



تصميم رقمي متقدم  
Advanced Digital Design

جامعة  
المنارة  
HAMARA UNIVERSITY

Dr.-Eng. Samer Sulaiman

2020-2021

# مفردات المنهاج

- 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

## • تصميم بنى منطقية تسلسلية ضمن اللغة VHDL:

• أهم التعليمات المستخدمة مع نبضات الساعة

- تسمى أيضاً بـ Clock Functions وهي موجودة في مكتبة الـ iee ضمن الـ "std\_logic\_1164" package
- التعليمة event: حيث تأخذ الشكل التالي Clk'event وتأخذ القيمة true عند حدوث اي تغير في إشارة نبضات الساعة
- التعليمة rising\_edge(Clk): تأخذ القيمة true عند تغير إشارة نبضات الساعة إلى النبضة الصاعدة
- تعتبر التعليمتين التاليتين متكافئتان (if(Clk'event and Clk = '1') و if rising\_edge(Clk)
- التعليمة falling\_edge(Clk): تأخذ القيمة true عند تغير إشارة نبضات الساعة إلى النبضة الهابطة
- يمكن استخدام التعليمة (الكلمة المحجوزة) OTHERS لإسناد قيم إلى شعاع
- مثال:

- Q <= "00000001"  
Q <= (0 => '1', OTHERS => '0')
- Q <= "10000001"  
Q <= (7 => '1', 0 => '1', OTHERS => '0') or Q <= (7 | 0 => '1', OTHERS => '0')
- Q <= "00011110"  
Q <= (4 downto 1 => '1', OTHERS => '0')

• تستخدم التعليمة & في لغة الـ VHDL لجمع سلسلتين مع بعضهما  
• مثال:

- بفرض أن A="110" و B="111"
- فيكون ناتج التعليمة التالية:

- C <= A & B -- C = "110111"

• تستخدم التعليمة | ضمن شرط الاختيار لدمج مجموعة من الشروط التي تعطي نفس النتيجة

# تصميم الأنظمة الرقمية باستخدام VHDL

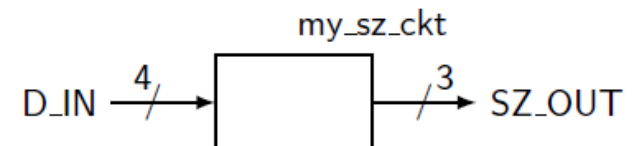
## • تصميم بنى منطقية تسلسلية ضمن اللغة VHDL:

• أهم التعليمات المستخدمة مع نبضات الساعة

• مثال: المطلوب تصميم دارة تحتوي على أربع مداخل وخرج من ثلاث إشارات. العلاقة بين إشارة الدخل والخرج موضحة في الجدول:

```
entity my_sz_ckt is
  port ( D_IN : in std_logic_vector(3 downto 0);
        SX_OUT : out std_logic_vector(2 downto 0));
end my_sz_ckt;
architecture spec_dec of my_sz_ckt is
begin
  with D_IN select
    SX_OUT <= "100" when "0000" | "0001" | "0010" | "0011",
              "010" when "0100" | "0101" | "0110" | "0111" | "1000" | "1001",
              "001" when "1010" | "1011" | "1100" | "1101" | "1110" | "1111",
              "000" when others;
end spec_dec;
```

range of D_IN	SZ_OUT
0000 → 0011	100
0100 → 1001	010
1001 → 1111	001
unknown value	000



# تصميم الأنظمة الرقمية باستخدام VHDL

• تصميم بنى منطقية تسلسلية ضمن اللغة VHDL:

• أهم التعليمات المستخدمة مع نبضات الساعة

• مثال: المطلوب تصميم دائرة تابعها المنطقي معطى بالعلاقة التالية:

