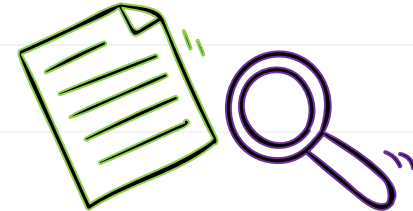
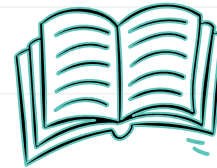




تصميم الدارات الإلكترونية بالحاسوب



Computer Design of Electronic Circuits



إعداد: د. علا جزماتي

السنة الرابعة قسم التحكم والأتمتة
العام الدراسي 2023-2024

المحاضرة الرابعة

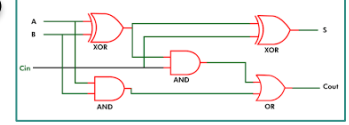
محتويات المقرر

1. مدخل إلى أهمية تطوير أدوات التصميم باستخدام الحاسب (Introduction to The Need of Developing CAD Tools)
2. تصنيف عام لأنواع أدوات التصميم (General Classification of CAD Tools Used in Electronic Systems Design)
3. مدخل إلى اللغات المستخدمة في التصميم (Introduction to Design Languages VHDL, Verilog, Verilog System, ..)
4. مدخل إلى مراحل بناء النظم الرقمية (Introduction to Digital Systems Synthesis)
5. مرحلة البناء منخفض المستوى (Low Level Synthesis)
6. تصميم الدارات المتكاملة للنظم عالية التكامل (Layout Design for VLSI Systems)
7. تطبيقات تصميمية (Design Applications)
8. اتجاهات التطور الحديثة (Trends and New Directions)

مراحل بناء النظم الرقمية باستخدام الشرائح القابلة للبرمجة

وضع التصميم

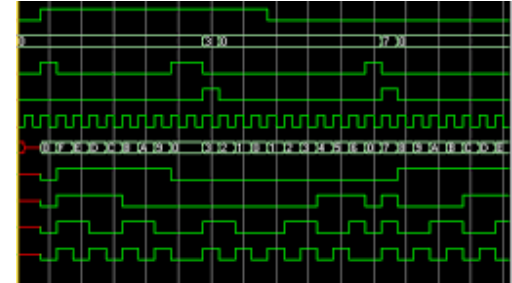
HDL
code



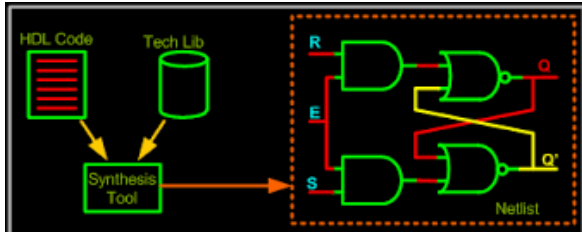
توليد ملف البرمجة

المحاكاة

تشكيل التصميم و
وضع المكونات في
أماكنها والربط بينها



01001101010
11010110101
01101111001
10110001011
0000100010
10111001010
10011101100



مراحل بناء النظم الرقمية باستخدام الشرائح القابلة للبرمجة

المستخدم

1- وضع التصميم

يعتبر الخطوة الأولى في التصميم , وهنا توجد طريقتان :

الطريقة الأولى :

رسم الهيكل البنائي للدائرة الالكترونية **Schematic** ، وهذه الطريقة غير مجدية للتصميمات كبيرة الحجم التي تحتوي على العديد من المكونات و العناصر .

الطريقة الثانية :

كتابة برنامج باستخدام إحدى لغات **HDL** و التي تصف تركيب الدائرة الالكترونية باستخدام برامج **VHDL** أو **Verilog** أو اللغات الرسومية....

مراحل بناء النظم الرقمية باستخدام الشرائح القابلة للبرمجة

المستخدم

2- المحاكاة (Simulation)

من الأفضل أن يتم اكتشاف الأعطال وإصلاحها قبل أن تتم عملية البرمجة، وبهذه الطريقة يمكن توفير الوقت اللازم لبرمجة شرائح الدارات القابلة للبرمجة لذلك نلجأ لعملية المحاكاة لاكتشاف الأعطال ومن ثم إصلاحها.

