



جامعة حلب في المناطق المحررة  
كلية الهندسة المعلوماتية  
السنة الرابعة

مقرر عملي

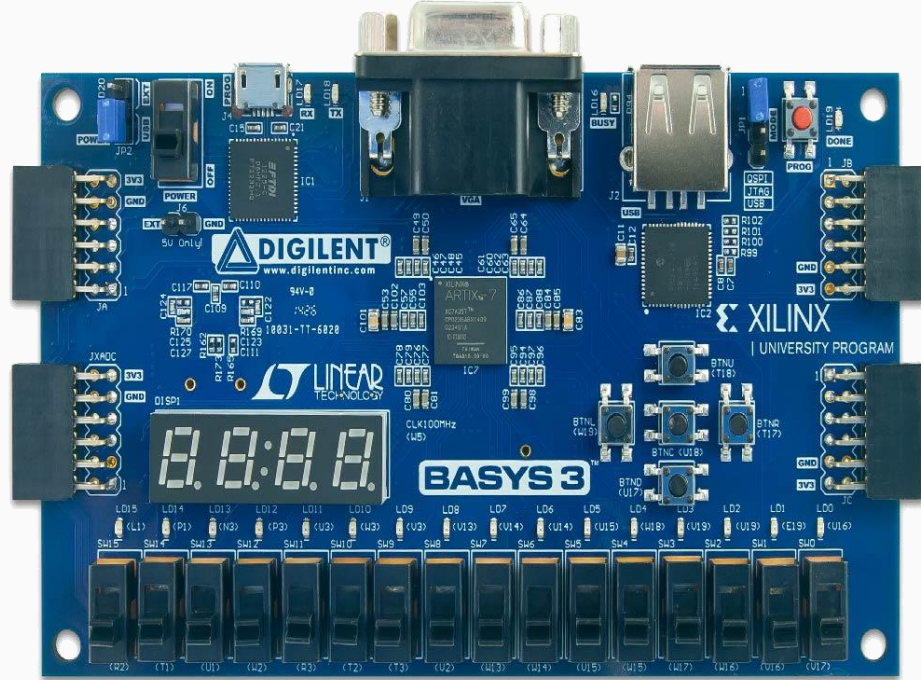
# بنية وتنظيم الحواسيب 2

تنفيذ توابع منطقية بسيطة بلغة VHDL

د.م. عبد القادر غزال

م. محمد نور بدوي

العام الدراسي: 2023 - 2024



المحاضرة العملية الثالثة



# لغات وصف الكيان الصلب HDLs

هي عبارة عن لغات برمجة تستخدم تعابير نصية تصريحية (Declarative Textual) لوصف سلوك أو عمل الوحدات المنطقية (Digital Logic) على الكيان الصلب بشكل مباشر، تمتاز هذه اللغات بأنها وصفية تتعلق مباشرة ببناء الكيان الصلب، كما أنها تختلف عن اللغات البرمجية المتخصصة لتطبيقات الحاسب بشكل جذري

من أشهر اللغات المستخدمة:

- VHDL
- Verilog





# VHDL

**V**ery High speed integrated Circuit

**H**ardware

**D**escription

**L**anguage

هي لغة توصيف عمل أو سلوك الكيان الصلب على شرائح الدارات المتكاملة الرقمية فائقة السرعة، ظهرت عام 1980، وتم تطويرها من قبل وزارة الدفاع الأمريكية.

إن الهيكلية البرمجية للغة VHDL تنفذ بشكل تفرعي متزامن، كما أنها ليست خاصة بنظام أو كيان محدد أو شركة معينة، وإنما هي نظام قياسي تعتمد عليه الشركات المصنعة، فيمكن تحويل أي برنامج مكتوب بلغة ال VHDL واستخدامه لتنفيذ أي نظام بترجمته وبرمجته إلى العناصر المنطقية القابلة للبرمجة.



## VHDL



**V**ery High speed integrated Circuit  
**H**ardware  
**D**escription  
**L**anguage

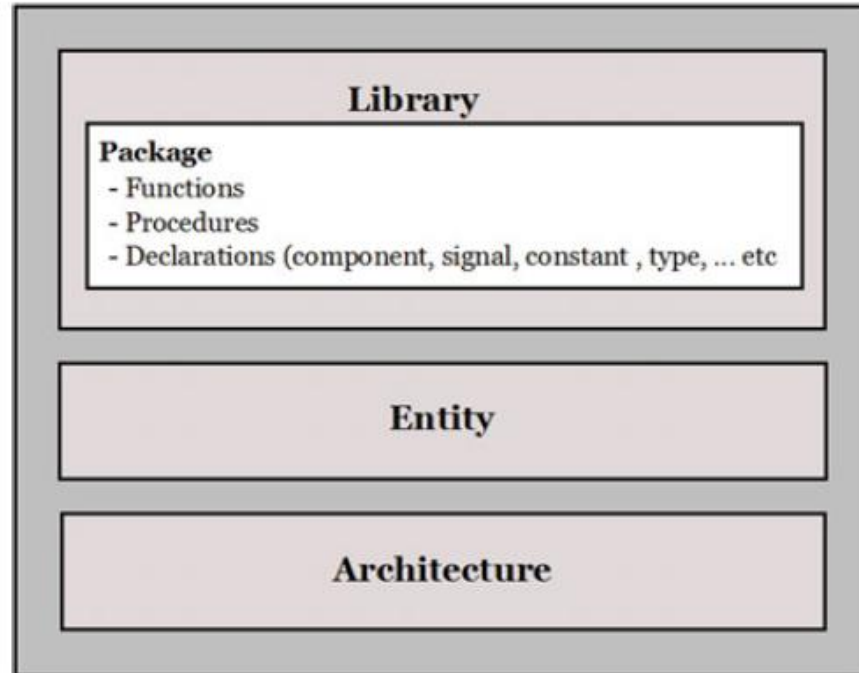
تُستخدم VHDL أيضًا لإجراء محاكاة واختبار للأنظمة الإلكترونية المصممة، مما يساعد في تحليل أدائها واكتشاف الأخطاء وإصلاحها قبل تنفيذها على أجهزة فعلية.

