جامعة حلب في المناطق المحررة
كلية الهندسة المعلوماتية
السنة الرابعة

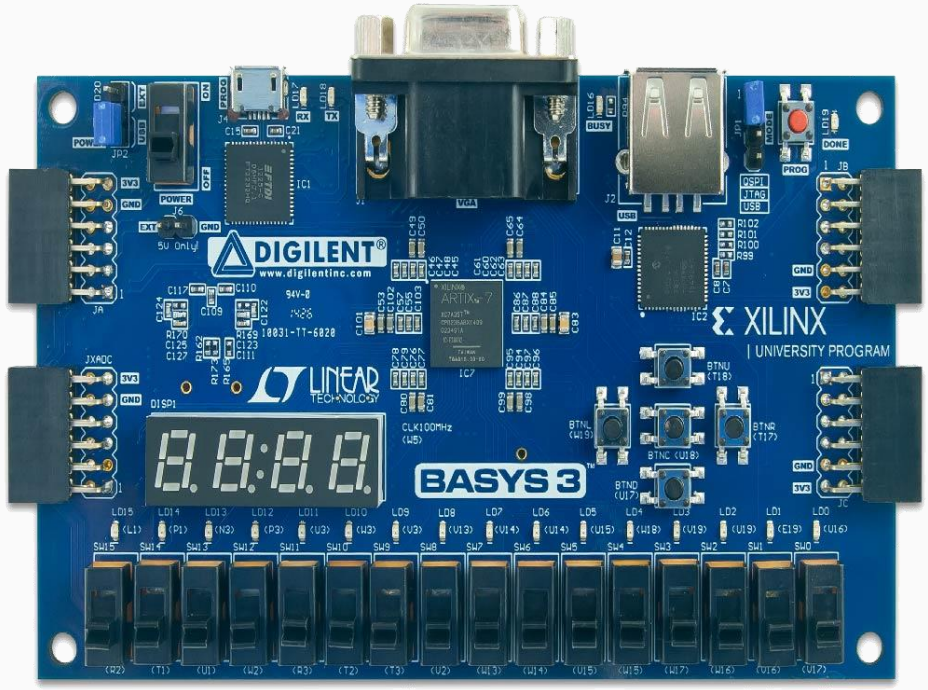
مقرر عملي

بنية وتنظيم الحواسيب2

تنفيذ دارة الجامع الكامل ونصف الجامع

د.م. عبد القادر غزال
م. محمد نور بدوي

العام الدراسي: 2023 - 2024



المحاضرة العملية الرابعة

قواعد عامة

1. يجب أن تبدأ جميع الأسماء بحرف (a - z) أو (A - Z).
2. لا تستخدم في الاسم محرفًا محجوزًا (- ,+ ,& ,? ,!).
3. يجب عدم تكرار الاسم في الكيان أو المعمارية.
4. يشار إلى التعليق بخطين متقطعين متتاليين (--).
5. لا تستخدم خطيين تحتيين (__) متتاليين أو أكثر داخل الاسم (مثلًا Or__gate غير مسموح).
6. لغة VHDL غير حساسة لحالة الأحرف (يمكن استخدام الأحرف الصغيرة أو الكبيرة أو خليط منهما AnD | and).
7. لا توجد قيود عند كتابة الأكواد، كالفراغات بين الكلمات، فالمترجم يعالج الفراغ والانتقال إلى السطر التالي بالطريقة ذاتها

