

**การทดลองที่ 2:** การใช้งานเอ้าท์พุทชนิดตัวเลข 7 ส่วน

**วัตถุประสงค์**

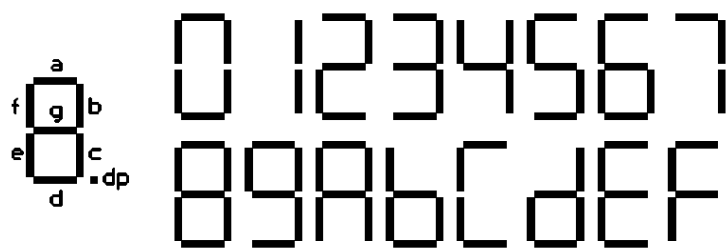
1. เพื่อให้นักศึกษาฝึกเขียนภาษา VHDL เบื้องต้น
2. เพื่อศึกษาตัวอย่างการเขียนโปรแกรมภาษา VHDL เพื่อใช้งานเอ้าท์พุทตัวเลข 7 ส่วน
3. เพื่อให้นักศึกษาสามารถนำโปรแกรมตัวอย่างลงทดสอบบนบอร์ดทดสอบได้

**หมายเหตุ**

1. นักศึกษาที่มีหนังสือ ให้พิมพ์เฉพาะหน้าแรกของเอกสารการทดลองนี้ก็พอ
2. ทำการทดลองตามหนังสือหัวข้อการทดลองที่ 3.3 หน้าที่ 220 ถึง 229 ก่อน

**การทดลอง**

1. ทำการทดลองตามรายละเอียดด้านล่างนี้ โดยตัวอย่างจะเป็นโปรแกรมภาษา VHDL ของการบวกเลข 3 บิต และแสดงผลบนตัวแสดงผลชนิดตัวเลข 7 ส่วน
2. ทำการแก้ไขโปรแกรมตัวอย่างให้เป็นการบวกเลขไบนารีขนาด 4 บิต และแสดงผลที่ตัวแสดงผลเจ็ดส่วนจำนวน 2 หลัก โดยกำหนดให้
  - 2.1 เลื่อนสวิตช์ขึ้นให้อินพุต เป็น 1 เลื่อนลงเป็น 0
  - 2.2 แสดงผลเป็นตัวเลข 0-9 และ a – F ตามรูป



3. ทำการโปรแกรมตัวอย่างลงบอร์ดทดลอง แล้วเรียกอาจารย์ผู้คุมการทดลองมาตรวจ

## รายละเอียดการทดลอง

### 3.3 วงจรถอดรหัส

วงจรถอดรหัสเป็นวงจรคอมบินชันเช่นกัน ตัวอย่างวงจรถอดรหัส เช่น วงจร 2 to 4 Decoder และวงจรถอดรหัส BCD เป็นเซเวนเซกเมนต์ (BCD to 7 Segment decoder) เป็นต้น

วงจรถอดรหัสเข้า 2 ออก 4 (2 to 4 Decoder) นั้นได้อธิบายไปบ้างแล้วในตัวอย่างที่ 2.1 ตัวอย่างที่ 2.10 และตัวอย่างที่ 2.21 โดยในรูปที่ 3.15 และรูปที่ 3.16 เป็นตารางความจริงและโค้ดของวงจร 2 to 4 Decoder ที่ให้เอาต์พุตเป็นลอจิก '1' (One hot decoder) ส่วนในรูปที่ 3.17 และรูปที่ 3.18 จะให้เอาต์พุตเป็นลอจิก '0' (One cold decoder) โดยที่ “-” คือ Don't care โค้ดในรูปที่ 3.16 และรูปที่ 3.18 จะเป็นการเขียนโดยใช้คำสั่ง Conditional signal assignment ซึ่งในทางปฏิบัติเราสามารถเขียนโค้ดโดยใช้คำสั่ง selected signal assignment คำสั่ง if และคำสั่ง case ได้เช่นกัน

Input A			Output Y			
E	A(1)	A(0)	Y(3)	Y(2)	Y(1)	Y(0)
0	-	-	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

รูปที่ 3.15 ตารางความจริงของวงจร 2 to 4 Decoder ที่ให้เอาต์พุตเป็นลอจิก '1' (One hot)

```

2 library IEEE;
3 use IEEE.STD_LOGIC_1164.ALL;
4
5 entity ONE_HOT2to4 is
6     Port ( E : in  STD_LOGIC;
7           A : in  STD_LOGIC_VECTOR (1 downto 0);
8           Y : out STD_LOGIC_VECTOR (3 downto 0));
9 end ONE_HOT2to4;
10
11 architecture Behavioral of ONE_HOT2to4 is
12     signal X : STD_LOGIC_VECTOR (2 downto 0);
13 begin
14     X <= E&A;
15     Y <= "0001" when X="100" else
16         "0010" when X="101" else
17         "0100" when X="110" else
18         "1000" when X="111" else
19         "0000";
20 end Behavioral;

```

รูปที่ 3.16 โค้ดของวงจร 2 to 4 Decoder ที่ให้เอาต์พุตเป็นลอจิก '1' (One hot)

Input A			Output Y			
E	A(1)	A(0)	Y(3)	Y(2)	Y(1)	Y(0)
0	-	-	1	1	1	1
1	0	0	1	1	1	0
1	0	1	1	1	0	1
1	1	0	1	0	1	1
1	1	1	0	1	1	1

รูปที่ 3.17 ตารางความจริงของวงจร 2 to 4 Decoder ที่ให้เอาต์พุตเป็นลอจิก '0' (One cold)