تعد VHDL أداة أساسية للمهندسين في مجال تصميم الأنظمة الإلكترونية والدوائر المنطقية لتسهيل عملية التطوير والتصميم والاختبار.

إن الخرج النهائي لبرنامج مكتوب بلغة VHDL هو ملف التوصيلات "Netlist"



# عناصر التصميم في لغة ال VHDL



إن عملية وصف الكيان الصلب في لغة ال VHDL يتضمن:

- Entity: هي عناصر تصميم رئيسية تصف واجهة التصميم.
- Architecture: عناصر تصميم ثانوية تحوي التمثيل والوصف الحقيقي للتصميم
- Libraries: المكتبات



## 1. Entity الكيان

يعتبر الكيان المستوى الأعلى والأكثر أهمية في التصميم، وفيه يتم تعريف مداخل ومخارج النظام على البوابات إضافة إلى جميع الموارد المرتبطة بالكيان. في الكيان نظام واحد أو أكثر ، يتم التصريح عنه كما هو مبين في الصورة الجانبية.

```
entity gate is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           Z : out STD_LOGIC);
end gate;
```



## 2. Architecture البنية الهيكلية

تضم الوظائف التي تصف سلوك الكيان الصلب المرتبط بالوظيفة الأساسية المعرفة بالكيان Entity، يمكن أن يضم الكيان هيكلية أو أكثر وكل واحدة منهما يمكن أن تكون وصفًا سلوكيًا للنظام أو وصفًا بنيويًا، ويتم التصريح عن الهيكلية كما هو موضح في الصورة الجانبية.

```
architecture Behavioral of gate is  
begin
```

```
    Q <= A and B;
```

```
end Behavioral;
```



## 3. Libraries المكتبات

تضم المكتبات مجموعات النماذج

التصميمية والتوابع الوحدات للغة VHDL،

ويمكن أن يضم البرنامج مكتبة أو أكثر تبعًا

للوظائف المطلوبة من البرنامج

```
library IEEE;
```

```
use IEEE.STD_LOGIC_1164.ALL;
```

## ملاحظات:

$D \leq A \text{ and } B;$      $==$      $d \leq a$      $\text{AnD } b;$

-- This is a comment

لغة VHDL ليست حساسة لحالة الأحرف وللمسافات

تبدأ التعليقات في VHDL بالرمز "--" (شرطتان متتاليتان).





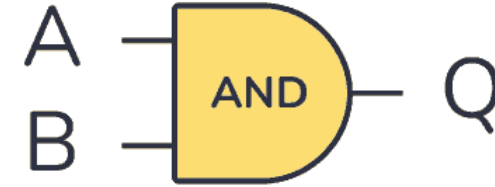
# تصميم بوابة And بلغة VHDL



```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity and_gate is
    Port ( A : in STD_LOGIC;
          B : in STD_LOGIC;
          Q : out STD_LOGIC);
end and_gate;

architecture Behavioral of and_gate is
begin
    Q <= A and B;
end Behavioral;
```



A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

قم بإجراء التعديل المناسب ليكون لدينا بوابة And بثلاث مداخل على أحد أقطابها بوابة Not



# تصميم بوابة And بلغة VHDL



## ضبط الدخل A

Force Clock: /And\_Gate/A

Enter parameters below to force the signal to a constant value. Assignments made from within HDL code or any previously applied constant or clock force will be overridden.

Signal name: /And\_Gate/A

Value radix: Binary

Leading edge value: 1

Trailing edge value: 0

Starting after time offset: 0ns

Cancel after time offset: 1000ns

Duty cycle (%): 50

Period: 100ns

OK Cancel

## ضبط الدخل B

Force Clock: /and\_gate/B

Enter parameters below to force the signal to a constant value. Assignments made from within HDL code or any previously applied constant or clock force will be overridden.

