تصميم رقمي متقدم
Advanced Digital Design

Dr.-Eng. Samer Sulaiman

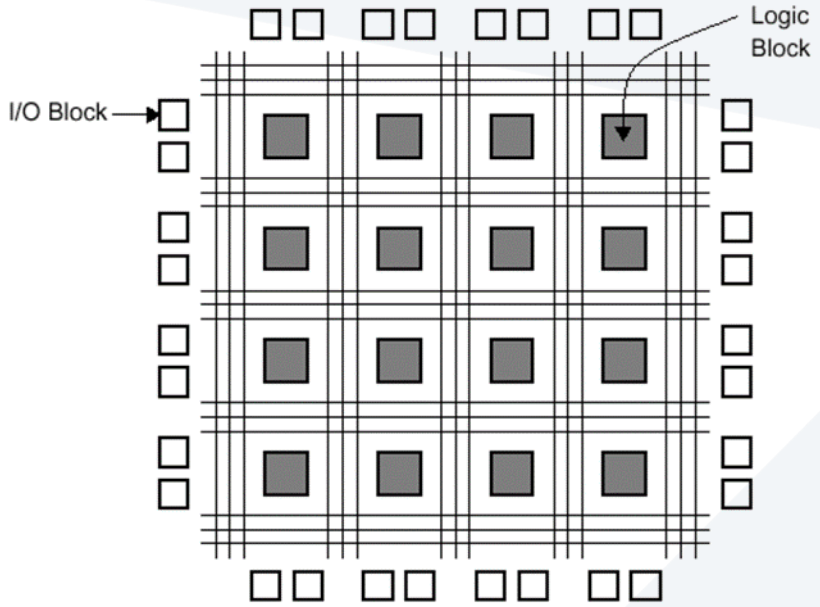
2021-2022

مفردات المنهاج

- أساسيات التصميم الرقمي
- عناصروتقنيات التصميم الرقمي التو افقي والتعاقبي (المتسلسل)
- نمذجة التصميم الرقمي باستعمال لغة توصيف الكيان الصلب VHDL
- المحاكاة الوظيفية والزمنية للأنظمة الرقمية

تصميم الأنظمة الرقمية باستخدام VHDL

- شرائح مصفوفات البوابات المنطقية القابلة للبرمجة حقلياً FPGA:
  - اختصار للعبارة التالية (Field Programmable Gate Arrays)
  - أي مصفوفات من البوابات المنطقية القابلة للبرمجة حقليًا
  - تحتوي على بلوكات قابلة للبرمجة مع مجموعة من الوصلات الداخلية القابلة للتعديل برمجياً.
  - تتكون البنية العامة لشرائح الـFPGA داخليا :
    - كتل منطقية قابلة للبرمجة موزعة على هيئة مجموعة من الخلايا المنطقية تختلف بنيتها الداخلية تبعا للشركة الصانعة.
    - كتل التوصيلات البينية المسؤولة عن توصيل الكتل المبرمجة مع بعضها البعض ومع نقاط الدخل والخرج
    - نقاط الدخل والخرج



- أهمية شرائح الـFPGA:
  - 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 بدلا من الـDSP، كذلك الـFPGA أسرع من المتحكمات الصغرية microcontrollers لأنها تعتمد مبدأ البرمجة التفرعية، كما أن أبسط شريحة FPGA تملك أكثربكثير من الإمكانيات المطلوبة لتقوم بعمل المتحكم الصغري بالإضافة إلى الموثوقية العالية في الأداء.