$$F3 = \overline{LMN} + LM$$

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity my_ckt_f3 is
    port ( L,M,N : in std_logic;
          F3 : out std_logic);
end my_ckt_f3;
architecture f3_2 of my_ckt_f3 is
begin
    F3<=((NOT L)AND(NOT M)AND N)OR(L AND M);
end f3_2;
```



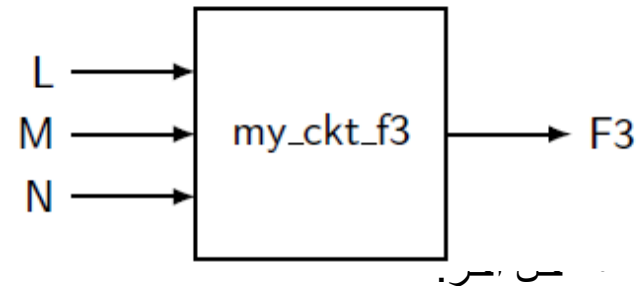
# تصميم الأنظمة الرقمية باستخدام VHDL

• تصميم بنى منطقية تسلسلية ضمن اللغة VHDL:

• أهم التعليمات المستخدمة مع نبضات الساعة

• مثال: المطلوب تصميم دائرة تابعها المنطقي معطى بالعلاقة التالية:  $F3 = \overline{LMN} + LM$  باستخدام تعليمة when

```
• architecture f3_3 of my_ckt_f3 is
begin
  F3 <= '1' when (L='0' AND M='0' AND N='1') else
    '1' when (L='1' AND M='1') else
    '0';
end f3_3;
```



```
• architecture f3_8 of my_ckt_f3 is
  signal t_sig : std_logic_vector(2 downto 0); -- local bundle
begin
  t_sig <= (L & M & N); -- concatenation operator
  with (t_sig) select
    F3 <= '1' when "001" | "110" | "111",
    '0' when others;
end f3_8;
```

# تصميم الأنظمة الرقمية باستخدام VHDL

• تصميم بنى منطقية تسلسلية ضمن اللغة VHDL:

• التعليمة Process:

- إن تعليمة Process من التعليمات التفرعية في لغة VHDL
- تشير إلى أن كل ما يكتب ضمنها يتم تنفيذ بصورة تتابعية وعادة ما تكتب التعليمات IF, WAIT, CASE, LOOP والتي تنفذ بصورة تسلسلية من الأعلى إلى الأسفل وبصورة دورية مادامت تعليمة Process فعالة
- تتضمن تعليمة Process عدد من المتحولات أو الشروط المنطقية تسمى قائمة الحساسية sensitivity list والتي تعتبر مجموعة من المتحولات لا يتم الدخول إلى كتلة التعليمة Process إلا عند حدوث تغيير على هذه المتحولات
- تكتب تعليمة Process في الكود الرئيسي وتنفذ كل مرة يتغير فيها أحد متحولات الحساسية أو الشروط المنطقية
- الصيغة العامة لهذه التعليمة مع ملاحظة أن كل ما يكتب ضمن [ ] هو اختياري:

- [label:] PROCESS (sensitivity list)  
[VARIABLE name type [range] [:= initial\_value;]]  
BEGIN  
(sequential code)  
END PROCESS [label];

# تصميم الأنظمة الرقمية باستخدام VHDL

• تصميم بنى منطقية تسلسلية ضمن اللغة VHDL:

• التعليمة Process:

• مثال: المطلوب تصميم دائرة XOR تعطي خرج في حال تغير احدى مداخلها