Signal name: /and\_gate/B

Value radix: Hexadecimal

Leading edge value: 1

Trailing edge value: 0

Starting after time offset: 0ns

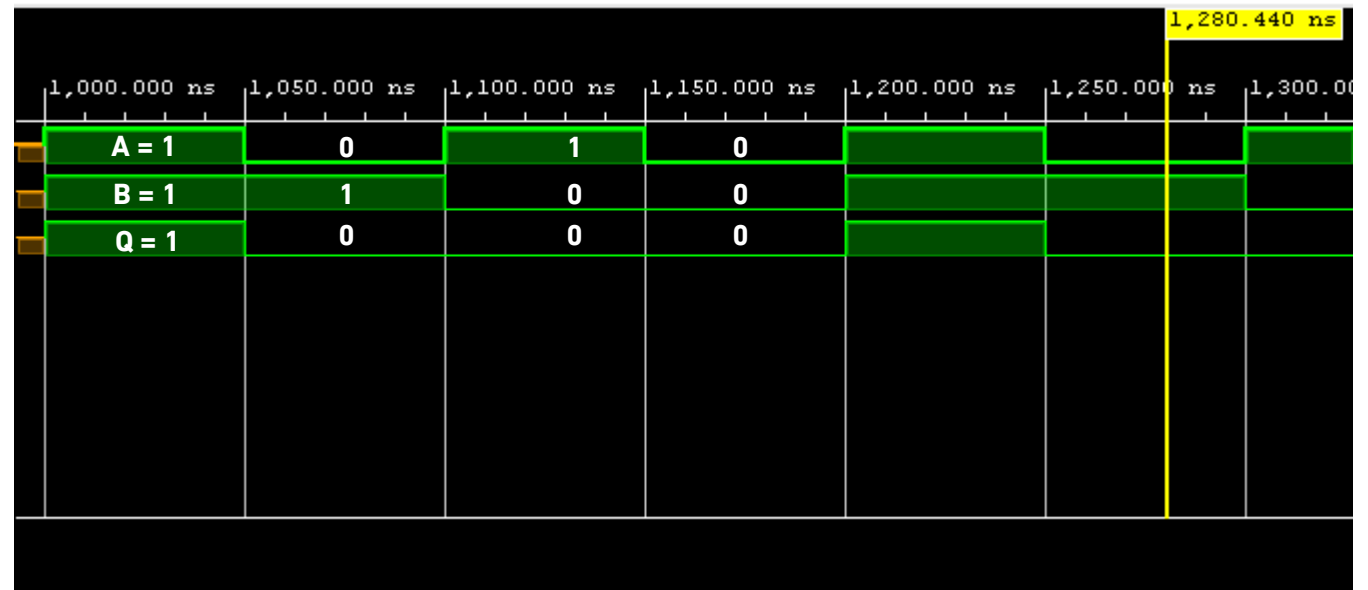
Cancel after time offset: 1000ns

Duty cycle (%): 50

Period: 200ns

OK Cancel

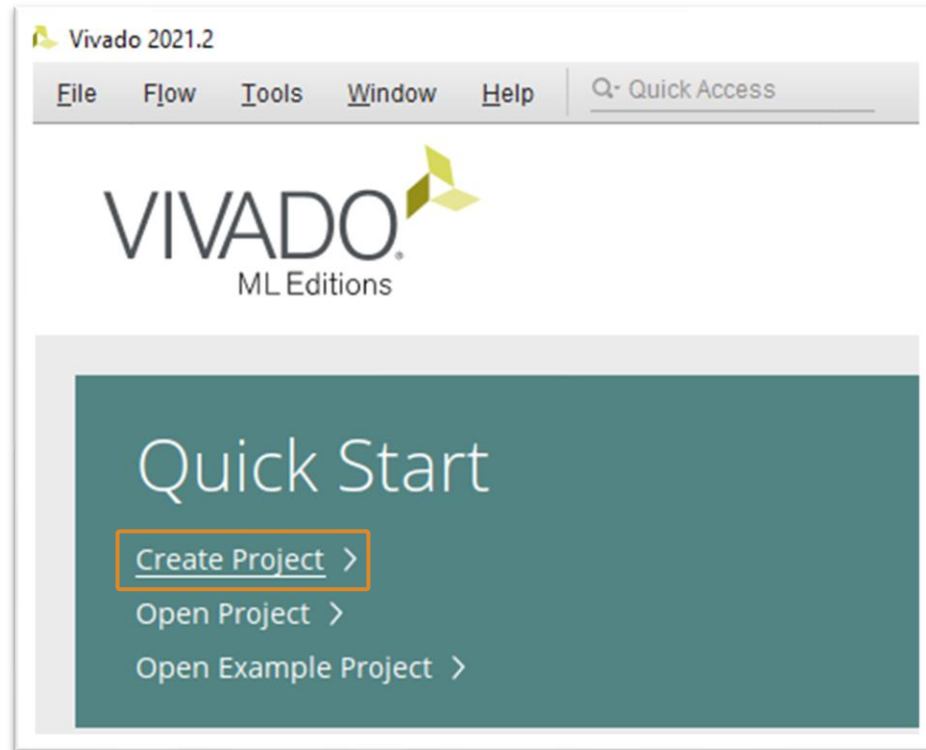
## نتيجة خرج المحاكى





# طريقة تصميم بوابة And بلغة VHDL باستخدام برنامج Vivado

1



2

**Project Name**  
Enter a name for your project and specify a directory where the project will be created.

Project name:

Project location:

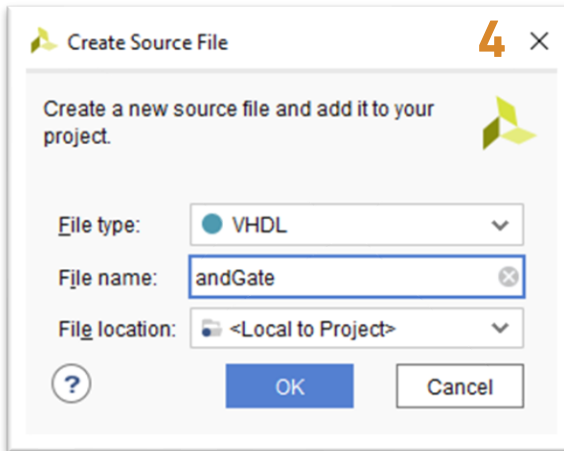
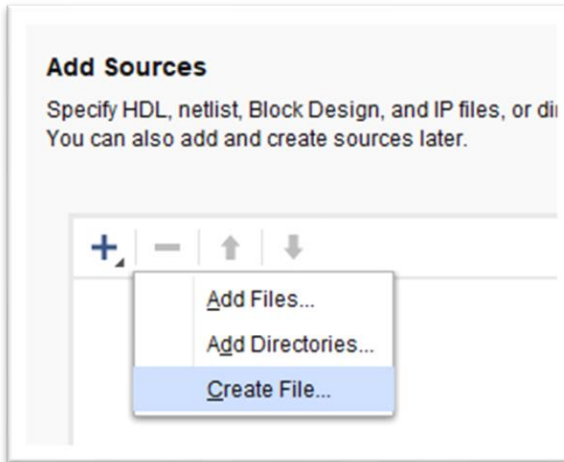
☒ Create project subdirectory