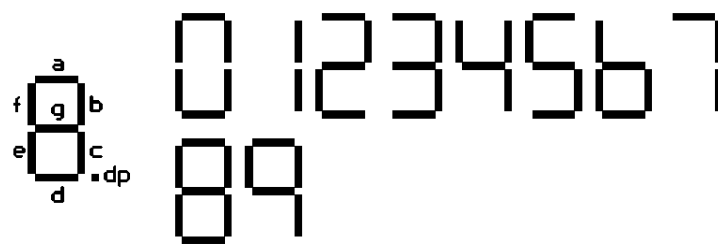
```

2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4
5  entity ONE_COLD2to4 is
6      Port ( E : in  STD_LOGIC;
7            A : in  STD_LOGIC_VECTOR (1 downto 0);
8            Y : out STD_LOGIC_VECTOR (3 downto 0));
9  end ONE_COLD2to4;
10
11 architecture Behavioral of ONE_COLD2to4 is
12     signal X : STD_LOGIC_VECTOR (2 downto 0);
13 begin
14     X <= E&A;
15     Y <= "1110" when X="100" else
16         "1101" when X="101" else
17         "1011" when X="110" else
18         "0111" when X="111" else
19         "1111";
20 end Behavioral;

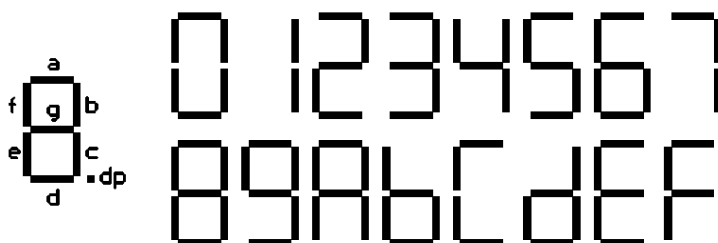
```

รูปที่ 3.18 โค้ดของวงจร 2 to 4 Decoder ที่ให้เอาต์พุตเป็นลอจิก '0' (One cold)

วงจรถอดรหัสอีกประเภทหนึ่งที่นิยมใช้กันมาก คือ วงจรถอดรหัสของตัวแสดงผลเซเวนเซกเมนต์ (7-Segment LED display) ได้แก่ วงจรถอดรหัสตัวแสดงผลเซเวนเซกเมนต์ (7-Segment decoder) ในรูปแบบเลขฐานสิบ (Decimal digit display format) และเลขฐานสิบหก (Hexadecimal digit display format) แสดงดังรูปที่ 3.19a) และรูปที่ 3.19b) ตามลำดับ ซึ่งตัวแสดงผลเซเวนเซกเมนต์ประกอบด้วย LED 7 ส่วน (Segment) คือ a, b, c, d, e, f, g และจุด (dp = Decimal point) หลักในการจัดตำแหน่ง Segment บนแสดงผลนั้นให้เริ่มนับจาก Segment a อยู่ที่ตำแหน่ง 12 นาฬิกา จากนั้นนับวนตามเข็มนาฬิกาก็จะเป็น Segment b, c, d, e และ f ตามลำดับและมี Segment g อยู่ตรงกลาง ความแตกต่างระหว่างรูปแบบเลขฐานสิบและฐานสิบหกจะอยู่ที่เลข 6 และเลข 9 ที่มีเซกเมนต์ (Segment) a และ d เพิ่มเข้ามาเพื่อทำให้เลข 6 ไม่ไปซ้ำกับตัว b และรูปแบบเลขฐานสิบหกจะมี A, b, C, d, E, F และ g เพิ่มเข้ามา ตัวแสดงผลมี 2 แบบ คือ แอนโอดร่วม (Common Anode) และแคโทดร่วม (Common cathode) โดยที่แบบแคโทดร่วมนั้น LED แต่ละส่วนจะติดสว่างได้ก็ต่อเมื่อป้อนลอจิก '1' (โดยต่อผ่านตัวต้านทานเพื่อจำกัดกระแส) และป้อนลอจิก '0' หรือต่อลงกราวด์ที่ขาแอนโอดร่วม แต่ถ้าตัวแสดงผลแบบแอนโอดร่วมก็จะทำงานตรงกันข้าม



3.19a) รูปแบบเลขฐานสิบ



3.19b) รูปแบบเลขฐานสิบหก

รูปที่ 3.19 รูปแบบการแสดงผลบนตัวแสดงผลเซเวนเซกเมนต์

ตารางความจริงของวงจรถอดรหัสตัวแสดงผลเซเวนเซกเมนต์แบบเลขฐานสิบและเลขฐานสิบหกแบบแคโอดร่วมนั้นแสดงดังรูปที่ 3.20 และรูปที่ 3.21 ตามลำดับ

No.	Input A				Output Y						
	A(3)	A(2)	A(1)	A(0)	Y(6)=g	Y(5)=f	Y(4)=e	Y(3)=d	Y(2)=c	Y(1)=b	Y(0)=a
0	0	0	0	0	0	1	1	1	1	1	1
1	0	0	0	1	0	0	0	0	1	1	0
2	0	0	1	0	1	0	1	1	0	1	1
3	0	0	1	1	1	0	0	1	1	1	1
4	0	1	0	0	1	1	0	0	1	1	0
5	0	1	0	1	1	1	0	1	1	0	1
6	0	1	1	0	1	1	1	1	1	0	0
7	0	1	1	1	0	0	0	0	1	1	1
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	0	0	1	1	1

รูปที่ 3.20 ตารางความจริงของตัวถอดรหัสเป็น Decimal digit display format แบบแคโอดร่วม