```
• library IEEE;
  use IEEE.STD_LOGIC_1164.ALL;
  entity my_xor is
    port ( A,B : in std_logic;
          F : out std_logic);
  end my_xor;
  architecture behav of my_xor is
  begin
    xor_proc: process(A,B) is
    begin
      F <= A XOR B;
    end process xor_proc;
  end behav;
```





# تصميم الأنظمة الرقمية باستخدام VHDL

• تصميم بنى منطقية تسلسلية ضمن اللغة VHDL:

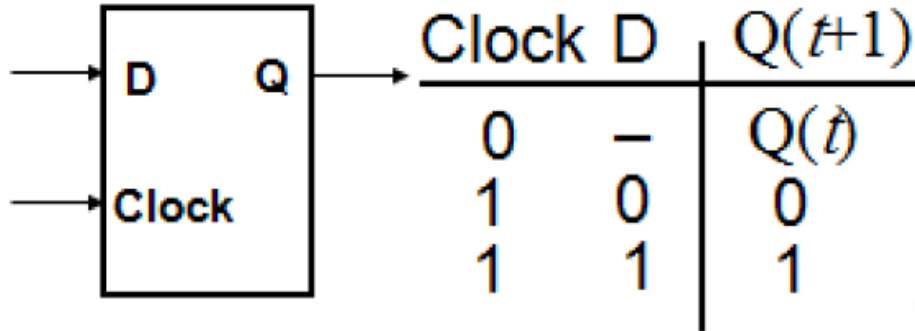
• التعليمة IF:

- أيضاً من التعليمات التتابعية التي تستخدم في حالات التفريع المشروط للبرنامج ،
- تستخدم ضمن تعليمة Process
- الصيغة العامة لها من الشكل :

- IF conditions THEN assignments;  
ELSIF conditions THEN assignments;  
..  
ELSE assignments;  
END IF;

- و من أهم التطبيقات والدارات التي توضح عمل هذه التعليمات هي المسجلات و العدادات
- أمثلة:

# تصميم الأنظمة الرقمية باستخدام VHDL



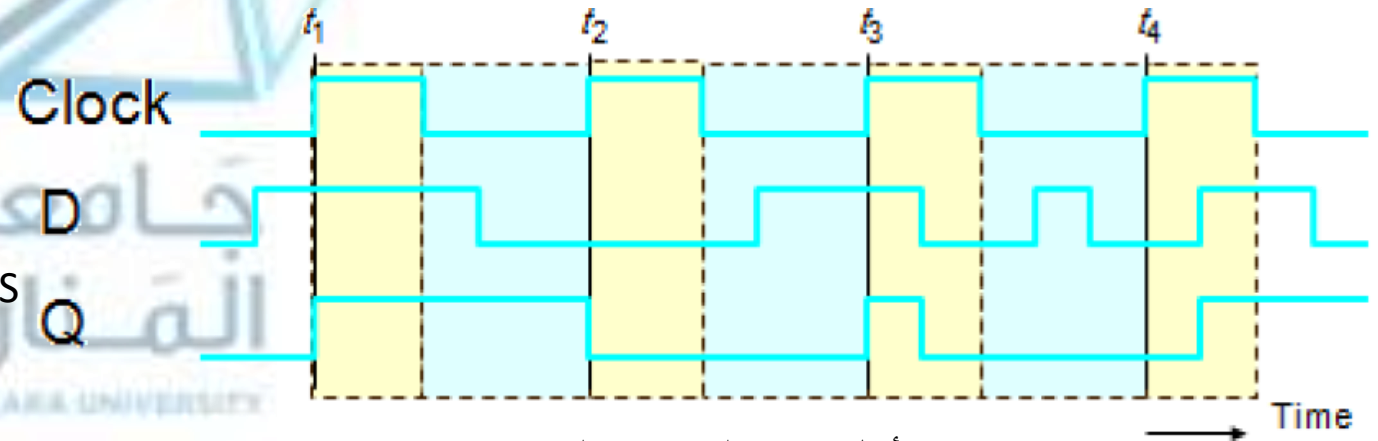
• تصميم بنى منطقية تسلسلية ضمن اللغة VHDL:

• التعليمة IF:

• مثال: المطلوب تصميم دائرة الماسك (D latch) الموضحة بالشكل التالي:

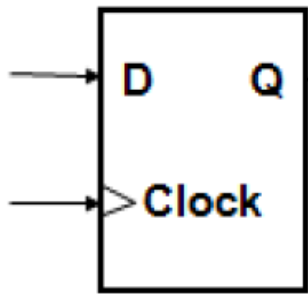
```

LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY latch IS
    PORT (D, Clock: IN STD_LOGIC;
          Q: OUT STD_LOGIC);
END latch;
ARCHITECTURE behavioral OF latch IS
BEGIN
    PROCESS (D, Clock)
    BEGIN
        IF Clock = '1' THEN
            Q <= D;
        END IF;
    END PROCESS;
END behavioral;
    
