرمجي بنفس الوقت تقريباً لكافة التعليمات
    - يمكن القول ان إشارات الدخل تنقل إلى الخرج بزمن مهملاً تقريباً باستثناء تأخير الانتشار
  - نظراً لهذه الخصوصية في تنفيذ التعليمات ونظراً للحاجة في بعض الحالات من تحويل التنفيذ إلى النمط التقليدي أي ترتيب تنفيذ التعليمات بحسب حاجة المبرمج وما يراه من ضرورة في عمل النظام
    - تم إيجاد مجموعة تعليمات كتعليمات Process .. End process التي يمكن إضافتها في وحدة التوصيف الداخلي ليتم تنفيذ التعليمات التي تذكر بعدها بترتيب ورودها من الأعلى إلى الأسفل
    - مما يؤدي إلى تحويل آلية التنفيذ إلى النمط التتابعي Sequential
  - من أنماط التعليمات التي يتم تنفيذها بصورة تسلسلية نذكر PROCESS, FUNCTION, PROCEDURE، حيث يتم تنفيذ ما بداخل هذه الكتل بشكل تسلسلي
  - لكن على مستوى الكتلة نفسها فإنها تنفذ تفرعياً شأنها شأن أي تعليمات في VHDL، وبالتالي تكون مهمتها تحقيق الترتيب أو التزامن في تنفيذ التعليمات
  - تسمى بـ GUARDED BLOCK، أو محطات الانتظار التي يكون من مهامها الأساسية منع الانتقال إلى كتلة برمجية أخرى حتى يتحقق شرط ما يكون شرط التزامن مثلاً

# تصميم الأنظمة الرقمية باستخدام VHDL



- تعليمات البنية التفرعية في اللغة VHDL:

- عمليات الإسناد ( $\leq$ )

- العمليات المنطقية.

- العمليات الحسابية.

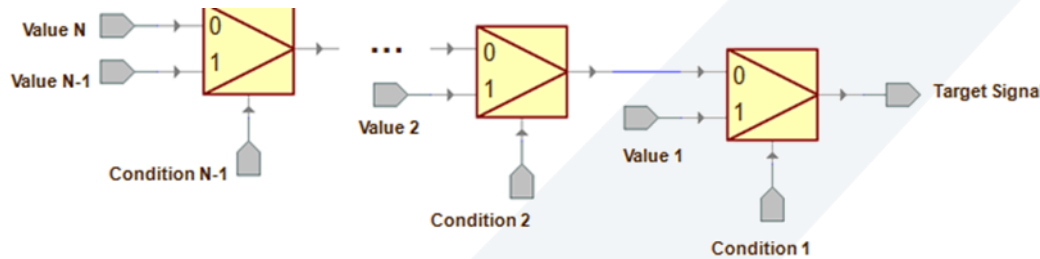
- تعليمة when بأشكالها (when-else و with-select-when)

- تعليمة Generate

- تعليمة when:

- تأخذ عدة أشكال:

- الشكل الأول When – Else



- ```
target_signal <= value1 when condition1 else
value2 when condition2 else
... valueN-1 when conditionN-1 else
valueN;
```

تصميم الأنظمة الرقمية باستخدام VHDL



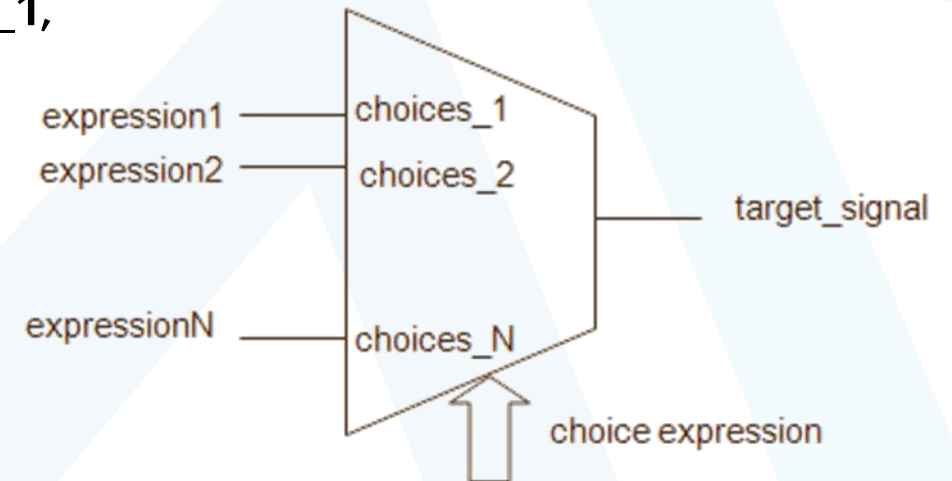
- تعليمات البنية التفرعية في اللغة VHDL:

- تعليمة when:

- تأخذ عدة أشكال:

- الشكل الثاني With –Select-When:

- with choice_expression select
target_signal <= expression1 when choices_1,
expression2 when choices_2,
... expressionN when choices_N;



تصميم الأنظمة الرقمية باستخدام VHDL

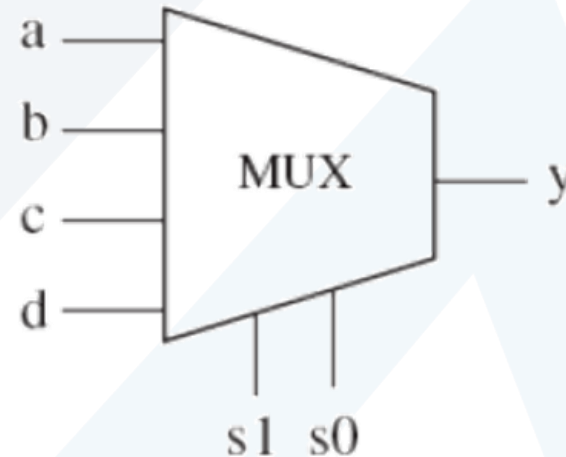


- تعليمات البنية التفرعية في اللغة VHDL:

- تعليمة when:

- مثال: أكتب شيفرة الـ VHDL لدارة الناخب المبينة في الشكل:

- ```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY example IS
PORT (a, b, c, d, s0, s1: IN STD_LOGIC;
y: OUT STD_LOGIC);
END example;
ARCHITECTURE Mux OF example IS
BEGIN
y <= (a AND NOT s1 AND NOT s0) OR
(b AND NOT s1 AND s0) OR
(c AND s1 AND NOT s0) OR
(d AND s1 AND s0);
END Mux;
```



| input |   | output |
|-------|---|--------|
| s     |   | x      |
| 0     | 0 | a      |
| 0     | 1 | b      |
| 1     | 0 | c      |
| 1     | 1 | d      |

# تصميم الأنظمة الرقمية باستخدام VHDL



- تعليمات البنية التفرعية في اللغة VHDL:

- تعليمة when:

- مثال: أكتب شيفرة ال VHDL لدارة الناخب المبينة في الشكل:

- الشكل الأول

- ```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY example IS
PORT ( a, b, c, d, s0, s1: IN STD_LOGIC;
y: OUT STD_LOGIC);
END example;
ARCHITECTURE Mux OF example IS
BEGIN
y <= (a AND NOT s1 AND NOT s0) OR
(b AND NOT s1 AND s0) OR
(c AND s1 AND NOT s0) OR
(d AND s1 AND s0);
END Mux;
```

- ```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY mul IS
PORT (a, b, c, d: IN STD_LOGIC;
s: IN STD_LOGIC_VECTOR (1 DOWNT0 0);
y: OUT STD_LOGIC);
END mul;
ARCHITECTURE mux OF mul IS
BEGIN
y <= a WHEN s="00" ELSE
b WHEN s="01" ELSE
c WHEN s="10" ELSE
d;
END mux;
```

# تصميم الأنظمة الرقمية باستخدام VHDL



- تعليمات البنية التفرعية في اللغة VHDL:

- تعليمة when:

- مثال: أكتب شيفرة الـ VHDL لدارة الناخب المبينة في الشكل:

- الشكل الثاني

- ```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY example IS
PORT ( a, b, c, d, s0, s1: IN STD_LOGIC;
y: OUT STD_LOGIC);
END example;
ARCHITECTURE Mux OF example IS
BEGIN
y <= (a AND NOT s1 AND NOT s0) OR
(b AND NOT s1 AND s0) OR
(c AND s1 AND NOT s0) OR
(d AND s1 AND s0);
END Mux;
```

- .

- ```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY mul IS
PORT (a, b, c, d: IN STD_LOGIC;
s: IN STD_LOGIC_VECTOR (1 DOWNT0 0);
y: OUT STD_LOGIC);
END mul;
ARCHITECTURE mux OF mul IS
BEGIN
WITH s SELECT
y <= a WHEN "00", -- notice "," instead of ";"
b WHEN "01",
c WHEN "10",
d WHEN OTHERS; -- cannot be "d WHEN "11" "
END mux;
```



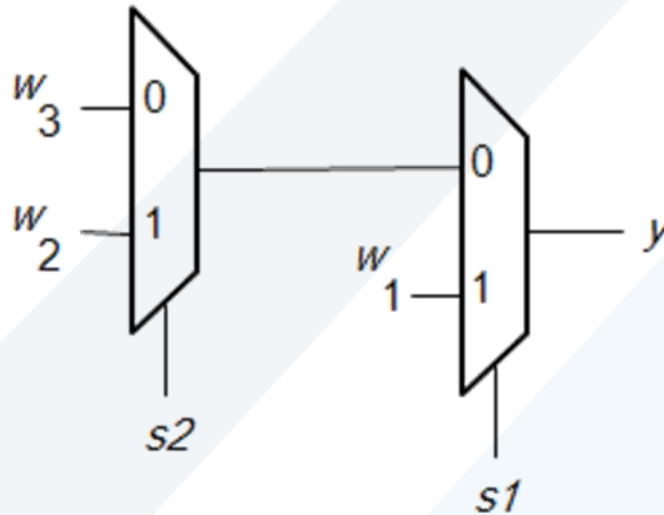
# تصميم الأنظمة الرقمية باستخدام VHDL



- تعليمات البنية التفرعية في اللغة VHDL:

- تعليمة when:

- وظيفة: أكتب شيفرة الـ VHDL لتوصيف النظام المبين بالشكل التالي:



# تصميم الأنظمة الرقمية باستخدام VHDL

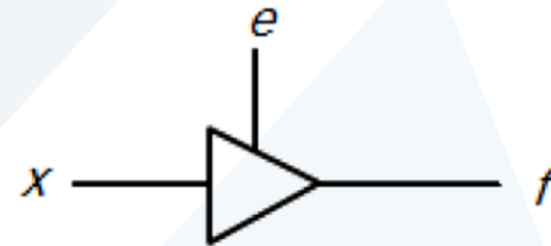


• تعليمات البنية التفرعية في اللغة VHDL:

• تعليمة when:

• مثال: أكتب شيفرة ال VHDL لتوصيف دائرة عازل ثلاثي الحالة بمدخل تمكين واحد:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY tri_state IS
PORT (ena: IN STD_LOGIC;
 input: IN STD_LOGIC;
 output: OUT STD_LOGIC);
END tri_state;
ARCHITECTURE dataflow OF tri_state IS
BEGIN
 output <= input WHEN (ena = '1') ELSE 'Z';
END dataflow;
```



A tri-state buffer

| <i>e</i> | <i>x</i> | <i>f</i> |
|----------|----------|----------|
| 0        | 0        | Z        |
| 0        | 1        | Z        |
| 1        | 0        | 0        |
| 1        | 1        | 1        |

# تصميم الأنظمة الرقمية باستخدام VHDL



- تعليمات البنية التفرعية في اللغة VHDL:

- تعليمة when:

- مثال: المطلوب تصميم دائرة عازل لثمانية خانات بمدخلين متعاكسين:

- ```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY tri IS
PORT ( E: IN STD_LOGIC_VECTOR(1 downto 0);
      input: IN STD_LOGIC_VECTOR (7 DOWNTO 0);
      Output: OUT STD_LOGIC_VECTOR (7 DOWNTO 0));
END tri ;
ARCHITECTURE tri_state OF tri IS
BEGIN
  with E select
    output <= input WHEN "01" | "10",
    "ZZZZZZZZ" when others;
END tri_state;
```

تصميم الأنظمة الرقمية باستخدام VHDL



• تعليمات البنية التفرعية في اللغة VHDL:

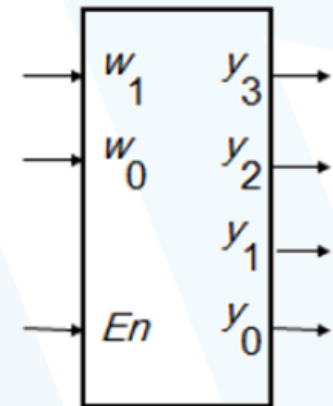
• تعليمة when:

• مثال: أكتب شيفرة الـ VHDL لتوصيف دائرة المشفر الموضحة بالشكل:

```

• LIBRARY ieee;
  USE ieee.std_logic_1164.all;
  ENTITY dec2to4 IS
    PORT (w: IN STD_LOGIC_VECTOR(1 DOWNTO 0);
          En: IN STD_LOGIC;
          y: OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
  END dec2to4;
  ARCHITECTURE dataflow OF dec2to4 IS
    SIGNAL Enw : STD_LOGIC_VECTOR(2 DOWNTO 0);
  BEGIN
    Enw <= En & w;
    WITH Enw SELECT
      y <= "0001" WHEN "100",
          "0010" WHEN "101",
          "0100" WHEN "110",
          "1000" WHEN "111",
          "0000" WHEN OTHERS;
  END dataflow;
  
```

En	w_1	w_0	y_3	y_2	y_1	y_0
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	x	x	0	0	0	0



تصميم الأنظمة الرقمية باستخدام VHDL

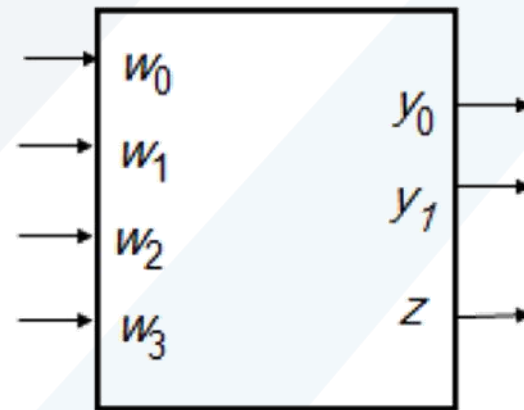


- تعليمات البنية التفرعية في اللغة VHDL:

- تعليمة when:

- مثال: أكتب شيفرة ال VHDL لتوصيف دارة فاك تشفير الأولوية الموضحة بالشكل:

- ```
LIBRARY ieee ;
USE ieee.std_logic_1164.all;
ENTITY priority IS
PORT (w : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
 y : OUT STD_LOGIC_VECTOR(1 DOWNTO 0);
 z : OUT STD_LOGIC);
END priority;
ARCHITECTURE dataflow OF priority IS
BEGIN
 y <= "11" WHEN w(3) = '1' ELSE
 "10" WHEN w(2) = '1' ELSE
 "01" WHEN w(1) = '1' ELSE "00";
 z <= '0' WHEN w = "0000" ELSE '1';
END dataflow
```



| w <sub>3</sub> | w <sub>2</sub> | w <sub>1</sub> | w <sub>0</sub> | y <sub>1</sub> | y <sub>0</sub> | z |
|----------------|----------------|----------------|----------------|----------------|----------------|---|
| 0              | 0              | 0              | 0              | d              | d              | 0 |
| 0              | 0              | 0              | 1              | 0              | 0              | 1 |
| 0              | 0              | 1              | x              | 0              | 1              | 1 |
| 0              | 1              | x              | x              | 1              | 0              | 1 |
| 1              | x              | x              | x              | 1              | 1              | 1 |