No.	Input A				Output Y						
	A(3)	A(2)	A(1)	A(0)	Y(6)=g	Y(6)=f	Y(6)=e	Y(6)=d	Y(2)=c	Y(1)=b	Y(0)=a
0	0	0	0	0	0	1	1	1	1	1	1
1	0	0	0	1	0	0	0	0	1	1	0
2	0	0	1	0	1	0	1	1	0	1	1
3	0	0	1	1	1	0	0	1	1	1	1
4	0	1	0	0	1	1	0	0	1	1	0
5	0	1	0	1	1	1	0	1	1	0	1
6	0	1	1	0	1	1	1	1	1	0	1
7	0	1	1	1	0	0	0	0	1	1	1
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	0	1	1	1	1
10 = A	1	0	1	0	1	1	1	0	1	1	1
11 = b	1	0	1	1	1	1	1	1	1	0	0
12 = C	1	1	0	0	0	1	1	1	0	0	1
13 = d	1	1	0	1	1	0	1	1	1	1	0
14 = E	1	1	1	0	1	1	1	1	0	0	1
15 = F	1	1	1	1	1	1	1	0	0	0	1

รูปที่ 3.21 ตารางความจริงของตัวถอดรหัสเป็น Hexadecimal digit display format แบบแคโอดร่วม

เพื่อให้ตัวเลขทุกตัวที่แสดงผลมีรูปแบบเดียวกัน ดังนั้นวงจรถอดรหัสตัวแสดงผลเซเวนเซกเมนต์ที่ใช้ในหนังสือเล่มนี้จะใช้รูปแบบเลขฐานสิบหกดังมีรายละเอียดรูปที่ 3.19b) ทั้งหมดแม้ว่าบางวงจรจะใช้ถอดรหัสเลข 0-9 ก็ตาม การออกแบบวงจรถอดรหัส BCD เป็นเซเวนเซกเมนต์ (BCD to 7-Segment Decoder) นั้นสามารถทำได้โดยนำตารางความจริงรูปที่ 3.21 เฉพาะเลข 0-9 ไปเขียนโค้ด

โค้ดของวงจรถอดรหัสตัวแสดงผลเซเวนเซกเมนต์ (7-Segment decoder) แบบแคโอดร่วมมีรายละเอียดในบทที่ 2 ในตัวอย่างที่ 2.3 ตัวอย่างที่ 2.11 ตัวอย่างที่ 2.12 ตัวอย่างที่ 2.22 และตัวอย่างที่ 2.23 ตามลำดับ

## การทดลองที่ 3.3.1 วงจร 2 to 4 Decoder

### วัตถุประสงค์

- 1) เพื่อทำความเข้าใจเกี่ยวกับหลักการทำงานของวงจรถอดรหัสแบบ 2 to 4 Decoder
- 2) เพื่อทดลองใช้ ISE WebPACK 8.1i สร้างวงจรถอดรหัสและโปรแกรมลงชิพ CPLD หรือ FPGA

### อุปกรณ์ทดลอง

บอร์ด CPLD Explorer XC9572XL หรือ CPLD Explorer XC9572 และ FPGA Discovery-III XC3S200F หรือ FPGA Discovery-III XC3S200F4

### 1 สร้างวงจร 2 to 4 Decoder ด้วย CPLD

ให้ออกแบบวงจร 2 to 4 Decoder แบบ One hot ด้วย CPLD ดังตารางความจริงในรูปที่ 3.15 โดยเขียนโค้ดดังรูปที่ L1.1 โดยใช้คำสั่ง if การทดลองนี้จะใช้ Project Location ชื่อ ch3v แล้วกำหนด Project Name และ Source File ชื่อ ex3\_3\_1vcx1

```

2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4
5  entity ex3_3_1vcx1 is
6      Port ( E : in  STD_LOGIC;
7            A : in  STD_LOGIC_VECTOR (1 downto 0);
8            D : out STD_LOGIC_VECTOR (3 downto 0));
9  end ex3_3_1vcx1;
10
11 architecture Behavioral of ex3_3_1vcx1 is
12     begin
13     process (A,E)
14         begin
15             if E='0' then D <= "0000";
16             else if A="00" then D <= "0001";
17                 elsif A="01" then D <= "0010";
18                 elsif A="10" then D <= "0100";
19                 elsif A="11" then D <= "1000";
20                 else D <= "0000";
21             end if;
22         end if;
23     end process;
24 end Behavioral;

```

รูปที่ L1.1 โค้ดของวงจร 2 to 4 Decoder แบบ One hot โดยใช้คำสั่ง if

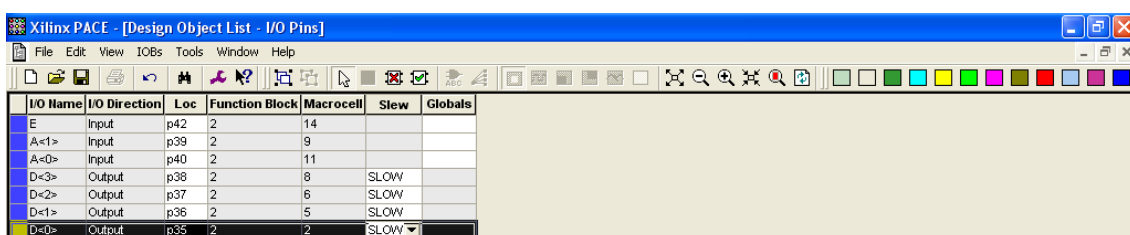
จากนั้นกำหนดขาสัญญาณต่างๆ จะใช้ปุ่มกด PB1–PB3 เป็นอินพุตและ LED1-LED4 เป็นเอาต์พุต กล่าวคือ