تصميم الأنظمة الرقمية باستخدام VHDL

- مراحل تصميم الأنظمة الرقمية باستخدام الـFPGA:
  - وضع التصميم :
    - يعد الخطوة الأولى في التصميم، وهنا توجد طريقتان:
      - الطريقة الأولى رسم الهيكلية البنائية للدارة الرقمية Schematic Entry، وهذه الطريقة غير مجدية للتصميمات كبيرة الحجم التي تحتوي على العديد من المكونات و العناصر
      - الطريقة الثانية كتابة برنامج باستخدام إحد لغات الـHDL و التي تصف ترتيب الدارة الرقمية باستخدام برامج مثل VHDL أو .... Verilog
  - المحاكاة (Simulation):
    - من الأفضل أن يتم اكتشاف الأعطال واصلاحها قبل أن تتم عملية البرمجة، وبهذه الطريقة يمكن توفير الوقت اللازم لبرمجة شرائح الـFPGA لذلك نلجأ لعملية المحاكاة لاكتشاف الأعطال ومن ثم إصلاحها.
  - التشكيل (Synthesis):
    - هي عملية اكتشاف عناصر و مكونات الدارة التي تم تصميمها بواسطة لغة الـVHDL لتحويل الوصف إلى دارة رقمية
  - وضع المكونات في أماكنها والربط بينها (Place and route):
    - هذه الخطوة تستخدم لمقابلة الدارات المصممة بالموارد المتاحة في الـFPGA، ووضع المكونات في الأماكن المناسبة في الشريحة حيث يتم ربطهم سويا طبقا 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.
  - توليد ملف البرمجة (Programming File): يحتوي ملف البرمجة
    - على كل معلومات تصميم الدارة وكيف يتم مقابلة التصميم بالموارد الموجودة في FPGA
    - كيف ينبغي أن تتصل المفاتيح الداخلية فيها،
    - عبارة عن الملف الرقمي (bit stream) الذي يستعمل لبرمجة الشريحة
    - كل شريحة لها طريقة برمجة محددة ويتم تزويدها ببرنامج خاص لبرمجتها .

تصميم الأنظمة الرقمية باستخدام VHDL

- اساسيات البرمجة بلغة الـ VHDL:
  - مصطلح VHDL هو اختصار للعبارة : VHSIC HDL VHSIC (Very High Speed Integrated Circuits) Hardware Description Language
  - تعني لغة توصيف الكيان الصلب للدارات المتكاملة ذات السرعات المرتفعة جدا
  - تعدّ لغة الـVHDL لغة برمجة قياسية صممت من قبل وزارة الدفاع الأمريكية عام 1980، حيط تستعمل في توصيف سلوك و محاكاة عمل الأنظمة الرقمية (ابتداءً من البوابات البسيطة و انتهاءً بأعقد الأنظمة الرقمية)، و تصميم ومحاكاة دارات الـ VHSIC
  - تم اتخاذها لغة قياسية معتمدة من قبل IEEE في عام 1987، و تمت مراجعتها في الولايات المتحدة الأميركية عام 1993
  - في عام 1999 ظهر امتداد للغة VHDL وهو VHDL – AMS (Analog Mixed Signal)،
  - في عام 2008 تم إضافة خصائص جديدة ضمن المعيار IEEE Std 1076 – 2008
  - يوجد نوعان من الأدوات التي تتعامل مع VHDL:
    - المحاكاة Simulation: لاختبار التصميم المنطقي باستخدام نماذج المحاكاة، ويمكن استخدام كل تعليمات VHDL
    - التركيب Synthesis: لتحويل الشيفرات إلى كيان صلب hardware

تصميم الأنظمة الرقمية باستخدام VHDL

- اساسيات البرمجة بلغة الـVHDL:
  - مراحل توصيف الأنظمة الرقمية باستخدام اللغة VHDL:
    - التصريح عن المكتبات :
      - استدعاء مكتبات تحتوي العناصر و التعليمات الأساسية والقياسية المستخدمة في التصميم مثل . ieee , std , work
    - وحدة التوصيف الخارجي Entity:
      - فيها يتم توصيف المداخل و المخارج
    - وحدة التوصيف الداخلي Architecture:
      - تخصص هذ الوحدة للتعريف بسلوك أو بنية الدارة أو النظام المطلوب توصيفه انطلاقا من إشارات الدخل وصولا إلى إشارات الخرج وفق العلاقات الخاصة بالنظام
  - طريقة كتابة شيفرة (كود) الـVHDL:
    - استدعاء المكتبات :
      - للتصريح عن مكتبة (أي لجعلها مرئية بالنسبة إلى التصميم) نكتب ما يلي :

- LIBRARY library_name;
  USE library_name.package_name.package_parts;

- مثال:

- library IEEE;
- use IEEE.STD_LOGIC_1164.ALL;

- بعض المكتبات تكون مرئية للتصميم افتراضيا default، لذلك لسنا بحاجة للتصريح عنها:
  - Std: مكتبة المصادر و تنسيق النصوص و بعض أنواع المعطيات، وذلك من أجل بيئة تصميم vhdl
  - Work: هي المكتبة التي نخزن فيها تصميمنا (ملف البرنامج) بالإضافة إلى كل الملفات المنشأة بوساطة برنامج الترجمة أو المحاكاة
- عادةً ما تتضمن المكتبات عدة أجزاء و أنماط معطيات وثوابت وعناصر جاهزة و توابع و إجرائيات ...

