

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



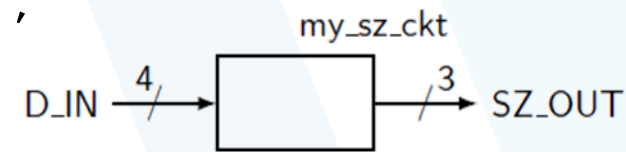
- تصميم بني منطقية تسلسلية ضمن اللغة VHDL:
 - أهم التعليمات المستخدمة مع نبضات الساعة
 - تسمى أيضاً بـ Clock Functions وهي موجودة في مكتبة الـ ieee ضمن الـ "std_logic_1164" package
 - التعليمة event: حيث تأخذ الشكل التالي Clk'event وتأخذ القيمة true عند حدوث اي تغيير في إشارة نبضات الساعة
 - التعليمة 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"
 - سيكون 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 بني منطقية تسلسلية ضمن اللغة VHDL:
- أهم التعليمات المستخدمة مع نبضات الساعة
- مثال: المطلوب تصميم دائرة تحتوي على أربع مداخل وخرج من ثلاث إشارات. العلاقة بين إشارة الدخل والخرج موضحة في الجدول:

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

- ```
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;
```



# تصميم الأنظمة الرقمية باستخدام VHDL



- تصميم بني منطقية تسلسلية ضمن اللغة VHDL:
- أهم التعليمات المستخدمة مع نبضات الساعة
- مثال: المطلوب تصميم دائرة تابعها المنطقي معطى بالعلاقة التالية:

```
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 بني 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;



- حل آخر:

- 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:

• التعليمات Process:

• إن تعليمات Process من التعليمات التفرعية في لغة VHDL

• تشير إلى أن كل ما يكتب ضمنها يتم تنفيذ بصورة متتابعة وعادة ما تكتب التعليمات IF, WAIT, CASE, LOOP والتي تنفذ بصورة تسلسلية من الأعلى إلى الأسفل وبصورة دورية مادامت تعليمات Process فعالة تتضمن تعليمات Process عدد من المتحولات أو الشروط المنطقية تسمى قائمة الحساسية sensitivity list والتي تعتبر مجموعة من المتحولات لا يتم الدخول إلى كتلة التعليمات Process إلا عند حدوث تغيير على هذه المتحولات

• تكتب تعليمات Process في الكود الرئيسي وتنفذ كل مرة يتغير فيها أحد متحولات الحساسية أو الشروط المنطقية

• الصيغة العامة لهذه التعليمات 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 بني منطقية تسلسلية ضمن اللغة VHDL:

- التعليمة IF:

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

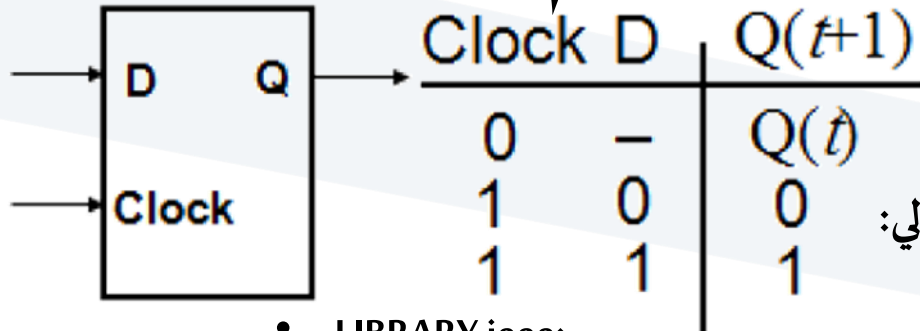
- IF conditions THEN assignments;
ELSIF conditions THEN assignments;

..

ELSE assignments;
END IF;

- ومن أهم التطبيقات والدارات التي توضح عمل هذه التعليمات هي المسجلات والعدادات
- أمثلة:

تصميم الأنظمة الرقمية باستخدام VHDL



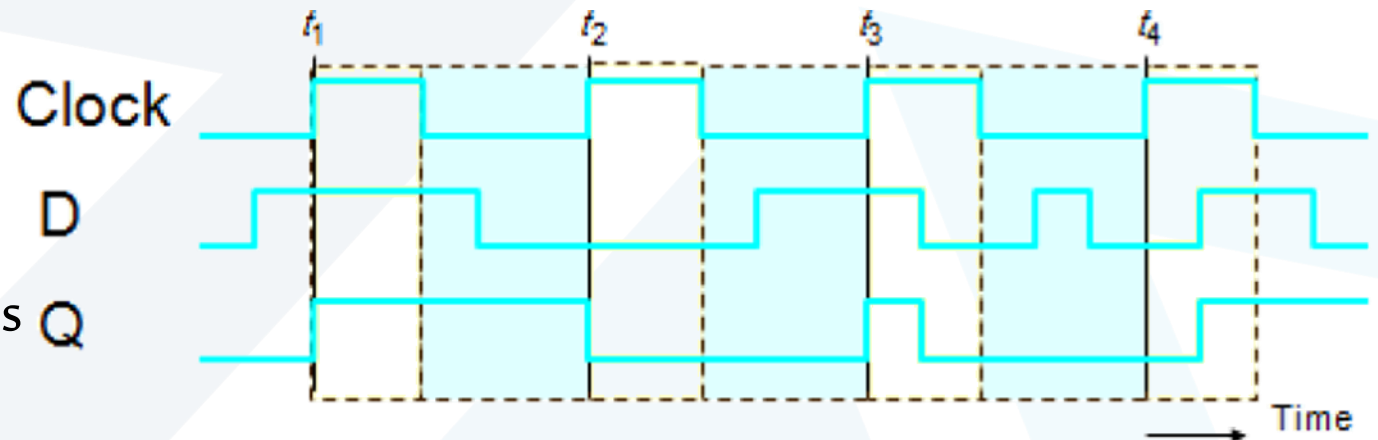
• تصميم بني منطقية تسلسلية ضمن اللغة VHDL:

• التعليمات 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 -- if(Clock'event and Clock='1')
            Q <= D;
        END IF;
    END PROCESS;
END behavioral;
    
```