A(1) = PB1 = INPUT = p39      D(0) = LED1 = OUTPUT= p35    D(2) = LED3 = OUTPUT= p37

A(0) = PB2 = INPUT = p40      D(1) = LED2 = OUTPUT= p36    D(3) = LED4 = OUTPUT= p38

E = PB3 (Slide SW1)= INPUT= p42

โดยพิมพ์ใน Assign Package Pins สรุปดังนี้



I/O Name	I/O Direction	Loc	Function Block	Macrocell	Stew	Globals
E	Input	p42	2	14		
A<1>	Input	p39	2	9		
A<0>	Input	p40	2	11		
D<3>	Output	p38	2	8	SLOW	
D<2>	Output	p37	2	6	SLOW	
D<1>	Output	p36	2	5	SLOW	
D<0>	Output	p35	2	2	SLOW	

รูปที่ L1.2 Assign Package Pins

หลังจากโปรแกรมวงจรที่ออกแบบลงซีพ CPLD แล้วให้ OFF Slide SW1 แล้วทดลองกดปุ่ม PB1–PB2 และให้สังเกตที่ LED1–LED4 ว่าให้ลอจิกเอาต์พุตเป็นไปตามทฤษฎีหรือไม่ จากนั้น ON Slide SW1 แล้วทดลองซ้ำ แล้วบันทึกผลการทดลอง

เมื่อทดลองเสร็จแล้วให้แก้ไขโค้ดเป็นวงจร 2 to 4 Decoder แบบ One cold แล้วทำการทดลองซ้ำ

## 2 สร้างวงจร 2 to 4 Decoder ด้วย FPGA

ให้ออกแบบวงจร 2 to 4 Decoder แบบ One hot ด้วย FPGA ดังตารางความจริงในรูปที่ 3.15 โดยเขียนโค้ดดังรูปที่ L2.1 โดยใช้คำสั่ง if การทดลองนี้จะใช้ Project Location ชื่อ ch3v แล้วกำหนด Project Name และ Source File ชื่อ ex3\_3\_1vf

```

2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4
5  entity ex3_3_1vf is
6      Port ( E : in  STD_LOGIC;
7            A : in  STD_LOGIC_VECTOR (1 downto 0);
8            D : out STD_LOGIC_VECTOR (3 downto 0));
9  end ex3_3_1vf;
10
11 architecture Behavioral of ex3_3_1vf is
12     begin
13     process (A,E)
14     begin
15         if E='0' then D <= "0000";
16         else if A="00" then D <= "0001";
17             elsif A="01" then D <= "0010";
18             elsif A="10" then D <= "0100";
19             elsif A="11" then D <= "1000";
20             else D <= "0000";
21         end if;
22     end if;
23 end process;
24 end Behavioral;

```

รูปที่ L2.1 โค้ดของวงจร 2 to 4 Decoder แบบ One hot โดยใช้คำสั่ง if

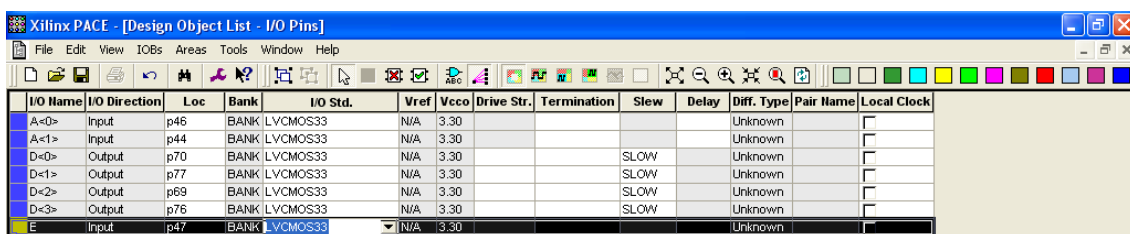
จากนั้นกำหนดขาสัญญาณต่างๆ จะใช้ปุ่มกด PB1–PB3 เป็นอินพุตและ LED L0–L3 เป็นเอาต์พุต กล่าวคือ

A(1) = PB1 = INPUT= p44      D(0) = L0 = OUTPUT = p70      D(2) = L2 = OUTPUT = p69

A(0) = PB2 = INPUT= p46      D(1) = L1 = OUTPUT = p77      D(3) = L3 = OUTPUT = p76

E = PB3 = INPUT= p47

โดยพิมพ์ใน Assign Package Pins ดังรูปนี้



I/O Name	I/O Direction	Loc	Bank	I/O Std.	Vref	Vcco	Drive Str.	Termination	Slew	Delay	Diff. Type	Pair Name	Local Clock
A<0>	Input	p46	BANK	LVC MOS33	N/A	3.30					Unknown		
A<1>	Input	p44	BANK	LVC MOS33	N/A	3.30					Unknown		
D<0>	Output	p70	BANK	LVC MOS33	N/A	3.30			SLOW		Unknown		
D<1>	Output	p77	BANK	LVC MOS33	N/A	3.30			SLOW		Unknown		
D<2>	Output	p69	BANK	LVC MOS33	N/A	3.30			SLOW		Unknown		
D<3>	Output	p76	BANK	LVC MOS33	N/A	3.30			SLOW		Unknown		
E	Input	p47	BANK	LVC MOS33	N/A	3.30					Unknown		

รูปที่ L2.2 Assign Package Pins

หลังจากโปรแกรมวงจรที่ออกแบบลงซีพ FPGA แล้วให้ทดลองกดปุ่ม PB1–PB2 และให้สังเกตที่ LED L0–L3 ว่าให้ลอจิกเอาต์พุตเป็นไปตามทฤษฎีหรือไม่ จากนั้นให้กดปุ่ม PB3 ค้างไว้แล้วทำการทดลองซ้ำ แล้วบันทึกผลการทดลอง