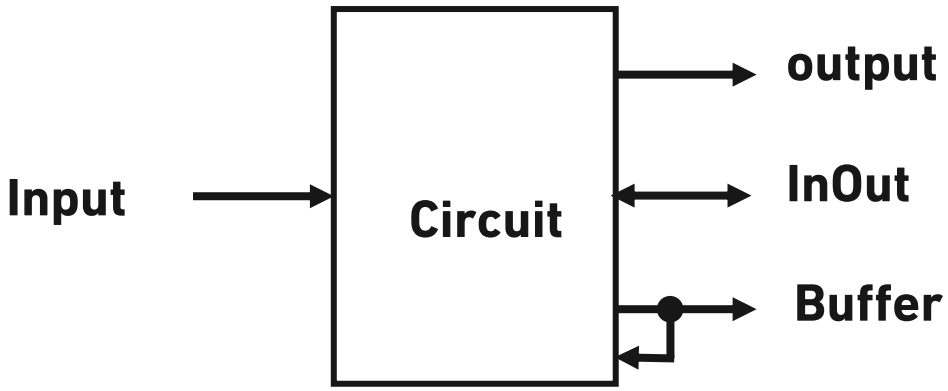
أنماط الإشارة

يوجد للإشارة أربعة أنماط IN, OUT, INOUT, BUFFER

- النمطان IN, OUT منفذان وحيدا الاتجاه

- النمط INOUT ثنائي الاتجاه

- النمط BUFFER يستخدم عندما يتطلب التطبيق استخدام إشارة الخرج داخل الكيان.



بعض الكلمات المحجوزة الشائعة

ALL, AND, ARCHITECTURE, ARRAY, BEGIN, CASE, COMPONENT, VARIABLE

CONFIGURATION, DOWNTO, ELSE, ELSIF, ENTITY, FOR, FUNCTION, GENERATE,

IF, IS, LIBRARY, NOT, OR, OTHERS, OUT, PORT, PROCESS, RANGE, RECORD,

REPORT, RETURN, SIGNAL, THEN, TO, TYPE.

أنواع المتغيرات في لغة VHDL

Signal X: std_logic_vector (3 DOWNTO 0); X<= "1111"

النوع	الوصف	مثال
Bit (bit_vector)	يُستخدم لتمثيل قيم ثنائية (0 أو 1).	signal X : bit := '1'; signal X: bit_vector(3 DOWNTO 0) := "1111" ;
std_logic(std_logic_vector)	يُستخدم لتمثيل قيم منطقية ('X', '0', '1', 'Z')	signal X: std_logic := '1'; signal X: std_logic_vector(3 TO 0) := "1010";
boolean	يُستخدم لتمثيل القيم الصحيحة True أو False	signal X: boolean := TRUE;
integer	يُستخدم لتمثيل الأعداد الصحيحة.	signal X: integer := 10;
signed	يُستخدم لتمثيل الأعداد الصحيحة المؤشرة (الموجبة والسالبة).	signal X: signed(7 DOWNTO 0) := "11011011"; Des = -37
unsigned	يُستخدم لتمثيل الأعداد الصحيحة الغير مؤشرة (جميع الأعداد موجبة)	signal X: unsigned(3 TO 0) := "1010"; Des = 10

تعريف نوع جديد في لغة VHDL

يمكن تعريف النوع الجديد باستخدام الكلمة المفتاحية TYPE لتمثيل بنية بيانات مخصصة تحمل أنواع البيانات والعناصر التي تحتاجها.

```
TYPE byte IS RANGE  0 TO 255;

TYPE nibble IS (‘0’ , ‘1’ , ‘2’ , ‘3’);
```

الجمل الشرطية في لغة VHDL

WHEN/ELSE Statement

تستخدم لتنفيذ أوامر معينة عندما تتوافر شروط محددة، وإلا ستتم تنفيذ أوامر بديلة، حيث يتم تحديد سلوك مختلف على أساس شرط معين.

```
ARCHITECTURE Xor_AR OF Xor_gate IS
BEGIN
S<=  '0' WHEN(a AND b)="00" OR (a AND b)="11"
ELSE '1';
END Xor_AR ;
```

الجمل الشرطية في لغة VHDL

IF Statement

```
ARCHITECTURE Xor_AR OF Xor_gate IS
BEGIN
PROCESS (a,b)
if a='0'and b='0' then
c <='0';
elsif a='1'and b='1'
c <='0';
else
c <='1';
end if;
end PROCESS;
END Xor_AR ;
```

❖ عندما يتم تغيير قيم a أو b، ستنفذ العملية **process** لفحص الشروط واتخاذ الإجراء المناسب وتحديث الإشارات بناءً على الشروط الموجودة داخلها. هذا يجعلها مفيدة لتصميم أنظمة رقمية تتفاعل مع تغييرات في الإشارات وتنفذ إجراءات محددة عندما تتغير الظروف المحددة.

