



ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

ภาคการศึกษาที่ 1 ปีการศึกษา 2564

รหัสวิชา 010113026 ชื่อวิชา Digital Laboratory

ตอนเรียน 7 หมายถึงได้ -

รหัสนักศึกษา 62010.11631188

ชื่อ-นามสกุล นายโสรณ ศุภสมบุรณ์

อาจารย์ผู้สอน CS.P

เวลาที่ทำการทดลอง พก 13.00-16.00 วันที่ 7/10/21

การทดลองที่ 10

4x3 Keypad, State Machines Applications

วัตถุประสงค์

1. เพื่อให้สามารถใช้โปรแกรมคอมพิวเตอร์จำลองการทำงานของวงจรลอจิกเกตได้
2. เพื่อให้สามารถประยุกต์ใช้วงจรและอุปกรณ์ดิจิทัลเพื่อออกแบบระบบงานที่ซับซ้อนได้
3. เพื่อให้สามารถประยุกต์ใช้การออกแบบระบบงานดิจิทัลด้วยเทคนิคทาง state diagram ได้
4. เพื่อให้สามารถประยุกต์ใช้เทคนิคทาง state diagram มาทำการควบคุมอุปกรณ์ Keypad ได้

อุปกรณ์

1. ระบบคอมพิวเตอร์ 1 เครื่อง พร้อมติดตั้งโปรแกรม Quartus II เวอร์ชัน 8.0 (Student Edition) ขึ้นไป
2. บอร์ดทดลอง Cyclone3-Lab01 1 บอร์ด
3. สาย J-TAG 1 เส้น ใช้รุ่น USB-Blaster (สำหรับเครื่อง Notebook) หรือรุ่น Byte-Blaster (สำหรับเครื่อง PC)
4. บอร์ดแสดงผล 7-segment (รุ่นแป้นพิมพ์ keypad สีขาว)

การทดลอง

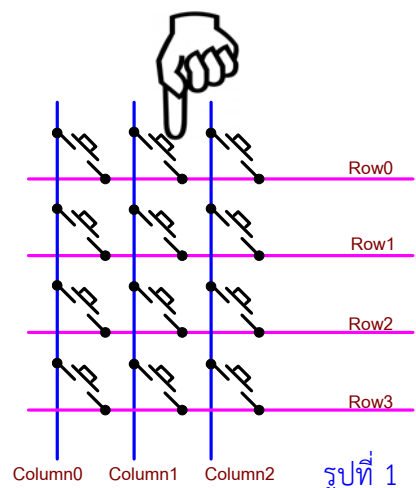
A1: ขั้นตอนเตรียมการ “แป้นคีย์บอร์ดแบบ Matrix Keypad”

แนวการออกแบบระบบงาน “A Keypad Encoder”

ในการทดลองนี้จะเป็นการสร้างระบบงานชื่อ “วงจรเข้ารหัสของแป้นพิมพ์ หรือ Keypad Encoder” ซึ่งจะเป็นตัวอย่างการนำเทคนิคการออกแบบด้วย state diagram มาช่วยในการออกแบบ ดังนั้นขอให้นักศึกษาทำการศึกษานำคิดลำดับขั้นตอนการทำงานของ เครื่อง Keypad Encoder นี้ให้เข้าใจพฤติกรรมของมันก่อนแล้วค่อยเริ่มต้นลงมือทำการออกแบบจะทำให้เข้าใจได้ดีขึ้น

การทำงานของ Keypad

โครงสร้างทางกลของสวิตช์กดแบบคีย์แพด (Keypad) 4x3 ในรูปที่ 1 ประกอบด้วยเส้นลวดตัวนำในแนวนอน (row, 4 เส้น) และเส้นลวดตัวนำในแนวตั้ง (column, 3 เส้น) ลวดทั้ง 2 แนวจะไม่สัมผัสกันโดยตรงที่จุดตัดกัน แต่จะมีสวิตช์มาเชื่อมต่อให้เกิดการสัมผัสหรือลัดวงจรแทน ดังนั้นถ้ากดปุ่มสวิตช์ปุ่มใดปุ่มหนึ่ง เราจะสามารถระบุตำแหน่งของสวิตช์ที่ถูกกดได้ด้วยการวิเคราะห์จากคู่ลำดับ (Row, Column) ของจุดสัมผัสที่มีการลัดวงจร (แนวทางนี้จะไม่รวมกรณีที่มีการกดสวิตช์มากกว่า 1 ปุ่ม)