เมื่อทดลองเสร็จแล้วให้แก้ไขโค้ดเป็นวงจร 2 to 4 Decoder แบบ One cold แล้วทำการทดลองซ้ำ

**แบบฝึกหัด** ให้เขียนโค้ด 2 to 4 Decoder โดยใช้คำสั่ง selected signal assignment, Conditional signal assignment และ case

## การทดลองที่ 3.3.2 วงจรถอดรหัสตัวแสดงผลเซเวนเซกเมนต์

### วัตถุประสงค์

- 1) เพื่อทำความเข้าใจเกี่ยวกับวงจรถอดรหัสตัวแสดงผลเซเวนเซกเมนต์
- 2) เพื่อทดลองใช้ ISE WebPACK 8.1i สร้างวงจรถอดรหัสและโปรแกรมลงชิพ CPLD และ FPGA

### อุปกรณ์ทดลอง

บอร์ด CPLD Explorer XC9572XL หรือ CPLD Explorer XC9572 และ FPGA Discovery-III XC3S200F หรือ FPGA Discovery-III XC3S200F4

### 1 สร้างวงจรถอดรหัสตัวแสดงผลเซเวนเซกเมนต์ด้วย CPLD

โค้ดวงจรถอดรหัสตัวแสดงผลเซเวนเซกเมนต์แบบเลขฐานสิบหกแบบแคโอดร่วมและวงจร Common cathode แสดงดังรูปที่ L1.1 โดยจะเขียนไว้ใน Project Location ชื่อ ch3v แล้วกำหนด Project Name และ Source File ชื่อ ex3\_3\_2vcx1

```

2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4
5  entity ex3_3_2vcx1 is
6      Port ( A : in  STD_LOGIC_VECTOR (3 downto 0);
7            COMMON : out  STD_LOGIC;
8            Y : out  STD_LOGIC_VECTOR (6 downto 0) );
9  end ex3_3_2vcx1;
10
11 architecture Behavioral of ex3_3_2vcx1 is
12 begin
13     -----HEX to 7-segments decoder-----
14     --gfedcba
15     Y <= "1110001" when A="1111" else --F
16         "1111001" when A="1110" else --E      Y(0)=a
17         "1011110" when A="1101" else --d      ---
18         "0111001" when A="1100" else --C Y(5)=f|   | Y(1)=b
19         "1111100" when A="1011" else --b      --- Y(6)=g
20         "1110111" when A="1010" else --A Y(4)=e|   | Y(2)=c
21         "1101111" when A="1001" else --9      ---
22         "1111111" when A="1000" else --8      Y(3)=d
23         "0000111" when A="0111" else --7
24         "1111101" when A="0110" else --6
25         "1101101" when A="0101" else --5
26         "1100110" when A="0100" else --4
27         "1001111" when A="0011" else --3
28         "1011011" when A="0010" else --2
29         "0000110" when A="0001" else --1
30         "0111111"; --0
31     -----Common cathode-----
32     COMMON <= '0';
33 end Behavioral;

```

รูปที่ L1.1 วงจรทดสอบวงจรถอดรหัสตัวแสดงผลเซเวนเซกเมนต์แบบเลขฐานสิบหกแบบแคโอดร่วม

การกำหนดขาสัญญาณต่างๆ จะใช้ปุ่มกด PB3-PB6 (Slide SW1- Slide SW4) เป็นอินพุต มีตัวแสดงผลเซเวนเซกเมนต์แบบแคโอดร่วมหลักที่ 1 คือ DIGIT1 เป็นเอาต์พุต กล่าวคือ

A(3) = PB3 (Slide SW1) = INPUT = p42    Y(0) = a = OUTPUT = p27    Y(5) = f = OUTPUT = p20

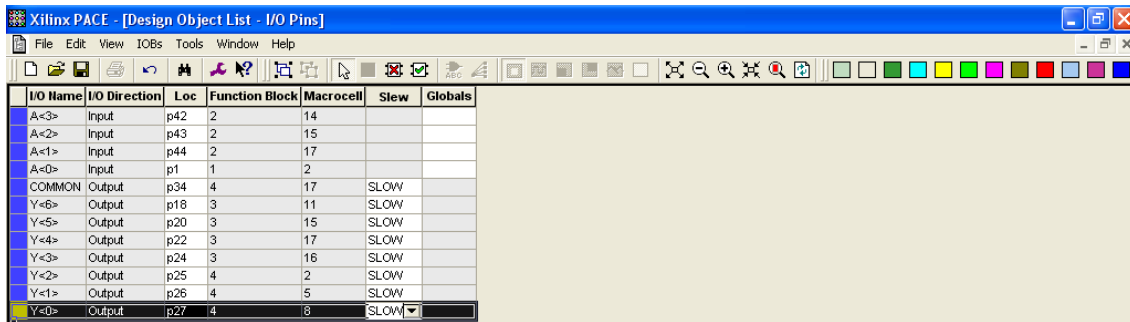
A(2) = PB4(Slide SW2) = INPUT = p43    Y(1) = b = OUTPUT = p26    Y(6) = g = OUTPUT = p18

