



# Introduction to VHDL

---



## Agenda

---

- Introduction
- Data Type
- Example



# Introduction

---

- VHDL
  - VHSIC Hardware Description Language
  - พัฒนาโดย IBM, TI, Intermetrics ในปี 1993
  - IEEE Std 1076 – 1987 and 1993
- VHSIC = Very High Speed Integrated Circuits
- History
  - Department of Defense (DOD) program in 1981
  - Publication of first VHDL standard 1987 (VHDL'87)
  - Publication of revised VHDL standard 1993 (VHDL'93)

3



# Introduction

---

- VHDL เป็นภาษาที่อธิบายลักษณะการทำงานของฮาร์ดแวร์ (Hardware Description Language) ไม่ใช่โปรแกรมภาษา (Programming language)
- ภาษาที่เป็นคู่แข่งของภาษา VHDL (Ada-based) คือภาษา Verilog (C-based)
- สามารถนำไปใช้กับดีจิตอลฮาร์ดแวร์ต่าง ๆ เช่น
  - PLDs, CPLDs, FPGAs

4



# Introduction

---

- ประโยชน์ของภาษา VHDL
  - เป็นภาษามาตรฐานสากลโดยรับรองจากสถาบัน IEEE ทำให้มีโปรแกรมและเครื่องมือต่าง ๆ มากมาย
  - เป็นภาษาที่เป็นมาตรฐาน ที่ภาคอุตสาหกรรมนำไปใช้ออกแบบ
  - เป็นภาษาที่สามารถใช้ได้หลายระบบ จึงเป็นการออกแบบที่ไม่ยึดติดกับซอฟต์แวร์ที่ใช้ในการออกแบบ
  - เป็นภาษาที่สามารถจำลองรูปแบบการทำงานของวงจรระดับมาโครบล็อก (Macro block) จึงถึงระดับเกต (Gate)

5



# Introduction

---

- ประโยชน์ของภาษา VHDL
  - สามารถออกแบบวงจรที่มีความซับซ้อนสูงและมีขนาดใหญ่มากได้
  - เป็นภาษาที่สามารถนำกลับมาใช้ใหม่ได้ คล้ายคลึงกับการโปรแกรม
  - เป็นภาษาที่สามารถนำไปใช้เป็นเอกสารประกอบการออกแบบ

6



# Introduction

---

- ลักษณะการใช้งานภาษา VHDL
  - Document Language : เป็นภาษาที่ใช้บรรยายรายละเอียดการทำงานของวงจรที่ออกแบบ
  - Design Language : เป็นภาษาที่ใช้สำหรับออกแบบวงจรที่มีความซับซ้อนสูง
  - Verification Language : เป็นภาษาที่ใช้ตรวจสอบความถูกต้องของวงจรที่ออกแบบ
  - Test Language : เป็นภาษาที่ใช้สำหรับทดสอบการทำงานของวงจรที่ออกแบบ
  - Synthesis Language : เป็นภาษาที่ใช้สำหรับสังเคราะห์วงจร ( Synthesis ) จริง

7



# Introduction

---

- ข้อกำหนดของภาษา VHDL
  - สามารถออกแบบได้หลายระดับ จาก Behavioral to Gate Level
  - สนับสนุนการออกแบบแบบลำดับชั้น (Hierarchy Design)
  - สนับสนุนระบบไลบรารี (Library Support)
  - สามารถควบคุมเวลาได้
  - สามารถทำงานแบบ Concurrent และ Sequential
  - สามารถกำหนด Type ของสัญญาณหรือข้อมูลได้