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

Q <= D;

END IF;

END PROCESS;

END behavioral;

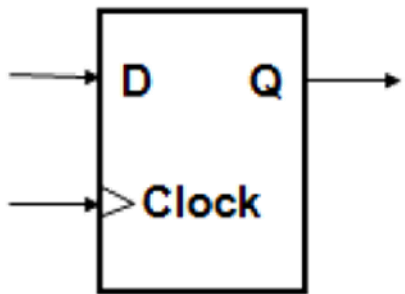
تصميم الأنظمة الرقمية باستخدام VHDL

• تصميم بني منطقية تسلسلية ضمن اللغة VHDL:

• التعليمة IF:

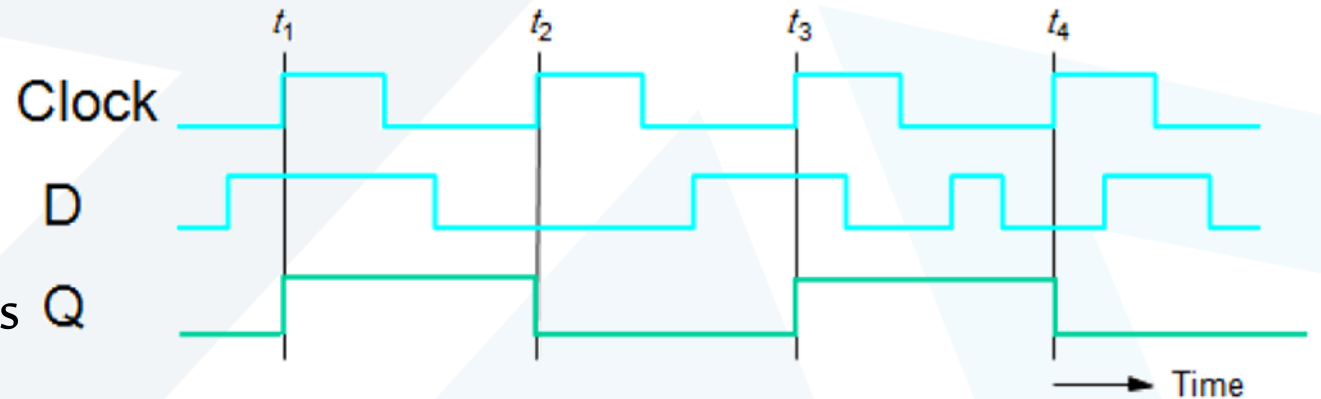
• مثال: المطلوب تصميم دائرة القلاب (D flip-flop) الموضحة بالشكل التالي:

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



```

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 -- IF rising_edge(Clock) THEN
            Q <= D ;
        END IF ;
    END PROCESS ;
END behavioral ;
    
```



طريقة أخرى لتوصيف النبضة الصاعدة

تصميم الأنظمة الرقمية باستخدام VHDL

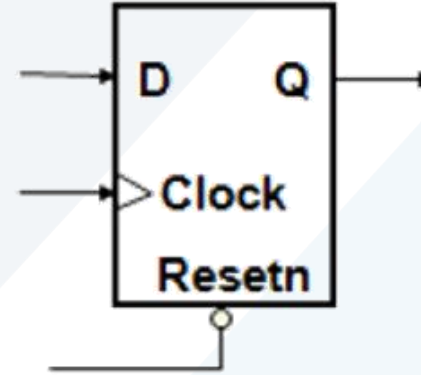


• تصميم بني منطقية تسلسلية ضمن اللغة VHDL:

• 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

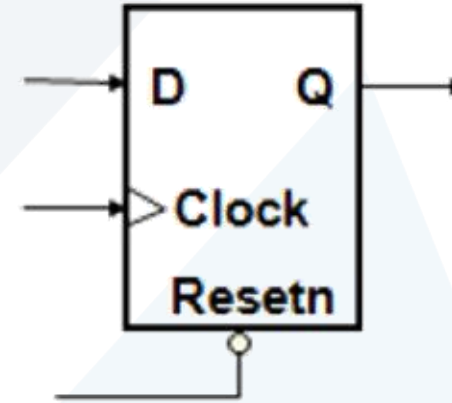


• تصميم بني منطقية تسلسلية ضمن اللغة VHDL:

• التعليمة IF:

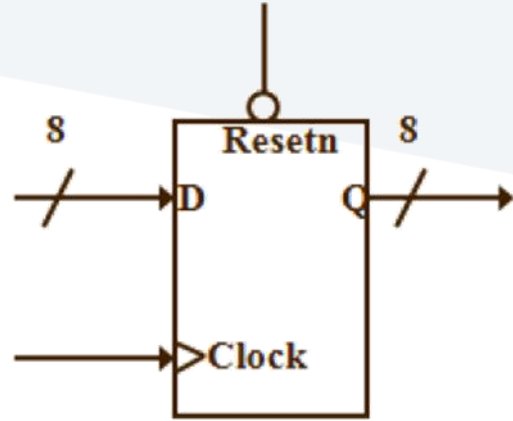
• مثال: المطلوب تصميم دائرة القلاب (D flip-flop) السابق مع قطب تصفير متزامن (synchronous):

```
• 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;
```



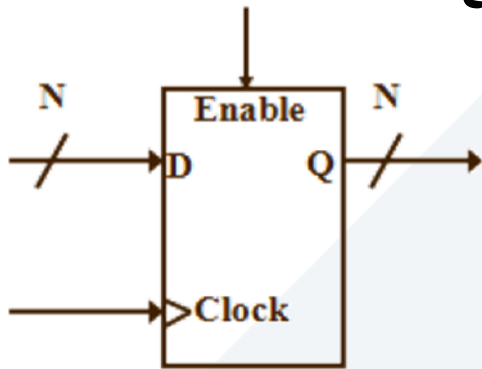
تصميم الأنظمة الرقمية باستخدام VHDL

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



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

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



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