برمجيات
خاصة

3- تشكيل التصميم (Synthesis)

هي عملية اكتشاف عناصر و مكونات الدارة التي تم تصميمها بواسطة لغة التوصيف لتحويل الوصف إلى دارة الكترونية .

مراحل بناء النظم الرقمية باستخدام الدارات القابلة للبرمجة

برمجيات
خاصة

4- وضع المكونات في أماكنها والربط بينها (Place and route)

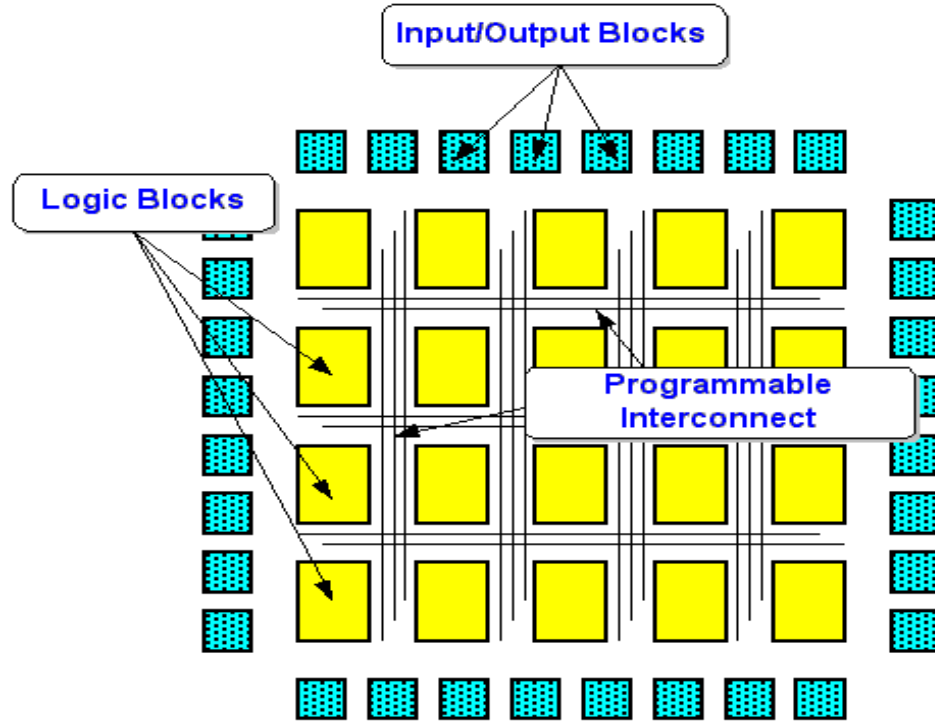
هذه الخطوة تستخدم لمقابلة الدارات المصممة بالموارد المتاحة في الدارات القابلة للبرمجة و وضع المكونات في الأماكن المناسبة في الشريحة حيث يتم ربطهم سوياً طبقاً لتصميم الدارة باستخدام قنوات التوصيل والأسلاك الداخلية، هذه الخطوة تربط كذلك بين أطراف التوصيل الخارجية للشريحة pins مع باقي أجزاء الدارة الداخلية التي سيتم توصيل الشريحة بها...

برمجيات
خاصة

5 - توليد ملف البرمجة (Generating programming file)

يحتوي ملف البرمجة على كل معلومات تصميم الدارة وكيف يتم مقابلة التصميم بالموارد الموجودة في الدارات القابلة للبرمجة وكيف ينبغي أن تتصل المفاتيح الداخلية فيها، وهو الملف الذي يستعمل لبرمجة الشريحة عادةً، وكل شريحة لها طريقة برمجة محددة ويتم تزويدها ببرنامج خاص لبرمجتها.