```



من أجل تحديد التغير في النبضة ('1') - - if(Clock'event and Clock='1')

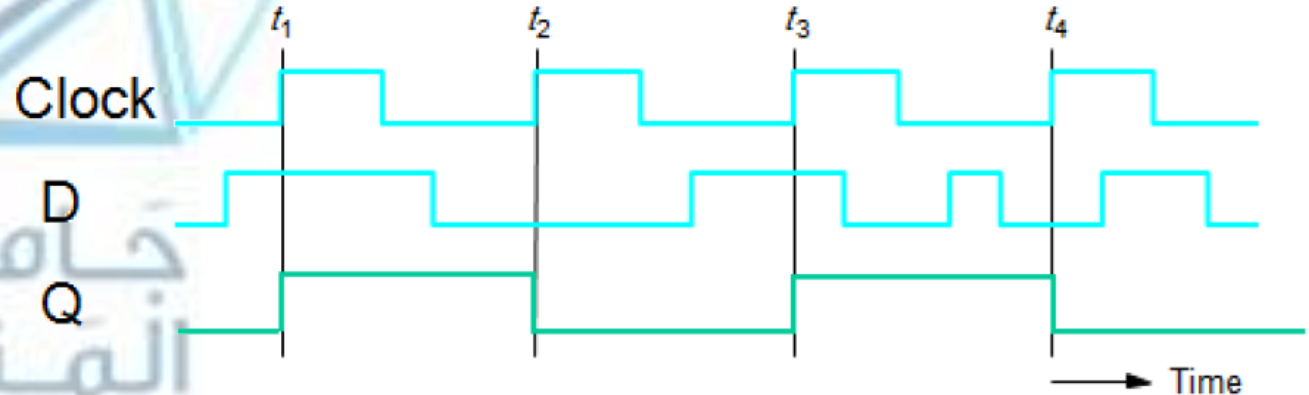
# تصميم الأنظمة الرقمية باستخدام VHDL



Clk	D	Q(t+1)
↑	0	0
↑	1	1
0	—	Q(t)
1	—	Q(t)

- تصميم بنى منطقية تسلسلية ضمن اللغة VHDL:
- التعليمة IF:

• مثال: المطلوب تصميم دائرة القلاب (D flip-flop) الموضحة بالشكل التالي:



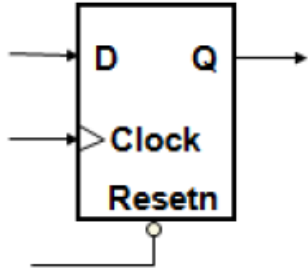
```

• LIBRARY ieee ;
  USE ieee.std_logic_1164.all ;
  ENTITY flipflop IS
    PORT (D, Clock: IN STD_LOGIC ;
          Q: OUT STD_LOGIC) ;
  END flipflop ;
  ARCHITECTURE behavioral OF flipflop IS
  BEGIN
    PROCESS ( Clock )
    BEGIN
      IF Clock'EVENT AND Clock = '1' THEN
        Q <= D ;
      END IF ;
    END PROCESS ;
  END behavioral ;

```

طريقة أخرى لتوصيف النبضة الصاعدة IF rising\_edge(Clock) THEN

# تصميم الأنظمة الرقمية باستخدام VHDL

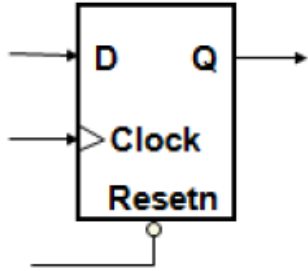


- تصميم بنى منطقية تسلسلية ضمن اللغة VHDL:
- التعليمة IF:

• مثال: المطلوب تصميم دائرة القلاب (D flip-flop) السابق مع قطب تصفير اللامتزامن (asynchronous reset):

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
ENTITY flipflop_ar IS
    PORT (D, Resetn, Clock: IN STD_LOGIC ;
          Q: OUT STD_LOGIC) ;
END flipflop_ar ;
ARCHITECTURE behavioral OF flipflop_ar IS
BEGIN
    PROCESS ( Resetn, Clock )
    BEGIN
        IF Resetn = '0' THEN
            Q <= '0' ;
        ELSIF rising_edge(Clock) THEN
            Q <= D ;
        END IF ;
    END PROCESS ;
END behavioral ;
```

# تصميم الأنظمة الرقمية باستخدام VHDL



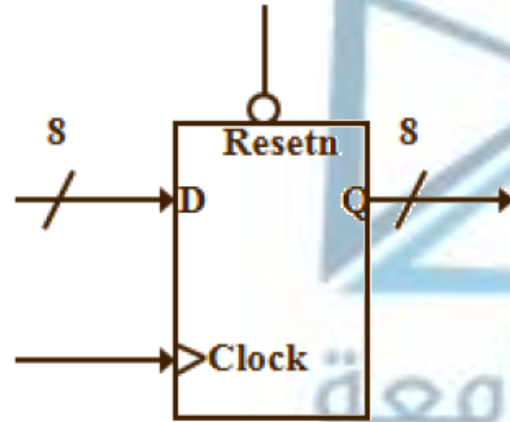
- تصميم بنى منطقية تسلسلية ضمن اللغة VHDL:
- التعليمة IF:

• مثال: المطلوب تصميم دائرة القلاب (D flip-flop) السابق مع قطب تصفير متزامن (synchronous):