8

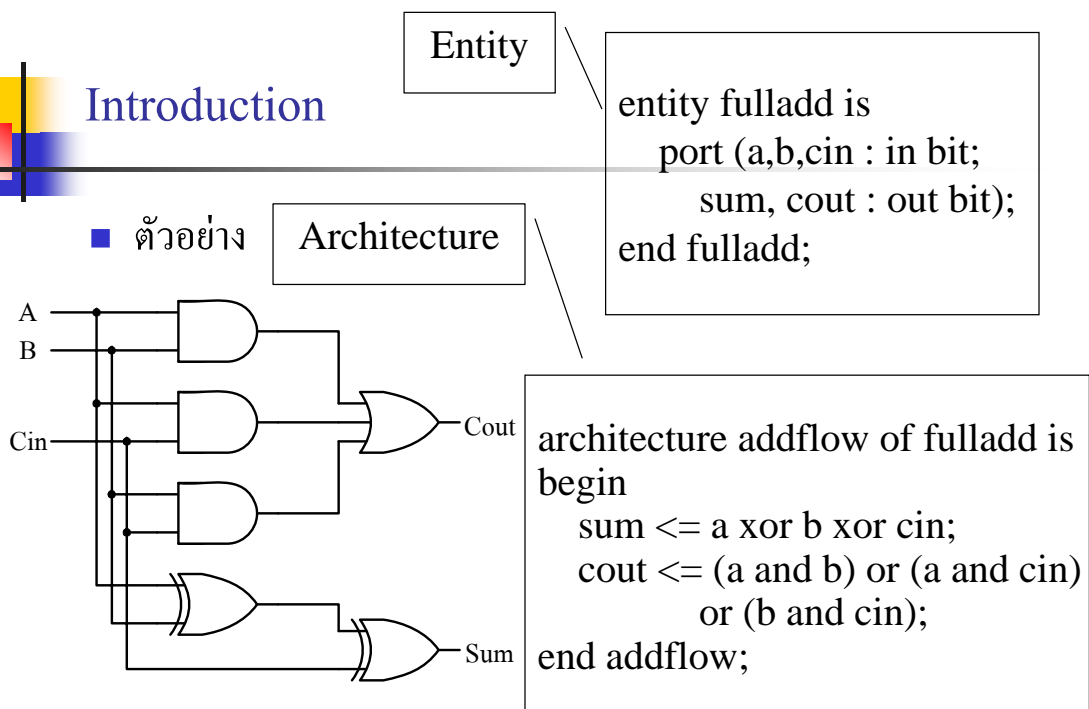
# Introduction

- ส่วนประกอบของภาษา VHDL
  - Entity ใช้บรรยายถึงส่วน Interface ของอุปกรณ์
  - Architecture ใช้บรรยายถึงฟังก์ชันและ Timing ของอุปกรณ์
    - Concurrent
    - Sequential

9

# Introduction

- ตัวอย่าง



10



# Introduction

---

- การประกาศ Entity

```
ENTITY entity_name IS  
    generic clause  
    port clause  
END entity_name;
```

```
ENTITY and3 IS  
    GENERIC (delay : TIME := 5 ns);  
    PORT (a, b, cin : in bit; sum, cout : out bit);  
END and3;
```

11



# Introduction

---

- Port หมายถึงช่องสัญญาณที่ใช้สำหรับติดต่อกับ Component อื่น
- ประกอบไปด้วย
  - Name
  - Mode : IN, OUT, INOUT, BUFFER
  - Type : BIT, BIT\_VECTOR, BOOLEAN, INTEGER, REAL, TIME, CHARACTER, STRING

12



# Introduction

---

- การประกาศ ARCHITECTURE

```
ARCHITECTURE arch_name OF entity_name IS
    architecture declarative part
BEGIN
    concurrent statement
END arch_name;
```

13



# Introduction

---

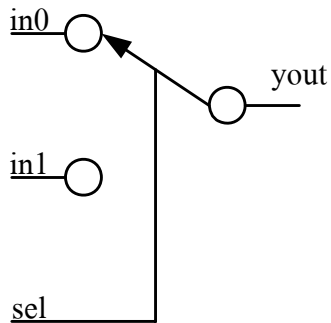
- ลักษณะของการเขียนอธิบาย Architecture
  - แบบอธิบายพฤติกรรม (Behavioral Description) คล้ายกับการเขียนโปรแกรมระดับสูง
  - แบบอธิบายการไหลของข้อมูล (Dataflow Description) คล้ายกับการเขียนสมการลอจิก
  - แบบอธิบายโครงสร้าง (Structure Description) คล้ายกับการต่ออุปกรณ์
  - แบบผสม (Mixed Model Description)