FPGA (Field Programmable Gate Arrays)



مصفوفات البوابات المنطقية القابلة للبرمجة حقليا :

هي عبارة عن دائرة متكاملة رقمية تحتوي على بلوكات قابلة للبرمجة مع مجموعة من الوصلات الداخلية القابلة للتعديل . يمكن تعديل هذه الوصلات (برمجة الشريحة) لتقوم الشريحة بإنجاز مهمة معينة .

FPGA (Field Programmable Gate Arrays)

تتكون البنية العامة لشرائح FPGA داخلياً:

- كتل منطقية قابلة للبرمجة موزعة على هيئة مجموعة من الخلايا المنطقية تختلف بنيتها الداخلية تبعاً للشركة الصانعة.
 - كتل التوصيلات البينية المسؤولة عن توصيل الكتل المبرمجة مع بعضها البعض و مع نقاط الدخل و الخرج .
 - نقاط الدخل و الخرج .
- الوصلات عبارة عن مقاطع سلكية و مفاتيح قابلة للبرمجة لها عدة أشكال مثل:
- (Anti fuse, SRAM, EPROM, EEPROM)، واعتماداً على الطريقة المستخدمة في تعديل الوصلات يمكن لبعض شرائح ال FPGA أن تبرمج لمرة واحدة بينما يمكن للبعض الآخر أن يبرمج مرات كثيرة .

FPGA (Field Programmable Gate Arrays)

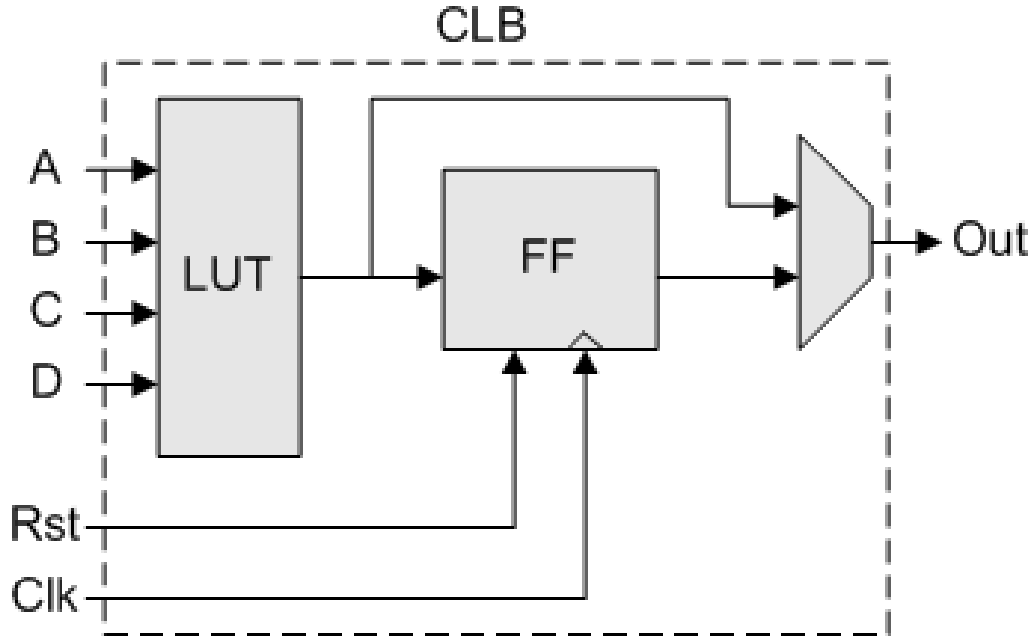
ملاحظة :

إن معنى "Field Programmable" هو أن البنية الداخلية للشريحة يمكن أن
تحدد من قبل المبرمج، خلافاً للشرائح ذات الوظائف الداخلية التي يتم بناء كيانها الصلب
FPGA من قبل الشركة الصانعة ، أي أن شريحة FPGA قابلة للبرمجة أثناء وجودها
في النظام الإلكتروني.