الجمل الشرطية في لغة VHDL

CASE Statement

```
ARCHITECTURE Xor_AR OF Xor_gate IS
BEGIN
PROCESS (a, b)
BEGIN
    CASE (a AND b) IS
        WHEN "00"   =>   c <= '0';
        WHEN "11"   =>   c <= '0';
        WHEN OTHERS =>   c <= '1';
    END CASE;
END PROCESS;
END Xor_AR ;
```

مثال عملي 1

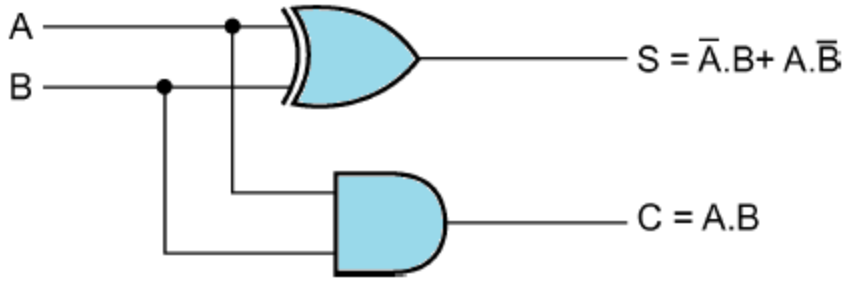
اكتب كود VHDL لتنفيذ دارة الجامع النصفي Half Adder

```vhdl
-- library declaration
library IEEE;
use IEEE.std_logic_1164.all;

-- entity
entity Half_Adder is
port (
a , b    : in std_logic;
s , cout : out std_logic);
end Half_Adder;

-- architecture
architecture Half_Adder_ar of Half_Adder is
begin
s    <= a xor b;
cout <= a and b;
end Half_Adder_ar;
```

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

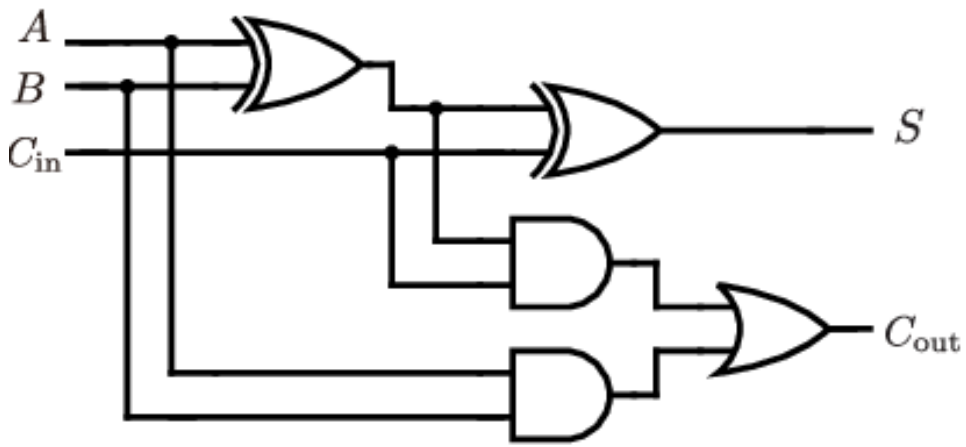


مثال عملي 2

اكتب كود VHDL لتنفيذ دارة الجامع الكامل Full Adder

$$S = A \oplus B \oplus Cin$$

$$Cout = A.B + B.Cin + Cin.A$$



Inputs			Outputs	
A	B	C_{in}	S	C_{out}
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

انتهت المحاضرة