14

# Introduction

- แบบอธิบายพฤติกรรม

ARCHITECTURE behavioral OF Mux2x1 IS  
BEGIN

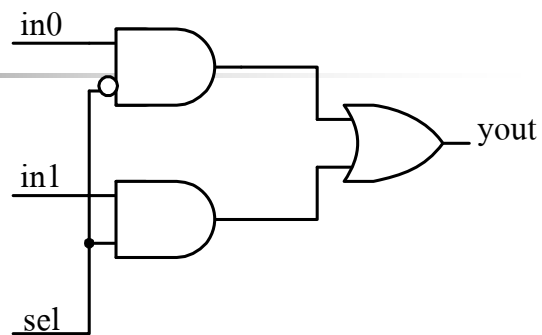


```
PROCESS (in0,in1,sel)
BEGIN
    IF sel = 0 THEN
        yout <= in0;
    ELSE
        yout <= in1;
    END IF;
END PROCESS;
```

END behavioral;

15

# Introduction



- แบบอธิบายการไหลของข้อมูล

ARCHITECTURE dataflow OF Mux2x1 IS  
BEGIN

```
yout <= ((NOT sel) AND in0) OR (sel AND in1)
END dataflow;
```

16



## ■ แบบอธิบายโครงสร้าง

ARCHITECTURE structural OF Mux2x1 IS

Component INV

PORT (i1 : in bit; y1 : out bit);

End Component

Component AND2

PORT (iand1, iand2 : in bit; yand : out bit);

End Component

Component OR2

PORT (ior1, ior2 : in bit; yor : out bit);

End Component

SIGNAL s0, s1, s2 : bit

BEGIN

IC1 : INV PORT MAP(i1=>sel, y1=>s0);

IC2 : AND2 PORT MAP(iand1=>in0, iand2=>s0; yand=>s1);

IC3 : AND2 PORT MAP(iand1=>in1, iand2=>sel; yand=>s2);

IC4 : OR2 PORT MAP(ior1=>s1, ior2=>s2, yor=>yout);

END structural;

17

## Introduction

### ■ คำสั่งภายใน Architecture

ARCHITECTURE arch\_name OF entity\_name IS

BEGIN

concurrent statements;

concurrent statements;

.....

PROCESS

BEGIN

Sequential statement;

Sequential statement;

.....

END PROCESS;

END arch\_name;

1 Process =  
1 Concurrent  
Statement

18



# Data Type Classification

- Scalar Types :
  - สามารถเก็บค่าได้เพียงค่าเดียว เช่น Integer, Real, Bit, Enumerated
- Composite Types :
  - สามารถเก็บข้อมูลได้เป็นกลุ่ม เช่น Array, Record
- Other Types:
  - File

19



# Data Type Classification

- Data Type 'time'
  - ใช้สำหรับตอนทำการกำหนด gate delays และ ขณะทำ test benches
  - หน่วยที่สามารถใช้ได้ คือ fs, ps, ns, us, ms, sec, min, hr

architecture Example of Time\_Type is  
signal CLK : bit := 0;  
constant PERIOD : time := 50 ns;  
begin  
process  
begin  
wait for PERIOD;  
.....  
wait for PERIOD \* 5.5;  
.....  
end process;  
.....