A(1) = PB5(Slide SW3) = INPUT = p44    Y(2) = c = OUTPUT = p25

A(0) = PB6(Slide SW4) = INPUT = p1    Y(3) = d = OUTPUT = p24

COMMON = DIGIT1 = OUTPUT = p34 Y(4) = e = OUTPUT = p22

โดยพิมพ์ใน Assign Package Pins สรุปดังนี้



I/O Name	I/O Direction	Loc	Function Block	Macrocell	Slew	Globals
A<3>	Input	p42	2	14		
A<2>	Input	p43	2	15		
A<1>	Input	p44	2	17		
A<0>	Input	p1	1	2		
COMMON	Output	p34	4	17	SLOW	
Y<6>	Output	p18	3	11	SLOW	
Y<5>	Output	p20	3	15	SLOW	
Y<4>	Output	p22	3	17	SLOW	
Y<3>	Output	p24	3	16	SLOW	
Y<2>	Output	p25	4	2	SLOW	
Y<1>	Output	p26	4	5	SLOW	
Y<0>	Output	p27	4	8	SLOW	

รูปที่ L1.2 Assign Package Pins

หลังจากโปรแกรมวงจรลงชิพ CPLD แล้วให้ทดลอง ON-OFF Slide SW1-Slide SW4 โดยเริ่มจาก ON Slide SW1-Slide SW4 ทุกตัว (คือ A = "0000") ที่ละจำนวนกระทั่งถึง OFF Slide SW1-Slide SW4 ทุกตัว (คือ A = "1111") พร้อมกับให้สังเกตที่ตัวแสดงผลเซเวนเซกเมนต์ว่ามีส่วนใดติดสว่าง ให้ลองจิกเอาต์พุตเป็นไปตามทฤษฎีหรือไม่ แล้วบันทึกผลการทดลอง

จากนั้นทำการแก้ไขโค้ดเป็นวงจรถอดรหัสแบบเลขฐานสิบแบบแคโอดร่วม แล้วทำการทดลองซ้ำ

## 2 สร้างวงจรถอดรหัสตัวแสดงผลเซเวนเซกเมนต์ด้วย FPGA

โค้ดวงจรถอดรหัสตัวแสดงผลเซเวนเซกเมนต์แบบเลขฐานสิบหกแบบแคโอดร่วมและวงจร Common cathode แสดงดังรูปที่ L2.1 โดยจะเขียนไว้ใน Project Location ชื่อ ch3v แล้วกำหนด Project Name และ Source File ชื่อ ex3\_3\_2vf

```

2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4
5  entity ex3_3_2vf is
6      Port ( A : in  STD_LOGIC_VECTOR (3 downto 0);
7            COMMON : out  STD_LOGIC;
8            Y : out  STD_LOGIC_VECTOR (6 downto 0) );
9  end ex3_3_2vf;
10
11  architecture Behavioral of ex3_3_2vf is
12  begin
13  -----HEX to 7-segments decoder-----
14      --gfedcba
15      Y <= "1110001" when A="1111" else --F
16           "1111001" when A="1110" else --E      Y(0)=a
17           "1011110" when A="1101" else --d      ---
18           "0111001" when A="1100" else --C Y(5)=f | Y(1)=b
19           "1111100" when A="1011" else --b      --- Y(6)=g
20           "1110111" when A="1010" else --A Y(4)=e | Y(2)=c
21           "1101111" when A="1001" else --9      ---
22           "1111111" when A="1000" else --8      Y(3)=d
23           "0000111" when A="0111" else --7
24           "1111101" when A="0110" else --6
25           "1101101" when A="0101" else --5
26           "1100110" when A="0100" else --4
27           "1001111" when A="0011" else --3
28           "1011011" when A="0010" else --2
29           "0000110" when A="0001" else --1
30           "0111111"; --0
31  -----Common cathode-----
32      COMMON <= '0';
33  end Behavioral;

```

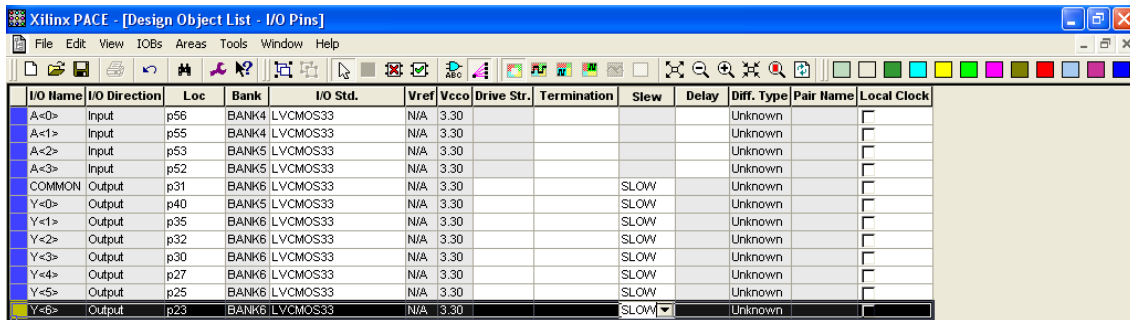
รูปที่ L2.1 วงจรทดสอบวงจรถอดรหัสตัวแสดงผลเซเวนเซกเมนต์แบบเลขฐานสิบหกแบบแคโอดร่วม

การกำหนดขาสัญญาณต่างๆ จะใช้ Dip SW1-Dip SW4 เป็นอินพุต มีตัวแสดงผลเซเวนเซกเมนต์แบบแคโอดร่วมหลักที่ 1 คือ DIGIT1 เป็นเอาต์พุต กล่าวคือ



A(3)= Dip SW1 = INPUT = p52      Y(0) = a = OUTPUT = p40      Y(5) = f = OUTPUT = p25  
 A(2)= Dip SW2 = INPUT = p53      Y(1) = b = OUTPUT = p35      Y(6) = g = OUTPUT = p23  
 A(1)= Dip SW3 = INPUT = p55      Y(2) = c = OUTPUT = p32  
 A(0)= Dip SW4 = INPUT = p56      Y(3) = d = OUTPUT = p30  
 COMMON = DIGIT1 = OUTPUT = p31      Y(4) = e = OUTPUT = p27