Project will be created at: D:/VHDL/And\_Gate

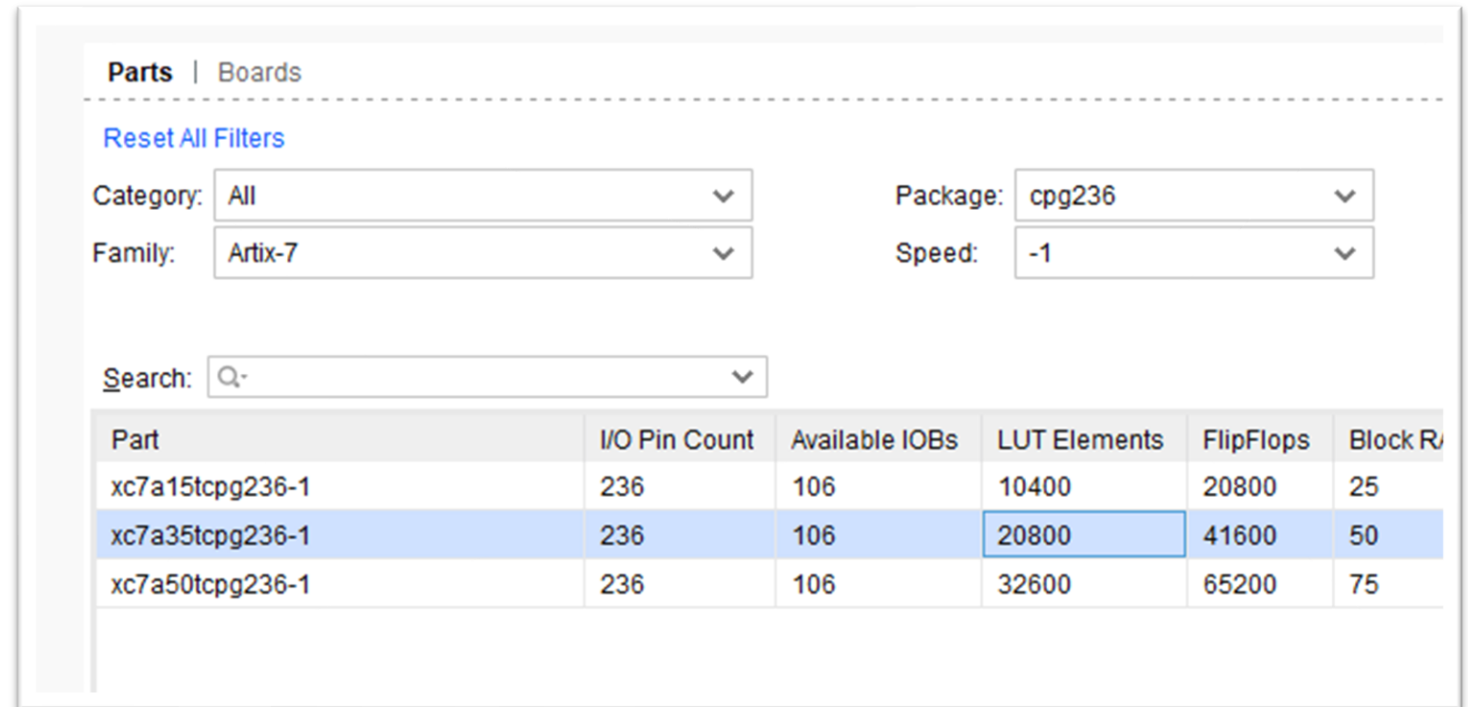


# طريقة تصميم بوابة And بلغة VHDL باستخدام برنامج Vivado

3



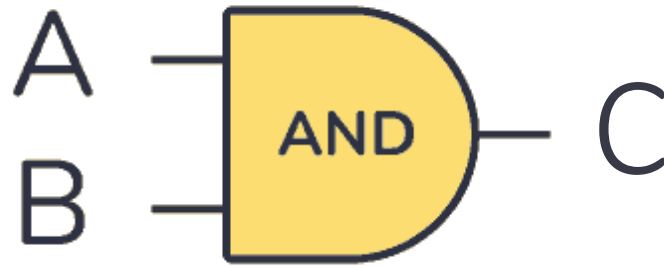
5





# طريقة تصميم بوابة And بلغة VHDL باستخدام برنامج Vivado

بعد إنشاء المشروع نقوم بتحديد المداخل والمخارج عن طريق نافذة إدخال، ويمكن الاستغناء عنها وكتابة الكود بشكل يدوي.



**Define Module**

Define a module and specify I/O Ports to add to your source file.  
For each port specified:  
MSB and LSB values will be ignored unless its Bus column is checked.  
Ports with blank names will not be written.

**Module Definition**

Entity name:

Architecture name:

**I/O Port Definitions**

Port Name	Direction	Bus	MSB	LSB
A	in	<input type="checkbox"/>	0	0
B	in	<input type="checkbox"/>	0	0
C	out	<input type="checkbox"/>	0	0

Buttons: ? OK Cancel



# طريقة تصميم بوابة And بلغة VHDL باستخدام برنامج Vivado



نقوم الآن بتكملة كتابة الكود بحيث يكون الملف ضمن مجلد Design Sources بعدها نضغط على Run Synthesis