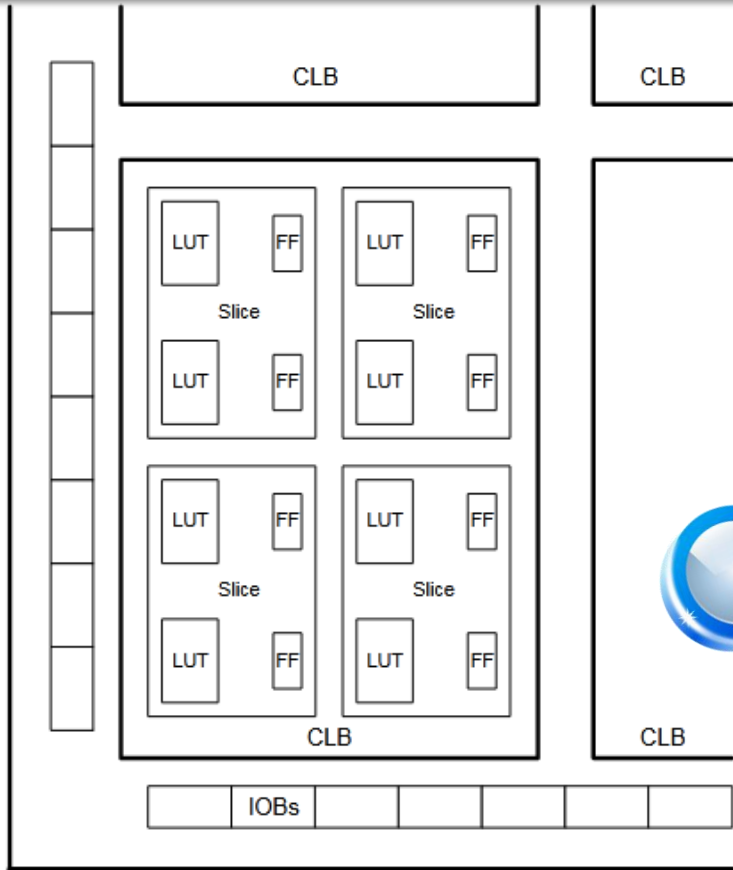
FPGA (Field Programmable Gate Arrays)

البنية العامة للوحدات الصندوقية القابلة للبرمجة

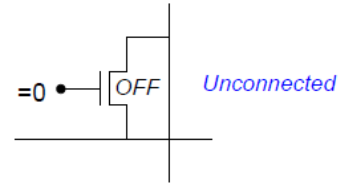
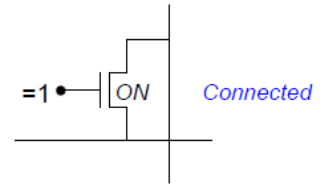
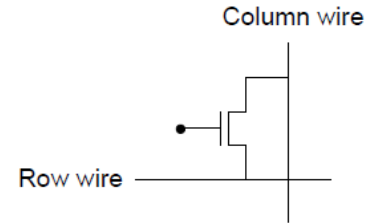


مثال

FPGA (Field Programmable Gate Arrays)