20



## Data Type Classification

- ARRAY
  - Bit\_vector (array of bit)
  - String (array of character)

- Example

```
signal A_BUS,Z_BUS :
bit_vector (3 downto 0);
```

```
Z_BUS <= A_BUS
Z_BUS(3)  <- A_BUS(3)
Z_BUS(2)  <- A_BUS(2)
Z_BUS(1)  <- A_BUS(1)
Z_BUS(0)  <- A_BUS(0)
```

```
Z_BUS(3) <= A_BUS(0)
Z_BUS(3)  A_BUS(3)
Z_BUS(2)  A_BUS(2)
Z_BUS(1)  A_BUS(1)
Z_BUS(0)  A_BUS(0)
```

21



## Data Type Classification

- Example

```
signal A_BUS: bit_vector (3 downto 0);
signal Z_BUS: bit_vector (0 to 3);
```

```
Z_BUS <= A_BUS
Z_BUS(3)  <- A_BUS(0)
Z_BUS(2)  <- A_BUS(1)
Z_BUS(1)  <- A_BUS(2)
Z_BUS(0)  <- A_BUS(3)
```

22



## Integer and bit

architecture example of datatype is

```
signal SEL : bit;
signal A,B,Z : integer range 0 to 3;
begin
  A <= 2;
  B <= 3;
  process (SEL,A,B)
  begin
    if SEL = '1' then
      z <= A;
    else
      z <= B;
    end if;
  end process;
End EXAMPLE;
```

architecture example of datatype is

```
signal SEL : bit;
signal A,B,Z : bit_vector (0 to 1);
begin
  A <= "01";
  B <= "11";
  process (SEL,A,B)
  begin
    if SEL = '1' then
      z <= A;
    else
      z <= B;
    end if;
  end process;
End EXAMPLE;
```

23



## bit data type

architecture EXAMPLE of ASSIGNMENT is

```
signal Z_BUS : bit_vector(3 downto 0);
signal BIG_BUS : bit_vector (15 downto 0);
begin
  Z_BUS(3) <= '1';
  Z_BUS <= "1100";
  Z_BUS <= b"1100";
  Z_BUS <= x"c";
  Z_BUS <= B"0000000100100011";
End EXAMPLE;
```

24



# Concatenation

---

architecture example1 of concatenation is

```
signal BYTE : bit_vector(7 downto 0);
```

```
signal A_BUS,B_BUS : bit_vector(3 downto 0);
```

```
begin
```

```
    BYTE <= A_BUS & B_BUS;
```

```
End example1;
```

BYTE(7)	←	A_BUS(3)
BYTE(6)	←	A_BUS(2)
BYTE(5)	←	A_BUS(1)
BYTE(4)	←	A_BUS(0)
BYTE(3)	←	B_BUS(3)
BYTE(2)	←	B_BUS(2)
BYTE(1)	←	B_BUS(1)
BYTE(0)	←	B_BUS(0)

25



# Concatenation

---

architecture example2 of concatenation is

```
signal Z_BUS : bit_vector(3 downto 0);
```

```
signal A_BIT,B_BIT,C_BIT,D_BIT : bit;
```

```
begin
```

```
    Z_BUS <= A_BIT & B_BIT & C_BIT & D_BIT;
```

```
End example2;
```

Z_BUS(3)	←	A_BIT
Z_BUS(2)	←	B_BIT
Z_BUS(1)	←	C_BIT
Z_BUS(0)	←	D_BIT

26



## Data Type Classification

---

- Enumeration Types – Example

architecture EXAMPLE of ENUMERATION is

```
type T_STATE is (RESET,START,EXECUTE,FINISH);
```

```
signal CURRENT_STATE,NEXT_STATE : T_STATE;
```

```
signal TWO_BIT_VEC : bit_vector(1 downto 0);
```

```
begin
```

```
-- statement
```

```
end
```

27



## Data Type Classification

---

- Multi-Value Types
  - Un-initialized
  - unknown
  - High impedance
- Types เหล่านี้ ไม่มีรวมอยู่ในภาษา VHDL ก่อนปี 1992
- IEEE ได้ทำการกำหนดค่าเหล่านี้เพิ่มเติมขึ้น โดยรวมอยู่ใน Standard IEEE 1164 (STU\_LOGIC\_1164)