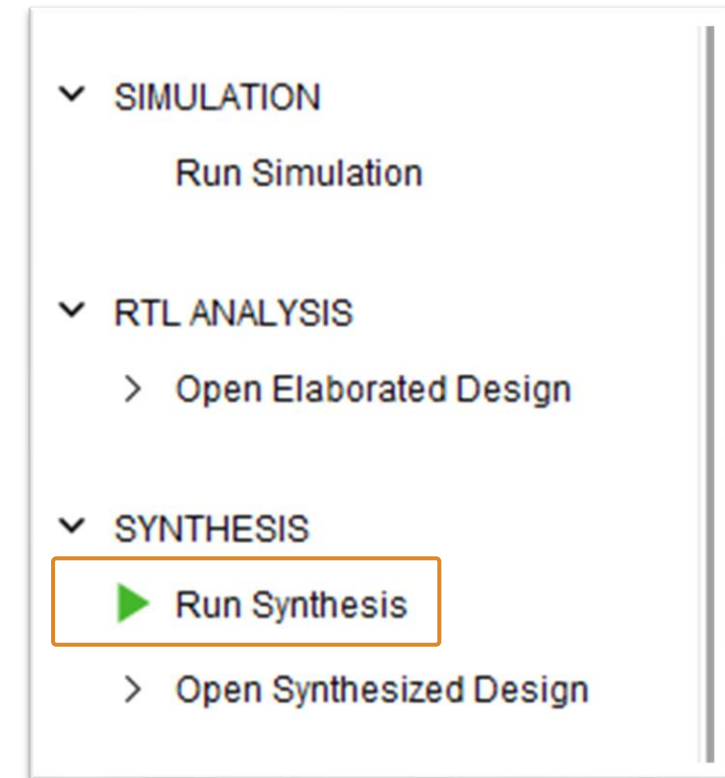
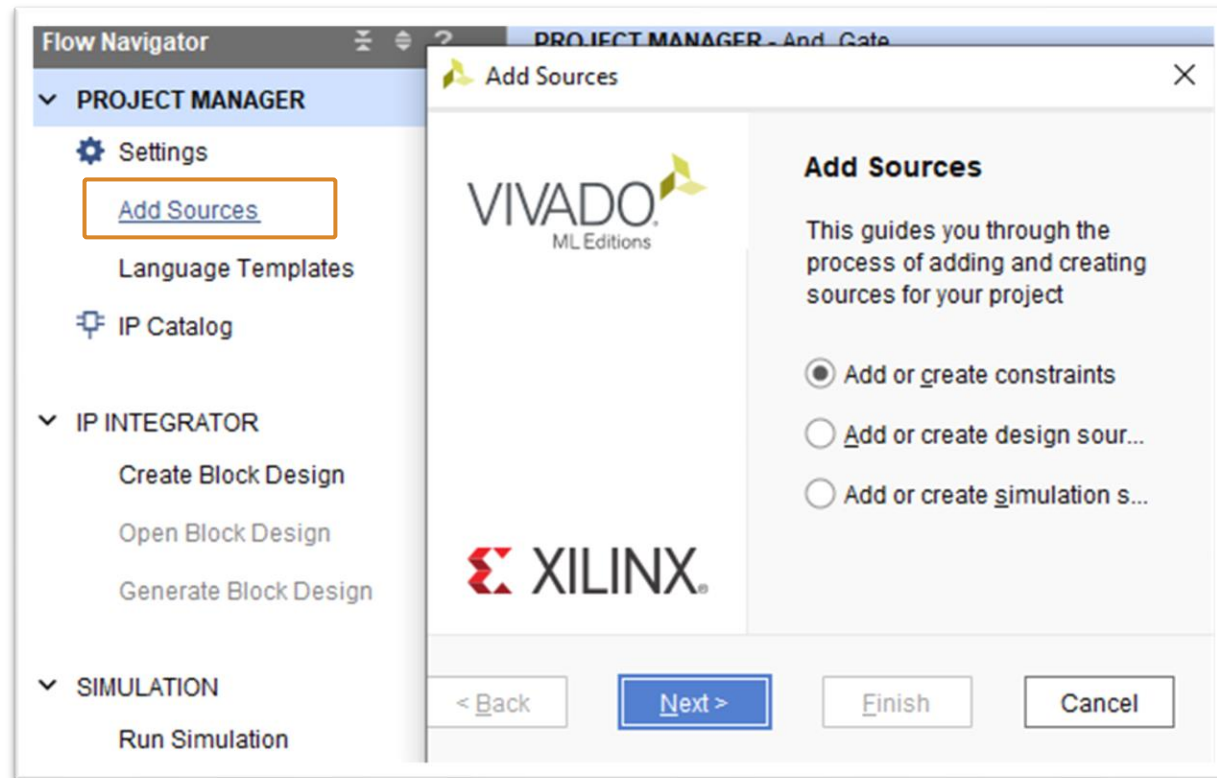
The screenshot displays the Vivado IDE interface. On the left, the 'Sources' window shows the project hierarchy with 'Design Sources (1)' containing 'And\_Gate(Behavioral) (And\_Gate.vhd)'. Below it, the 'Source File Properties' window for 'And\_Gate.vhd' is open, showing it is 'Enabled' and located at 'D:/VHDL/And\_Gate/And\_Gate.srscs/sources\_1/new'. The main editor window on the right shows the VHDL code for 'And\_Gate.vhd'. The code defines an entity 'And\_Gate' with two input ports 'A' and 'B' of type 'STD\_LOGIC', and one output port 'C' of type 'STD\_LOGIC'. The architecture 'Behavioral' contains a single line of code: 'C <= A and B;'. This line is highlighted in yellow, and the text 'C <= A and B;' is also enclosed in an orange box.

```
32 --use UNISIM.VComponents.all;
33
34 entity And_Gate is
35     Port ( A : in STD_LOGIC;
36           B : in STD_LOGIC;
37           C : out STD_LOGIC);
38 end And_Gate;
39 architecture Behavioral of And_Gate is
40 begin
41     C <= A and B;
42 end Behavioral;
43
```



# طريقة تصميم بوابة And بلغة VHDL باستخدام برنامج Vivado

بعدها نقوم بإنشاء ملف جديد constraints بعدها نضغط على Run Synthesis من القائمة الجانبية