Programmable Interconnect Junction

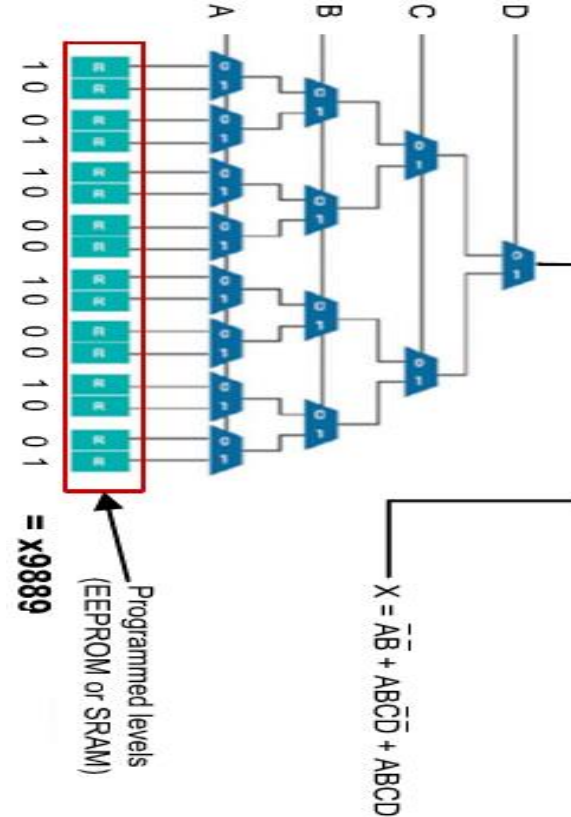


Architecture of a Spartan-3E FPGA

Look Up Table بنية

$$X = \bar{A}\bar{B} + ABC\bar{D} + ABCD$$

A	B	C	D	X
0	0	0	0	1
1	0	0	0	0
0	1	0	0	0
1	1	0	0	1
0	0	1	0	1
1	0	1	0	0
0	1	1	0	0
1	1	1	0	0
0	0	0	1	1
1	0	0	1	0
0	1	0	1	0
1	1	0	1	0
0	0	1	1	1
1	0	1	1	0
0	1	1	1	0
1	1	1	1	1



FPGA (Field Programmable Gate Arrays)

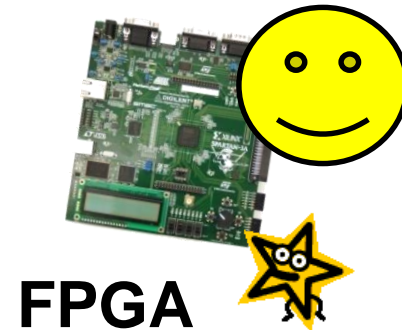
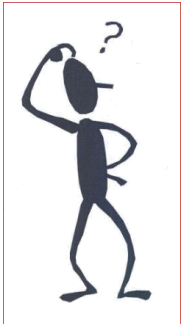
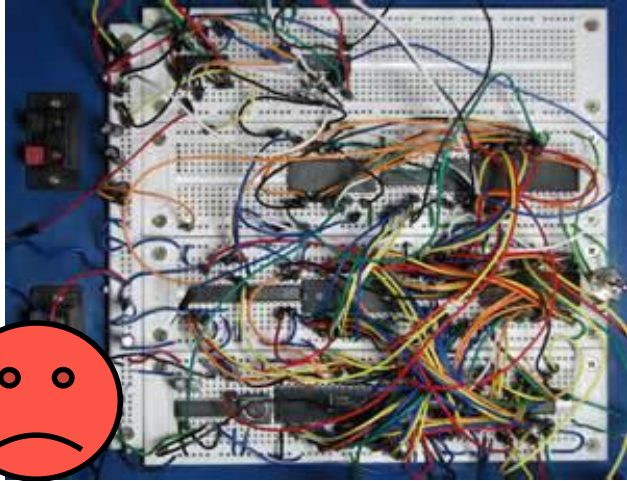
مقارنة

FPGA OR MCU?



FPGA (Field Programmable Gate Arrays)

مقارنة



FPGA

FPGA (Field Programmable Gate Arrays)

أهمية FPGA

التطوير

الأداء

استهلاك
الطاقة

التكلفة

الموثوقية

FPGA (Field Programmable Gate Arrays)

أهمية FPGA

- سهولة عملية تعديل التصميم حيث يمكن إعادة البرمجة ضمن النظام نفسه .
- إمكانية بناء نظام كامل (hardware + software) على الشريحة دون الحاجة لدفع التكاليف الكبيرة المطلوبة لبناء نفس النظام باستخدام الشرائح المصنعة لتطبيقات مخصصة ASIC .
- معالجة عالية السرعة للإشارة حيث يمكن استخدام FPGA بدلاً من DSP ، كذلك FPGA أسرع من المتحكمات المصغرة microcontrollers لأنها تعتمد مبدأ البرمجة التفرعية، كما أن أبسط شريحة FPGA تملك أكثر بكثير من الإمكانيات المطلوبة لتقوم بعمل المتحكم المصغر بالإضافة إلى الموثوقية العالية في الأداء.