โดยพิมพ์ใน Assign Package Pins สรุปดังนี้



I/O Name	I/O Direction	Loc	Bank	I/O Std.	Vref	Vcco	Drive Str.	Termination	Slew	Delay	Diff. Type	Pair Name	Local Clock
A<0>	Input	p56	BANK4	LVC MOS33	N/A	3.30					Unknown		
A<1>	Input	p55	BANK4	LVC MOS33	N/A	3.30					Unknown		
A<2>	Input	p53	BANK5	LVC MOS33	N/A	3.30					Unknown		
A<3>	Input	p52	BANK5	LVC MOS33	N/A	3.30					Unknown		
COMMON	Output	p31	BANK6	LVC MOS33	N/A	3.30			SLOW		Unknown		
Y<0>	Output	p40	BANK6	LVC MOS33	N/A	3.30			SLOW		Unknown		
Y<1>	Output	p35	BANK6	LVC MOS33	N/A	3.30			SLOW		Unknown		
Y<2>	Output	p32	BANK6	LVC MOS33	N/A	3.30			SLOW		Unknown		
Y<3>	Output	p30	BANK6	LVC MOS33	N/A	3.30			SLOW		Unknown		
Y<4>	Output	p27	BANK6	LVC MOS33	N/A	3.30			SLOW		Unknown		
Y<5>	Output	p25	BANK6	LVC MOS33	N/A	3.30			SLOW		Unknown		
Y<6>	Output	p23	BANK6	LVC MOS33	N/A	3.30			SLOW		Unknown		

รูปที่ L2.2 Assign Package Pins

หลังจากโปรแกรมวงจรที่ออกแบบลงชิพ FPGA แล้วให้ทดลอง ON-OFF Dip SW1-Dip SW4 โดยเริ่มจาก ON Dip SW1-Dip SW4 ทุกตัว (คือ A = “0000”) ที่ละค่าจนกระทั่งถึง OFF Dip SW1-Dip SW4 ทุกตัว (คือ A = “1111”) พร้อมกับสังเกตที่ตัวแสดงผลเซเวนเซกเมนต์ว่ามีส่วนใดติดสว่าง ให้ลองจิกเอาต์พุตเป็นไปตามทฤษฎีหรือไม่ แล้วบันทึกผลการทดลอง

จากนั้นทำการแก้ไขโค้ดเป็นวงจรถอดรหัสแบบเลขฐานสิบแบบแคโอดร่วม แล้วทำการทดลองซ้ำ

**แบบฝึกหัด** จงสร้างวงจรบวกเลขที่สามารถบวกเลขจำนวนเต็มบวกขนาด 3 บิต โดยแสดงผลทางตัวแสดงผลเซเวนเซกเมนต์

## การทดลองที่ 3.3.3 วงจรบวกเลข 3 บิตที่แสดงผลทางตัวแสดงผลเซเวนเซกเมนต์

### วัตถุประสงค์

- 1) เพื่อทำความเข้าใจเกี่ยวกับการประยุกต์ใช้วงจรลอจิกที่แสดงผลเซเวนเซกเมนต์
- 2) เพื่อทดลองใช้ ISE WebPACK 8.1i สร้างวงจรบวกและโปรแกรมลงชิพ CPLD และ FPGA

### อุปกรณ์ทดลอง

บอร์ด CPLD Explorer XC9572XL หรือ CPLD Explorer XC9572 และ FPGA Discovery-III XC3S200F หรือ FPGA Discovery-III XC3S200F4

### 1 สร้างวงจรบวกเลข 3 บิตที่แสดงผลทางตัวแสดงผลเซเวนเซกเมนต์ด้วย CPLD

สร้างไฟล์โค้ดดังรูป L1.1 ใน Project Location ชื่อch3v แล้วกำหนด Project Name และ Source File ชื่อex3\_3\_3vcx1

```

2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4  use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6  entity ex3_3_3vcx1 is
7      Port ( A,B : in  STD_LOGIC_VECTOR (2 downto 0);
8            COM : out STD_LOGIC;
9            Y : out  STD_LOGIC_VECTOR (6 downto 0));
10 end ex3_3_3vcx1;
11
12 architecture Behavioral of ex3_3_3vcx1 is
13     signal X : STD_LOGIC_VECTOR (3 downto 0);
14 begin
15     -----3 bits adder with 4 bits output-----
16     X <= ('0' & A) + ('0' & B);
17     -----HEX to 7-segments decoder-----
18     Y <= "1110001" when X="1111" else --F
19         "1111001" when X="1110" else --E      Y(0)=a
20         "1011110" when X="1101" else --d      ---
21         "0111001" when X="1100" else --C Y(5)=f | Y(1)=b
22         "1111100" when X="1011" else --b      --- Y(6)=g
23         "1110111" when X="1010" else --A Y(4)=e | Y(2)=c
24         "1101111" when X="1001" else --9      ---
25         "1111111" when X="1000" else --8      Y(3)=d
26         "0000111" when X="0111" else --7
27         "1111101" when X="0110" else --6
28         "1101101" when X="0101" else --5
29         "1100110" when X="0100" else --4
30         "1001111" when X="0011" else --3
31         "1011011" when X="0010" else --2
32         "0000110" when X="0001" else --1
33         "0111111"; --0
34     -----Common cathode-----
35     COM <= '0';
36 end Behavioral;

```

รูปที่ L1.1 วงจรบวกเลข 3 บิตที่แสดงผลทางตัวแสดงผลเซเวนเซกเมนต์

การกำหนดขาสัญญาณต่างๆ จะใช้ปุ่มกด PB1 ถึง PB6 เป็นอินพุต มีตัวแสดงผล DIGIT1 เป็นเอาต์พุต กล่าวคือ