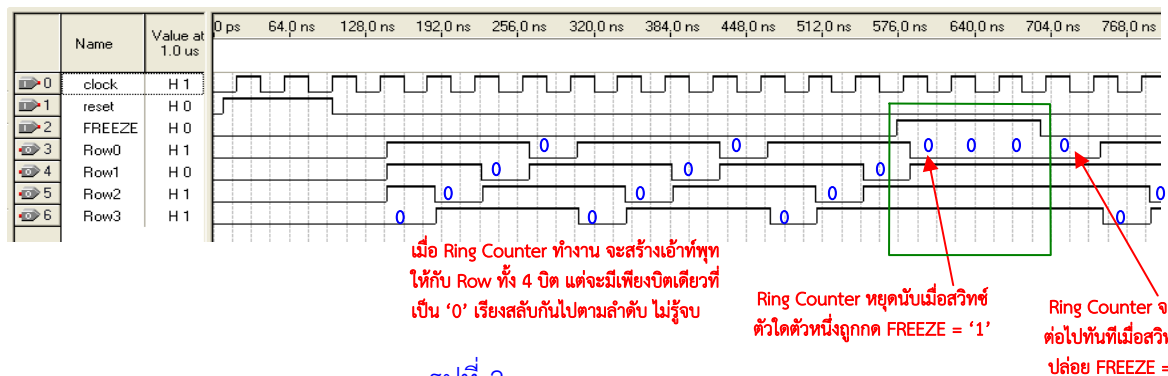
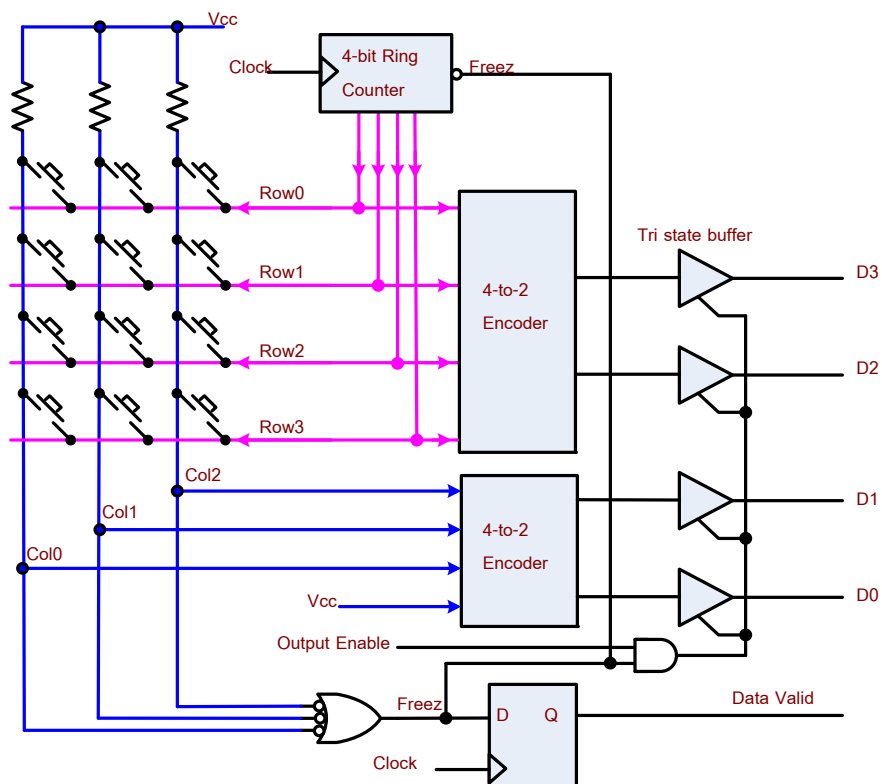




วงจร Encoder สำหรับ Keypad

วงจรตรวจจับตำแหน่งของการกดสวิตช์ในรูปแบบที่ 2 จะประกอบไปด้วยวงจร ring counter ที่จะสร้างสัญญาณขึ้นมา 4 บิตแต่จะมีเพียงบิตเดียวที่เป็น '0' ให้ค่าหมุนวนต่อเนื่องตามลำดับ 1110,1101,1011,0111,1110, ... ป้อนให้แนว Row ซึ่งจะเสมือนเป็นการสแกนด้วยค่า '0' ที่ละ Row ไม่ซ้ำกัน ในเวลาเดียวกันสัญญาณนี้จะถูกส่งไปให้ Row Encoder เพื่อถอดรหัสว่าตอนนี้กำลังสแกนดู row ที่เท่าใด **ถ้ามีการกดสวิตช์จะทำให้ค่าสัญญาณลอจิกของเส้นลวดในแนว column ที่ต่ออยู่กับสวิตช์นั้นเปลี่ยนจากเดิมที่เป็น '1' ไปเป็น '0' (ถ้าไม่มีการกดสวิตช์เลย ลวดทุกเส้นในแนวตั้งจะมีสถานะเป็น '1' ทั้งหมดเนื่องจากปลายอีกด้านของตัวนำต่ออยู่กับไฟ Vcc)** ดังนั้นการที่รู้ว่าเส้น column ใดที่เป็น '0' เพียง 1 เส้นจากใน 3 เส้นรวมกับการรู้ว่าในขณะนั้นกำลังสแกนดู row ใด ก็จะทำให้สามารถทราบตำแหน่งของสวิตช์ที่ถูกกดได้ว่าอยู่ใน column, row ที่เท่าใด

นอกจากนี้วงจร ring counter จะถูกสั่งให้หยุดนับเมื่อมีสวิตช์ถูกกด (สัญญาณ freeze='1') และค้างค่าไว้จนกว่าจะปล่อยจึงนับต่อ เอาท์พุทที่ได้จะเป็น $D_3D_2D_1D_0$ และสัญญาณ DataValid บอกสถานะว่ามีการกดสวิตช์