تصميم الأنظمة الرقمية باستخدام VHDL

- اساسيات البرمجة بلغة الـVHDL:
  - طريقة كتابة شيفرة (كود) الـVHDL:
    - استدعاء المكتبات :
      - تصنيف أنواع المعطيات المعرفة ضمن اللغة VHDL وفق المكتبات المتوفرة:
      - إن لغة VHDL تحتوي عدة أنماط للبيانات مسبقة التعريف و المحددة من خلال المعايير IEEE1076 و IEEE1164
        - مكتبة IEEE: تحتوي النظام المنطقي متعدد المستويات و أنواع المعطيات std _logic , std_ logic_vector
        - مكتبة STD: تحتوي أنواع المعطيات Bit ,Boolean ,Integer ,Real ,Signed ,Unsigned والعمليات الحسابية و التوابع الأساسية للتحويل بين الأنماط.
        - المنطق المعياري STD_LOGIC: نوع من المعطيات متعدد القيم ذو 9 مستويات موضحة بالجدول التالي :

القيمة المنطقية Logic Value	الوصف Description
U	Uninitialized
X	Forcing Unknown
0	Forcing Logic Low
1	Forcing Logic High
Z	Tri-state(High-Impedance)
W	Weak Unknown
L	Weak Low
H	Weak High
-	Don't Care

	X	0	1	Z	W	L	H	-
X	X	X	X	X	X	X	X	X
0	X	0	X	0	0	0	0	X
1	X	X	1	1	1	1	1	X
Z	X	0	1	Z	W	L	H	X
W	X	0	1	W	W	W	W	X
L	X	0	1	L	W	L	W	X
H	X	0	1	H	W	W	H	X
-	X	X	X	X	X	X	X	X

        - عندما يحدث تعرض بين إشارتين موصولتين إلى نفس العقدة سيتم حله آليا وفق الشكل التالي:

تصميم الأنظمة الرقمية باستخدام VHDL

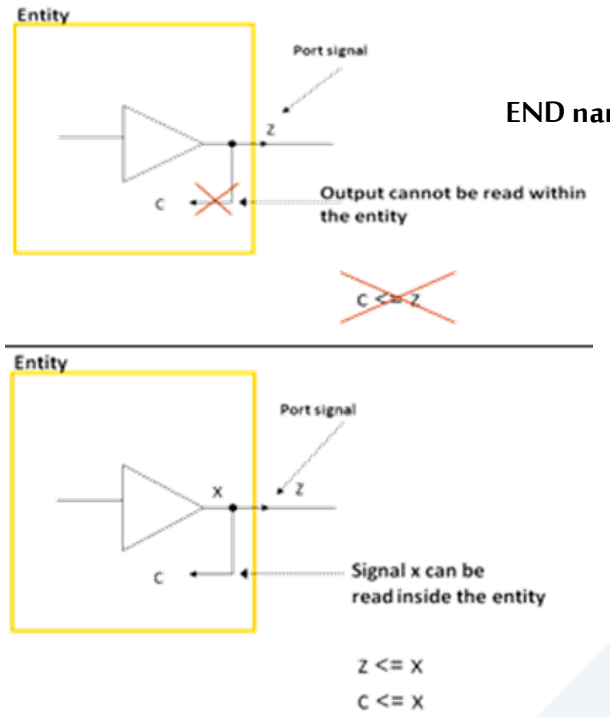
- اساسيات البرمجة بلغة الـ VHDL:
  - طريقة كتابة شيفرة (كود) الـ VHDL:
    - استدعاء المكتبات :
      - بعض الملاحظات المهمة عند كتابة شيفرة VHDL:
        - يمكن استخدم دائما نوع المعطيات std_logic أو std_logic_vector من أجل كل منافذ الدخل والخرج، أما باقي أنواع المعطيات يمكن استخدامها داخل وحدة التوصيف الداخلي عند الحاجة أو مع الثوابت العامة.
      - وحدة التوصيف الخارجي Entity:
        - فيها يتم توصيف المداخل والمخارج وفق النمط التالي

```
ENTITY entity_name IS
PORT (
        port_name : port_mode signal_type;
        port_name : port_mode signal_type;
        ..............
        port_name : port_mode signal_type);
END entity_name;
```