FPGA (Field Programmable Gate Arrays)

يوجد عدة أدوات لأتمتة التصميم الإلكتروني (Electronic Design Automation EDA) متاحة من أجل:

- تحليل التصميم (أي تحويل التصميم إلى دارة إلكترونية).
- بناء الدارة و وضع المكونات في أماكنها و الربط بينها.
- محاكاة عمل الدارة .

و ذلك باستخدام لغة vhdl أو verilog مثل البيئات البرمجية التالية:

- [Xilinx ISE \(Integrated Software Environment\)](#) for Xilinx FPGAs,
- [Quartus II](#) for Altera devices,
- [Libero](#) for Actel devices.

AMD

Intel

MicroChip

لمحة تاريخية



The first FPGA was Introduced by Xilinx in 1985, the XC2064 (which was created at the $2\mu\text{m}$ technology node) contained an $8 \times 8 = 64$ array of logic blocks

45nm

28nm

20nm

16nm

SPARTAN⁶

VIRTEX⁷

VIRTEX⁷
UltraSCALE

VIRTEX⁷
UltraSCALE⁺



Increasing Performance & Functionality

تصميم فاك تشفير لوحدة إظهار ذات السبع قطع – مصعد مشترك

```
module seg7( input [3:0] data_in, output reg [7:0] display_out );
```

```
always @(*)
```

```
begin
```

data_in [3:0]

seg7

display_out [7:0]

```
case (data_in)
```

```
4'b0000: display_out = 8'b00000011; //a,b,c,d,e,f,g,dot (zero)
```

```
4'b0001: display_out = 8'b10011111; //one
```

```
4'b0010: display_out = 8'b00100101; //two
```

```
4'b0011: display_out = 8'b00001101; //three
```

```
4'b0100: display_out = 8'b10011001; //four
```

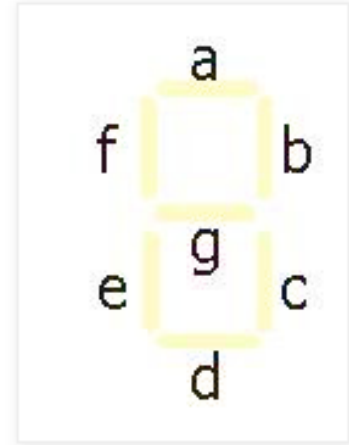
```
4'b0101: display_out = 8'b01001001; //five
```

```
4'b0110: display_out = 8'b01000001; //six
```

```
4'b0111: display_out = 8'b00011111; //seven
```

```
4'b1000: display_out = 8'b00000001; //eight
```

```
4'b1001: display_out = 8'b00001001; //nine
```



تصميم فاك تشفير لوحدة إظهار ذات السبع قطع – مصعد مشترك

```
4'b1010: display_out = 8'b00010001; //A
```

```
4'b1011: display_out = 8'b11000001; //b
```

```
4'b1100: display_out = 8'b01100011; //C
```

```
4'b1101: display_out = 8'b10000101; //d
```

```
4'b1110: display_out = 8'b01100001; //E
```

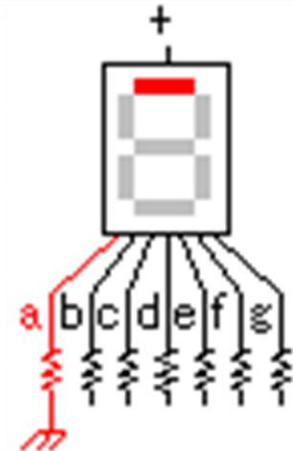
```
4'b1111: display_out = 8'b01110001; //F
```

```
endcase
```

```
end
```

```
endmodule
```

حاول أن تكتب كود محاكاة لهذا
التوصيف.



الأسئلة والمناقشة