28



## Data Type Classification

- type STD\_LOGIC is (
  - 'U', -- uninitialized
  - 'X', -- strong 0 or 1 (= unknown)
  - '0', -- strong 0
  - '1', -- strong 1
  - 'Z', -- high impedance
  - 'W', -- weak 0 or 1 (= unknown)
  - 'L', -- weak 0
  - 'H', -- weak 1
  - '-', -- don't care)

29



## Example

```
Library ieee;
use ieee.Std_Logic_1164.All;
Entity Comparator is
    Port ( a : in Std_Logic; b in Std_Logic;
          aequab : out Std_Logic);
end Comparator;
Architecture RTL of Comparator is
begin
    process
        if a = b then
            aequab <= '1';
        else
            aequab <='0';
        end if;
    begin
    end process;
end RTL
```

30



## Example

```
Library ieee;
use ieee.Std_Logic_1164.All;
Entity Comparator is
  Port (
    i : in integer range 0 to 9;
    o : out std_logic_vector(6 downto 0);
  );
end Comparator;
```

Architecture RTL of BCD\_7 is

```
begin
  process (i)
  begin
    case i is
      when 0 => o <= "1000000";
      when 1 => o <= "1111001";
      when 2 => o <= "0100100";
      when 3 => o <= "0110000";
      when 4 => o <= "0011001";
      when 5 => o <= "0010010";
      when 6 => o <= "0000011";
      when 7 => o <= "1111000";
      when 8 => o <= "0000000";
      when 9 => o <= "0011000";
    end case;
  end process;
end RTL;
```

31



## คำสั่ง Sequential

- ซึ่งคำสั่งที่เป็นรูปแบบ Sequential ที่มีให้ใช้ในภาษา VHDL มีดังนี้
  - IF Statements
    - IF-END
    - IF-ELSE
    - IF-ELSE-ELSE
  - CASE Statement
  - LOOP Statement
    - FOR LOOP
    - WHILE LOOP
  - WAIT Statement
    - WAIT FOR
    - WAIT ON
    - WAIT UNIT
    - WAIT

32



```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
```

```
ENTITY BinToBCD IS
```

```
PORT (
```

```
data : IN std_logic_vector(7 downto 0);
```

```
A,B,C : OUT std_logic_vector(3 downto 0)
```

```
);
```

```
END BinToBCD;
```

```
ARCHITECTURE arc_BinToBCD OF BinToBCD IS
```

```
BEGIN
```

```
PROCESS (data)
```

```
VARIABLE Num,i,x,y,z : INTEGER;
```

```
BEGIN
```

```
Num := 0;
```

```
FOR i IN 0 TO (data'LENGTH - 1) LOOP
```

```
IF data(i) = '1' THEN
```

```
num := num + 2**i;
```

```
END IF;
```

```
END LOOP;
```

```
x := num / 100;
```

```
num := num mod 100;
```

```
y := num / 10;
```

```
z := num mod 10;
```

33

```
FOR i IN 0 TO (c'LENGTH - 1) LOOP
```

```
IF (x MOD 2 = 1) THEN
```

```
c(i) <= '1';
```

```
ELSE c(i) <= '0';
```

```
END IF;
```

```
x := x/2;
```

```
END LOOP;
```

```
FOR i IN 0 TO (b'LENGTH - 1) LOOP
```

```
IF (y MOD 2 = 1) THEN
```

```
b(i) <= '1';
```

```
ELSE b(i) <= '0';
```

```
END IF;
```

```
y := y/2;
```

```
END LOOP;
```

```
FOR i IN 0 TO (a'LENGTH - 1) LOOP
```

```
IF (z MOD 2 = 1) THEN
```

```
a(i) <= '1';
```

```
ELSE a(i) <= '0';
```

```
END IF;
```

```
z := z/2;
```

```
END LOOP;
```

```
END PROCESS;
```

```
END arc_BinToBCD;
```

34