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
ENTITY flipflop_sr IS
    PORT (D, Resetn, Clock: IN STD_LOGIC ;
          Q: OUT STD_LOGIC) ;
END flipflop_sr ;
ARCHITECTURE behavioral OF flipflop_sr IS
BEGIN
    PROCESS(Clock)
    BEGIN
        IF rising_edge(Clock) THEN
            IF Resetn = '0' THEN
                Q <= '0' ;
            ELSE
                Q <= D ;
            END IF ;
        END IF ;
    END PROCESS ;
END behavioral ;
```

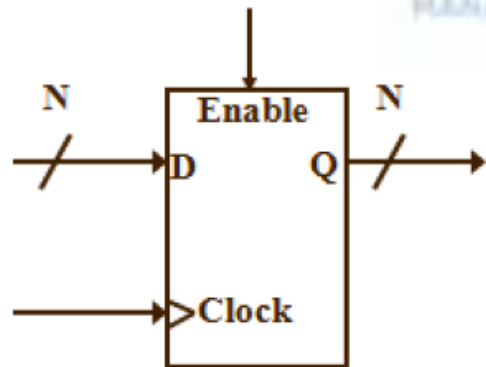
# تصميم الأنظمة الرقمية باستخدام VHDL

- تصميم بنى منطقية تسلسلية ضمن اللغة VHDL:  
• التعليمة IF:



- وظيفة: المطلوب تصميم دائرة المسجل (Register) مع قطب تصفير لامترامن ولـ 8 خانات والموضح بالشكل:

- وظيفة: المطلوب تصميم دائرة المسجل (Register) السابق مع قطب تصفير لامترامن ولـ n خانة



- وظيفة: المطلوب تصميم دائرة المسجل (Register) مع قطب تفعيل لامترامن ولـ n خانات والموضح بالشكل:

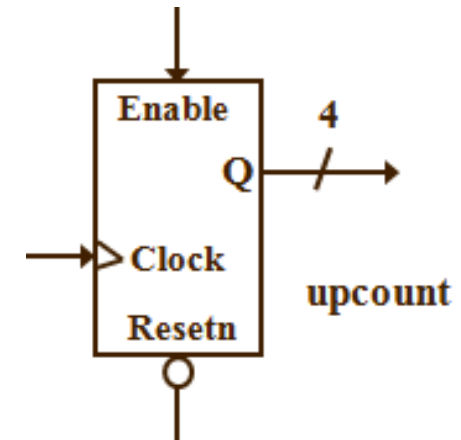
# تصميم الأنظمة الرقمية باستخدام VHDL

• تصميم بنى منطقية تسلسلية ضمن اللغة VHDL:

• التعليمة IF:

• مثال: المطلوب تصميم دائرة عداد تصاعدي بأربع خانات مع قطب تصفير لامتزامن:

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
USE ieee.std_logic_unsigned.all ;
ENTITY upcount_ar IS
    PORT (Clock, Resetn, Enable : IN STD_LOGIC ;
          Q: OUT STD_LOGIC_VECTOR (3 DOWNTO 0));
END upcount_ar ;
ARCHITECTURE behavioral OF upcount_ar IS
    SIGNAL Count : STD_LOGIC_VECTOR(3 DOWNTO 0);
BEGIN
    PROCESS ( Clock, Resetn )
    BEGIN
        IF Resetn = '0' THEN
            Count <= "0000" ;
        ELSIF rising_edge(Clock) THEN
            IF Enable = '1' THEN
                Count <= Count + 1 ;
            END IF ;
        END IF ;
    END PROCESS ;
    Q <= Count ;
END behavioral ;
```



# تصميم الأنظمة الرقمية باستخدام VHDL

• تصميم بنى منطقية تسلسلية ضمن اللغة VHDL:

• التعليمة IF:

• مثال: المطلوب تصميم دائرة عداد رقمي صاعد حلقي من القيمة 0 إلى القيمة 9 بحيث يزداد خرجة قيمة واحدة عند كل جبهة صاعدة لنبضات الساعة clk (عداد عشري حلقي):

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY counter IS
    PORT (clk : IN STD_LOGIC;
          digit : OUT INTEGER RANGE 0 TO 9);
END counter;
ARCHITECTURE counter OF counter IS
BEGIN
    count: PROCESS(clk)
        VARIABLE temp : INTEGER RANGE 0 TO 10;
    BEGIN
        IF (clk'EVENT AND clk='1') THEN
            temp := temp + 1;
            IF (temp=10) THEN
                temp := 0;
            END IF;
        END IF;
        digit <= temp;
    END PROCESS count;
END counter;
```