تصميم الأنظمة الرقمية باستخدام VHDL

- اساسيات البرمجة بلغة الـVHDL:
  - طريقة كتابة شيفرة (كود) الـVHDL:
    - وحدة التوصيف الخارجي Entity:
      - مثال: بو ابة NAND بمدخلين

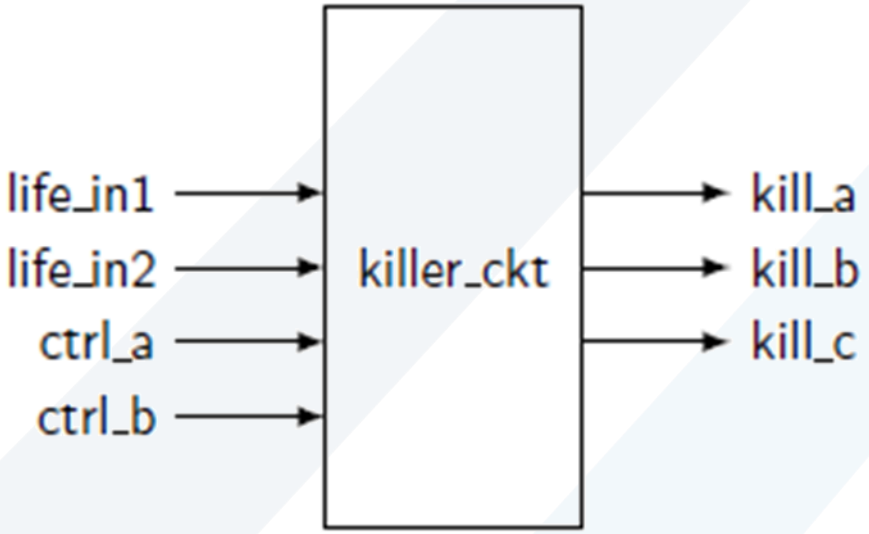
- ENTITY nand_gate IS
  PORT(a : IN STD_LOGIC;
  b : IN STD_LOGIC;
  z : OUT STD_LOGIC);
  END nand_gate;



- أنواع أنماط الإشارة التي تصف اتجاه نقل البيانات بالنسبة إلى العنصر الموصوف
  - دخل IN: يشير إلى أن الإشارة المطبقة على هذا القطب تتجه إلى داخل النظام
  - خرج OUT: يشير إلى أن الإشارة الناتجة من هذا القطب تتجه إلى خارج النظام
    - لا يمكن قراءة الخرج ضمن الـEntity لذلك يتم استخدام تعريف الإشارات كما هو موضح بالشكل
  - ثنائي الاتجاه INOUT: يشير إلى أن الإشارة المطبقة على هذا القطب ثنائية الاتجاه يمكن القراءة منها أو الكتابة عليها
  - عازل Buffer: يشير إلى أن الإشارة المطبقة على هذا القطب تشكل إشارة يمكن قرائتها من قبل وحدات توصيف خارجي اخر(أي توصيل بين الوحدات وليس من الدخل إلى الخرج او بالعكس)
    - هنا يمكن قراءة العازل ضمن الـEntity

تصميم الأنظمة الرقمية باستخدام VHDL

- اساسيات البرمجة بلغة الـVHDL:
  - طريقة كتابة شيفرة (كود) الـVHDL:
    - وحدة التوصيف الخارجي Entity:
      - مثال: أكتب شيفرة الـVHDL الخاصة بالدارة التالية:



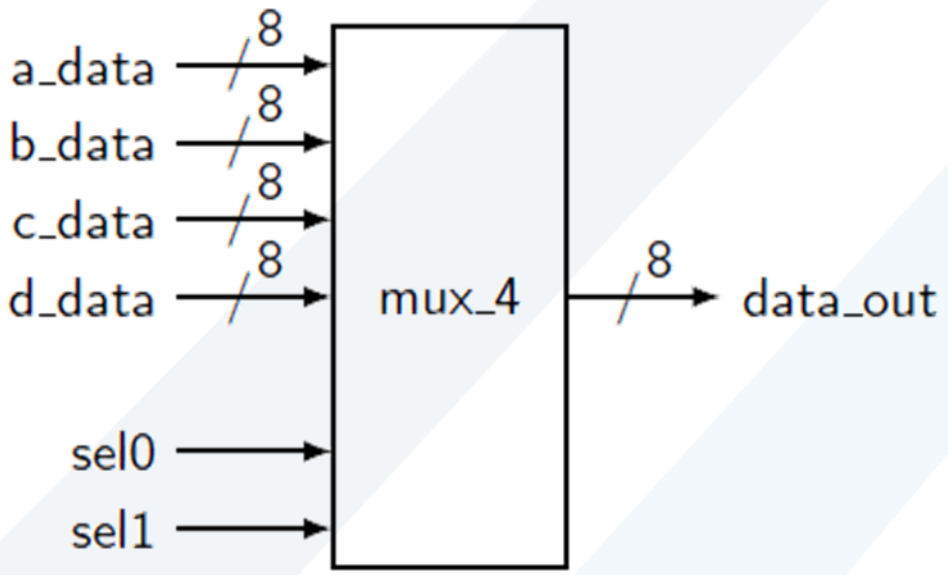
تصميم الأنظمة الرقمية باستخدام VHDL

- اساسيات البرمجة بلغة الـ VHDL:
  - طريقة كتابة شيفرة (كود) الـ VHDL:
    - وحدة التوصيف الخارجي Entity:
      - مثال: أكتب شيفرة الـ VHDL الخاصة بالدارة التالية:
        - الحل:

```
entity killer_ckt is
  port (
            life_in1 : in std_logic;
            life_in2 : in std_logic;
            crtl_a, ctrl_b : in std_logic;
            kill_a : out std_logic;
            kill_b, kill_c : out std_logic);
end killer_ckt;
```

تصميم الأنظمة الرقمية باستخدام VHDL

- اساسيات البرمجة بلغة الـVHDL:
  - طريقة كتابة شيفرة (كود) الـVHDL:
    - وحدة التوصيف الخارجي Entity:
      - مثال: أكتب شيفرة الـVHDL الخاصة بالدارة التالية:



تصميم الأنظمة الرقمية باستخدام VHDL

- اساسيات البرمجة بلغة الـ VHDL:
  - طريقة كتابة شيفرة (كود) الـ VHDL:
    - وحدة التوصيف الخارجي Entity:
      - مثال: أكتب شيفرة الـ VHDL الخاصة بالدارة التالية:
        - الحل:

```
entity mux4 is
  port ( a_data : in std_logic_vector(0 to 7);
         b_data : in std_logic_vector(0 to 7);
         c_data : in std_logic_vector(0 to 7);
         d_data : in std_logic_vector(0 to 7);
         sel1,sel0 : in std_logic;
         data_out : out std_logic_vector(0 to 7));
end mux4;
```

تصميم الأنظمة الرقمية باستخدام VHDL

- اساسيات البرمجة بلغة الـVHDL:
  - طريقة كتابة شيفرة (كود) الـVHDL:
    - وحدة التوصيف الداخلي Architecture:
      - عبارة عن وصف يعبرعن كيفية عمل الدارة أي توضيح لسلوك الدارة
      - طرق توصيف النظام داخليا
        - توصيف دارة ما من خال استدعاء عدد من المكونات الأولية والربط فيما بينها لتشكيل البنية المطلوبة
        - توصيف البنية الداخلية لهذه الدارة من خلال اتباع مخطط منهجي يوضح آلية العمل المطلوبة بغض النظرعن البنية
        - توصيف نظام منطقي ما من خال جملة معادلات منطقية،
        - كما يمكن المزج بين الطرق السابقة جميعها .
      - إذا يمكن توصيف النظام داخليا بعدة طرق :
        - توصيف بنيوي .
        - توصيف سلوتي .
        - توصيف رياضي منطقي .
        - توصيف مختلط .
      - يتم التصريح عنه بالشتل التالي :

- ARCHITECTURE architecture_name OF entity_name IS
  [declarations]
  BEGIN
  code
  END architecture_name;

تصميم الأنظمة الرقمية باستخدام VHDL

- اساسيات البرمجة بلغة الـ VHDL:
  - طريقة كتابة شيفرة (كود) الـ VHDL:
    - وحدة التوصيف الداخلي Architecture:
      - مثال: بوابة NAND:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY nand_gate IS
PORT(
        a : IN STD_LOGIC;
        b : IN STD_LOGIC;
        z : OUT STD_LOGIC);
END nand_gate;
ARCHITECTURE dataflow OF nand_gate IS
BEGIN
        z <= a NAND b;
END dataflow;
```