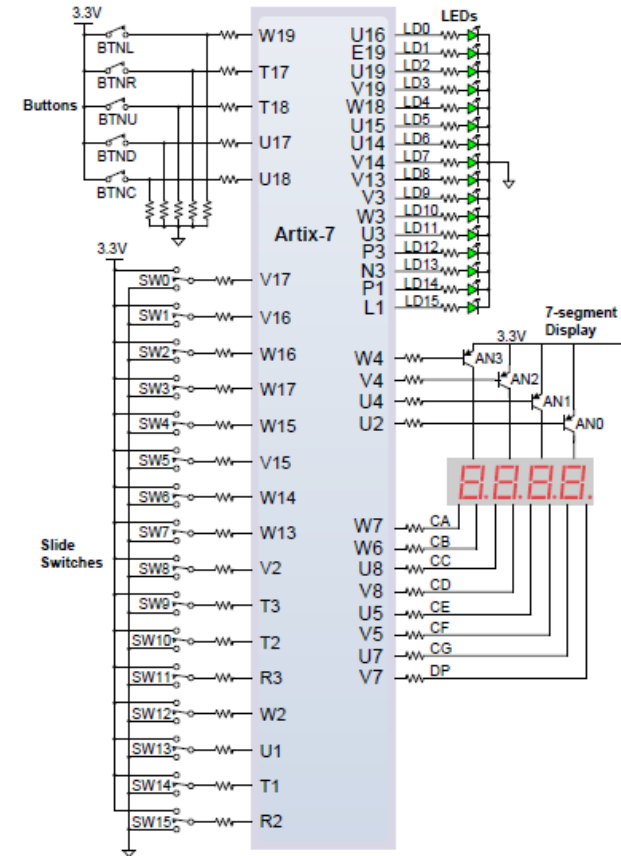
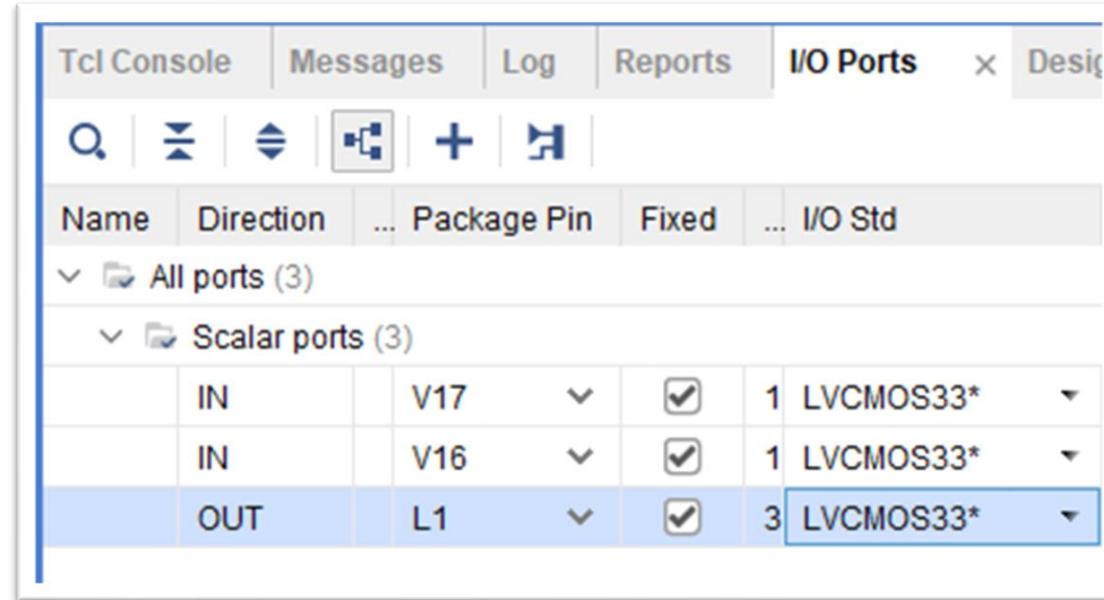
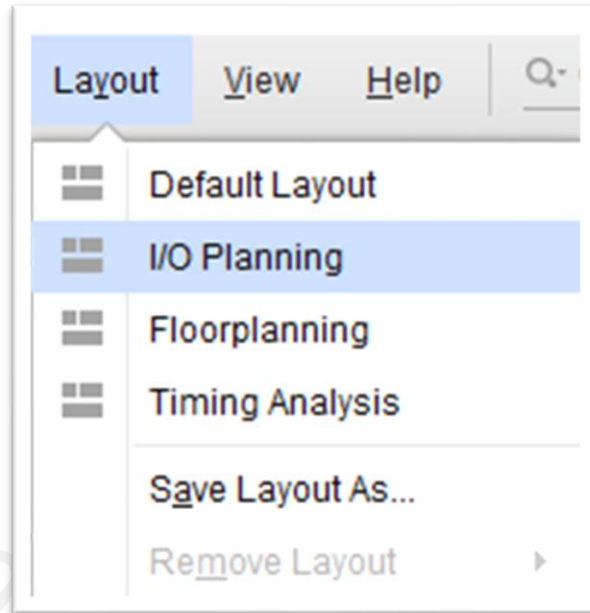


# طريقة تصميم بوابة And بلغة VHDL باستخدام برنامج Vivado



بعدها نقوم بتحديد أقطاب الدخل والخرج لشريحة الـ FPGA كما هو موضح في مخطط التوصيل في إصدار 3 Basys

في مثالنا: حددنا SW0 و SW1 كمداخل والتي يعبر عنها بـ A و B في بوابة AND، وخرج البوابة الليد L1

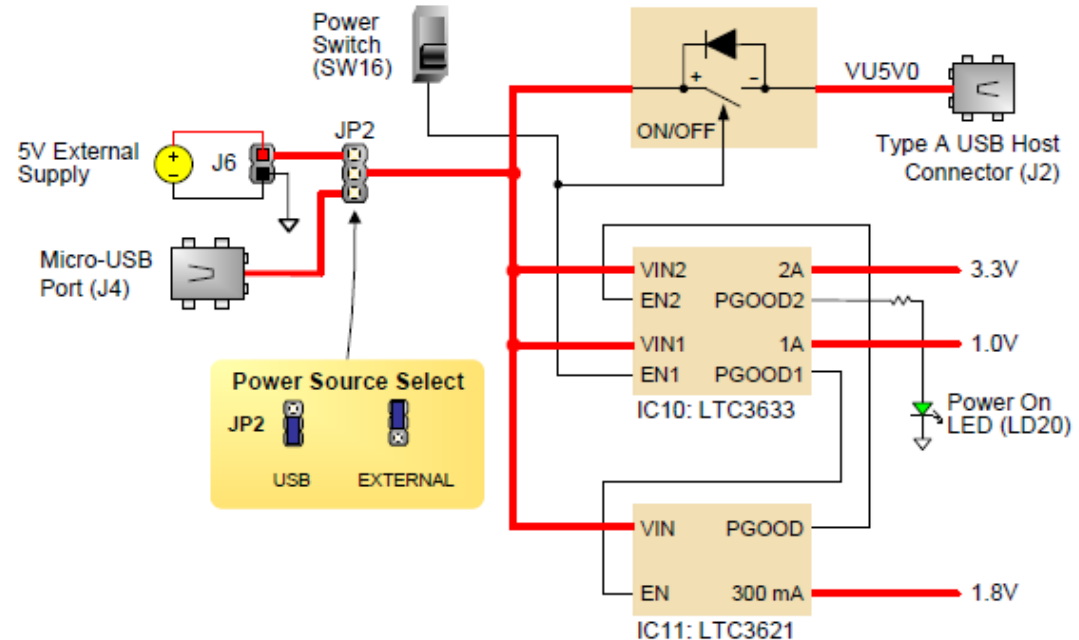
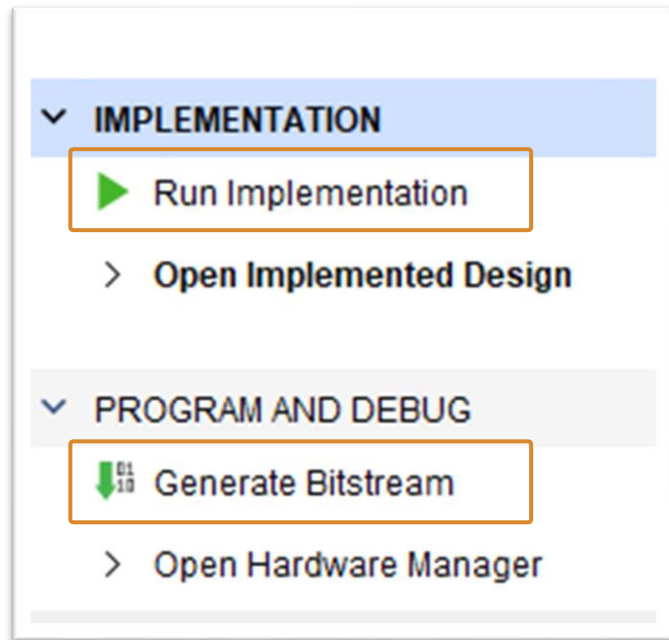