A(2) = PB1 = INPUT= p39	Y(0) = a = OUTPUT = p27	Y(6) = g = OUTPUT = p18
A(1) = PB2 = INPUT= p40	Y(1) = b = OUTPUT = p26	COM= DIGIT1= OUTPUT= p34
A(0) = PB3 (Slide SW1)= INPUT= p42	Y(2) = c = OUTPUT = p25	
B(2) = PB4 (Slide SW2)= INPUT= p43	Y(3) = d = OUTPUT= p24	
B(1) = PB5 (Slide SW3)= INPUT= p44	Y(4) = e = OUTPUT= p22	

B(0) = PB6 (Slide SW4)= INPUT= p1    Y(5) = f = OUTPUT = p20

โดยพิมพ์ใน Assign Package Pins สรุปดังนี้

I/O Name	I/O Direction	Loc	Function Block	Macrocell	Slew	Globals
A<2>	Input	p39	2	9		
A<1>	Input	p40	2	11		
A<0>	Input	p42	2	14		
B<2>	Input	p43	2	15		
B<1>	Input	p44	2	17		
B<0>	Input	p1	1	2		
COM	Output	p34	4	17	SLOW	
Y<6>	Output	p18	3	11	SLOW	
Y<5>	Output	p20	3	15	SLOW	
Y<4>	Output	p22	3	17	SLOW	
Y<3>	Output	p24	3	16	SLOW	
Y<2>	Output	p25	4	2	SLOW	
Y<1>	Output	p26	4	5	SLOW	
Y<0>	Output	p27	4	8	SLOW	

รูปที่ L1.2 Assign Package Pins

หลังจากโปรแกรม CPLD แล้วให้กดปุ่ม PB1-PB6 ร่วมกับ Slide SW1-Slide SW4 เพื่อสร้างเป็นอินพุตค่าต่างๆ แล้วให้สังเกตเอาต์พุตว่าเป็นไปตามทฤษฎีหรือไม่ แล้วจึงบันทึกผลการทดลอง

## 2 สร้างวงจรบวกเลข 3 บิตที่แสดงผลทางตัวแสดงผลเซเวนเซกเมนต์ด้วย FPGA

นำโค้ดในรูป L1.1 มาแก้ไข Entity เป็น ex3\_3\_3vf แล้วนำไปสร้างไว้ใน Project Location ชื่อch3v แล้วกำหนด Project Name และ Source File ชื่อex3\_3\_3vf

การกำหนดขาสัญญาณต่างๆ จะใช้ Dip SW1-Dip SW6 เป็นอินพุต มีตัวแสดงผล DIGIT1 เป็นเอาต์พุต กล่าวคือ

A(2) = Dip SW1 = INPUT= p52    Y(0) = a = OUTPUT = p40    Y(6) = g = OUTPUT = p23

A(1) = Dip SW2 = INPUT= p53    Y(1) = b = OUTPUT = p35    COM= DIGIT1= OUTPUT= p31

A(0) = Dip SW3 = INPUT= p55    Y(2) = c = OUTPUT = p32

B(2) = Dip SW4 = INPUT= p56    Y(3) = d = OUTPUT= p30

B(1) = Dip SW5 = INPUT= p59    Y(4) = e = OUTPUT= p27

B(0) = Dip SW6 = INPUT= p60    Y(5) = f = OUTPUT = p25

โดยพิมพ์ใน Assign Package Pins สรุปดังนี้

I/O Name	I/O Direction	Loc	Bank	I/O Std.	Vref	Vcco	Drive Str.	Termination	Slew	Delay	Diff. Type	Pair Name	Local Clock
A<0>	Input	p55	BANK5	LVC MOS33	N/A	3.30					Unknown		
A<1>	Input	p53	BANK5	LVC MOS33	N/A	3.30					Unknown		
A<2>	Input	p52	BANK5	LVC MOS33	N/A	3.30					Unknown		
B<0>	Input	p60	BANK4	LVC MOS33	N/A	3.30					Unknown		
B<1>	Input	p59	BANK4	LVC MOS33	N/A	3.30					Unknown		
B<2>	Input	p56	BANK4	LVC MOS33	N/A	3.30					Unknown		
COM	Output	p31	BANK6	LVC MOS33	N/A	3.30			SLOW		Unknown		
Y<0>	Output	p40	BANK5	LVC MOS33	N/A	3.30			SLOW		Unknown		
Y<1>	Output	p35	BANK6	LVC MOS33	N/A	3.30			SLOW		Unknown		
Y<2>	Output	p32	BANK6	LVC MOS33	N/A	3.30			SLOW		Unknown		
Y<3>	Output	p30	BANK6	LVC MOS33	N/A	3.30			SLOW		Unknown		
Y<4>	Output	p27	BANK6	LVC MOS33	N/A	3.30			SLOW		Unknown		
Y<5>	Output	p25	BANK6	LVC MOS33	N/A	3.30			SLOW		Unknown		
Y<6>	Output	p23	BANK6	LVC MOS33	N/A	3.30			SLOW		Unknown		

รูปที่ L2.2 Assign Package Pins

หลังจากโปรแกรมวงจรที่ออกแบบลงชิพ FPGA แล้วให้ทดลอง ON-OFF Dip SW1-Dip SW6 เพื่อสร้างเป็นอินพุตค่าต่างๆ แล้วให้สังเกตเอาต์พุตว่าเป็นไปตามทฤษฎีหรือไม่ แล้วจึงบันทึกผลการทดลอง