# طريقة تصميم بوابة And بلغة VHDL باستخدام برنامج Vivado

نقوم بتوصيل شريحة الـ FPGA مع الحاسب باستخدام كبل micro-usb بعدها نضغط على Run Implementation ومن ثم Generate Bitstream من القائمة الجانبية.

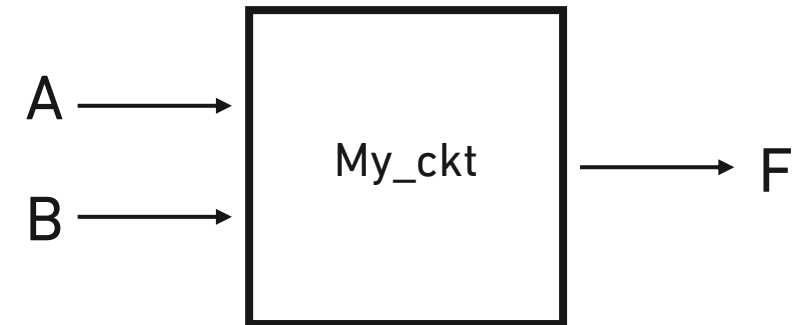


دائرة التغذية داخل الشريحة



اكتب كود VHDL لتنفيذ الوظيفة المعبر عنها بالمعادلة المنطقية التالية:  $F = A.B + \overline{B}$

```
-- library declaration
library IEEE;
use IEEE.std_logic_1164.all;
-- entity
entity My_ckt is
port (
A,B : in std_logic;
F    : out std_logic);
end My_ckt;
-- architecture
architecture My_ckt_Ar of My_ckt is
begin
F <= ( A AND B ) or (not B);
end My_ckt_Ar;
```





اكتب كود VHDL الذي ينفذ بوابة NAND ثلاثية المداخل. تتم تسمية إشارات الدخل الثلاثة A و B و C

واسم إشارة الخرج F

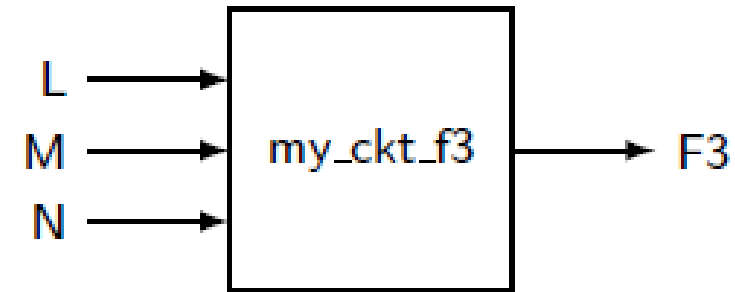
```
1 -- library declaration
2 library IEEE;
3 use IEEE.std_logic_1164.all;
4 -- entity
5 entity my_nand3 is
6 port (
7   A,B,C : in std_logic;
8   F : out std_logic);
9 end my_nand3;
10 -- architecture
11 architecture exa_nand3 of my_nand3 is
12 begin
13   F <= NOT (A AND B AND C);
14 end exa_nand3;
```





اكتب كود VHDL لتنفيذ الوظيفة المعبر عنها بالمعادلة المنطقية التالية:  $F3 = \overline{L} \overline{M} N + L M$

```
1 -- library declaration
2 library IEEE;
3 use IEEE.std_logic_1164.all;
4 -- entity
5 entity my_ckt_f3 is
6 port ( L,M,N : in std_logic;
7       F3 : out std_logic);
8 end my_ckt_f3;
9 -- architecture
10 architecture f3_2 of my_ckt_f3 is
11 begin
12   F3<= ( (NOT L) AND (NOT M) AND N) OR (L AND M) ;
13 end f3_2;
```

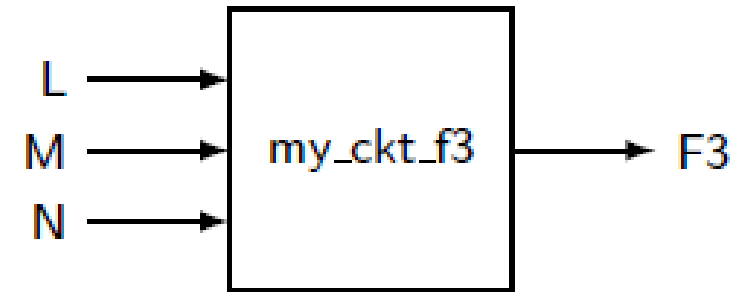




اكتب كود VHDL لتنفيذ الوظيفة المعبر عنها بالمعادلة المنطقية التالية:  $F3 = \bar{L} \bar{M} N + L M$

طريقة ثانية للحل باستخدام المتغيرات.

```
1 -- library declaration
2 library IEEE;
3 use IEEE.std_logic_1164.all;
4 -- entity
5 entity my_ckt_f3 is
6 port ( L,M,N : in std_logic;
7       F3 : out std_logic);
8 end my_ckt_f3;
9 -- architecture
10 architecture f3_1 of my_ckt_f3 is
11     signal A1, A2 : std_logic; -- intermediate signals
12 begin
13     A1 <= ((NOT L) AND (NOT M) AND N);
14     A2 <= L AND M;
15     F3 <= A1 OR A2;
16 end f3_1;
```





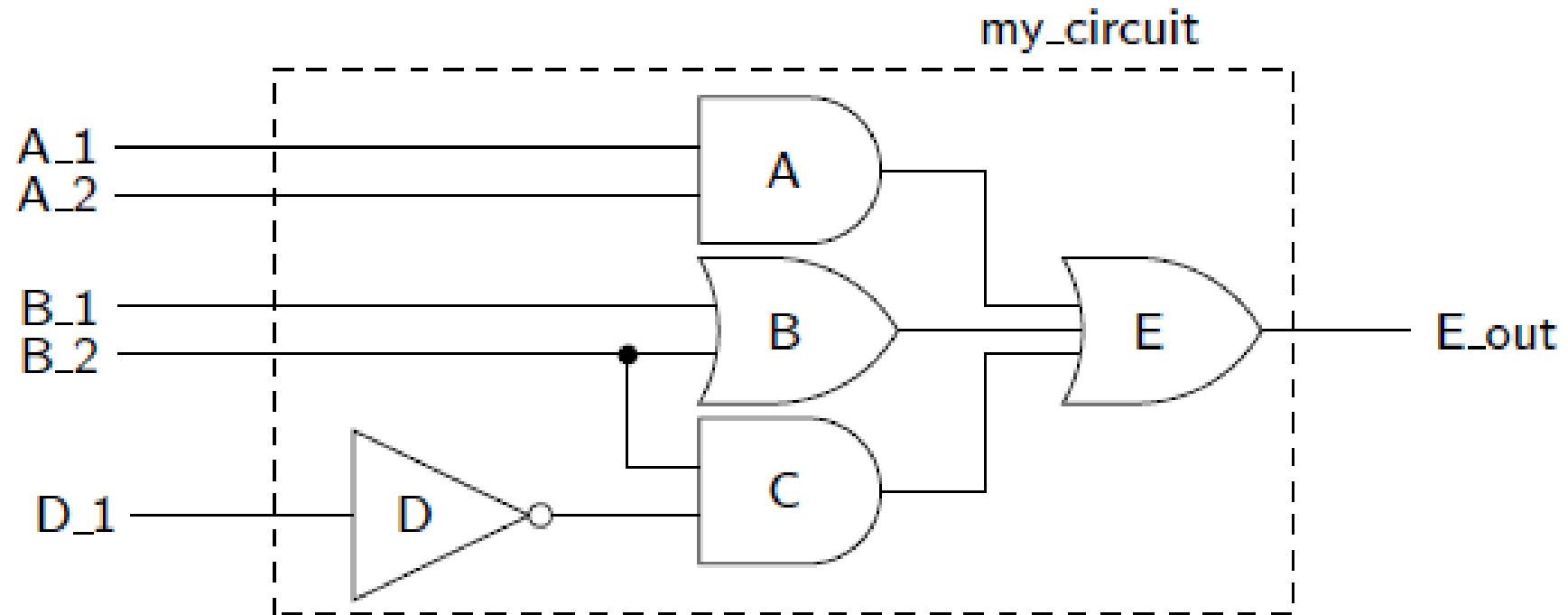
اكتب كود VHDL لتنفيذ الوظيفة المعبر عنها بالمعادلة المنطقية التالية:  $F3 = \bar{L} \bar{M} N + L M$

طريقة ثالثة للحل باستخدام الشروط.

```
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;
```

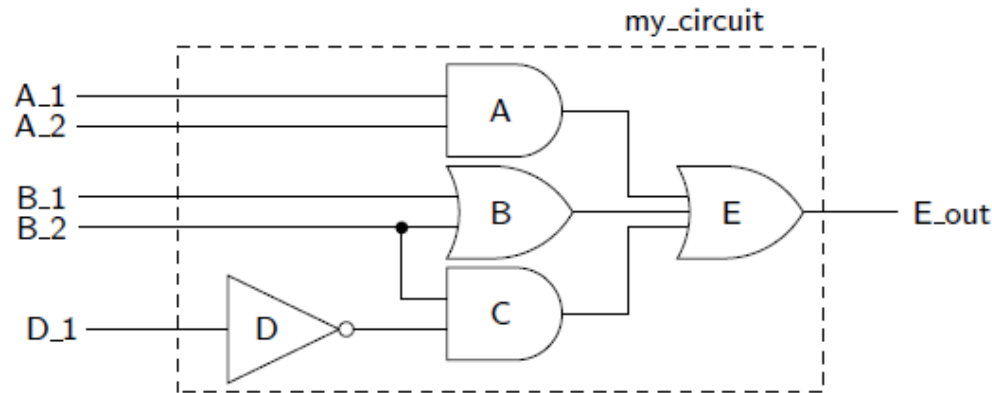


اكتب كود VHDL لتنفيذ الدارة الموضحة بالأسفل ومن بعدها محاكاة هذه الدارة على شريحة الـ FPGA





اكتب كود VHDL لتنفيذ الدارة الموضحة بالأسفل ومن بعدها محاكاة هذه الدارة على شريحة الـ FPGA



```
1 -- library declaration
2 library IEEE;
3 use IEEE.std_logic_1164.all;
4 -- entity
5 entity my_circuit is
6 port ( A_1,A_2,B_1,B_2,D_1 : in std_logic;
7       E_out : out std_logic);
8 end my_circuit;
9 -- architecture
10 architecture my_circuit_arc of my_circuit is
11 signal A_out, B_out, C_out : std_logic;
12 begin
13 A_out <= A_1 and A_2;
14 B_out <= B_1 or B_2;
15 C_out <= (not D_1) and B_2;
16 E_out <= A_out or B_out or C_out;
17 end my_circuit_arc;
```





اكتب كود VHDL لتنفيذ الوظيفة المعبر عنها بالمعادلة المنطقية التالية:

$$F5 = \overline{A} B C \overline{D} + A \overline{B} C D + A \overline{C} D + B C \overline{D}$$

✓ مرة باستخدام المتغيرات.

✓ ومرة أخرى باستخدام الشروط.





# انتهت المحاضرة