รูปที่ 2



ขั้นที่ 1 เตรียมการสร้างวงจร(ระบบงาน) 4 bit Ring Counter

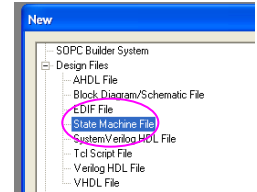
คำสั่งการทดลอง

1. ให้สร้างไฟล์เตอร์สำหรับเก็บงานขึ้นใหม่เพื่อเก็บงานในการทดลองนี้ชื่อ “Lab10_Keypad”
2. สร้างอุปกรณ์ชื่อ “RingCounter” ซึ่งเป็นวงจรของระบบงาน Ring Counter โดยดำเนินการตามขั้นตอนดังนี้
 - 2.1 สร้างโปรเจกต์ชื่อ “RingCounter” และให้ใช้ชิพ EP3C10E144C8
 - 2.2 สร้างไฟล์สำหรับเขียนแผนภาพ state diagram

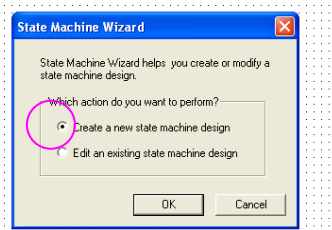
>> File >> New

เลือก State Machine file ดังรูปที่ 3a

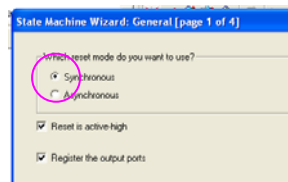
รูปที่ 3a



- 2.3 ให้เลือกใช้เครื่องมือเป็น State Machine Wizard ดังรูปที่ 3b -3d

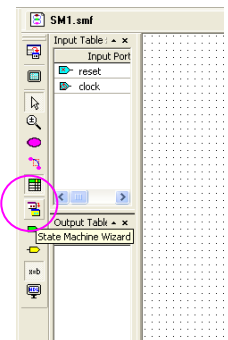


รูปที่ 3c



รูปที่ 3d

รูปที่ 3b



- 2.4 กำหนดคุณสมบัติของ state table ดังรูปที่ 4

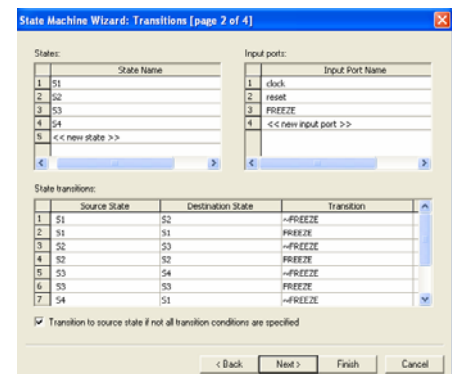
ในหน้าต่าง [Page 2 of 4]

ให้กำหนดพารามิเตอร์ลงในตารางซึ่งมี

- ชื่อและจำนวน State คือ **S1, S2, S3, S4**
- สัญญาณอินพุตที่ทำหน้าที่ควบคุมสเตตคือ **clk, reset, FREEZE**

- ตาราง State Transition โดยให้มีค่าตามแผนภาพ state diagram รูปที่ 5

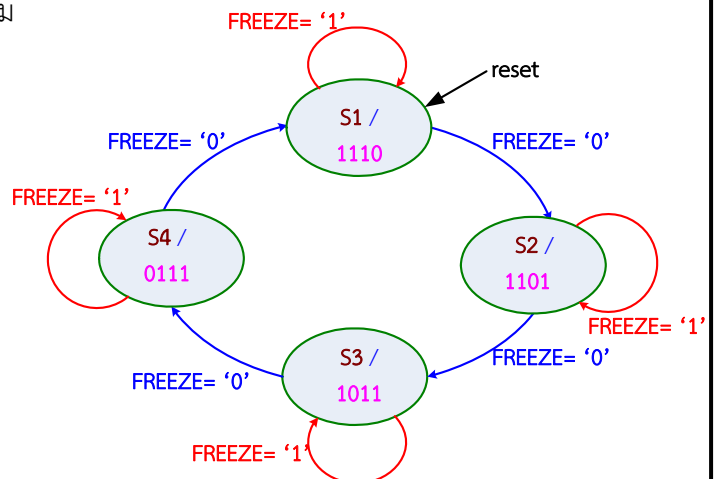
รูปที่ 4



รูปแบบของการกระทำทางลอจิกที่ใช้ในการเขียน state transition คือ

~ NOT & AND
| OR และ OTHERS

Input : FREEZE
Output : Row3, Row2, Row1, Row0,



รูปที่ 5

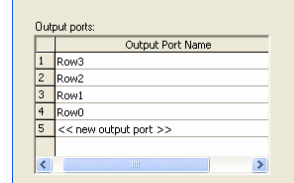


2.5 ที่หน้าต่าง [Page 3 of 4] ให้กำหนดพอร์ตเอาต์พุตชื่อ **Row0,Row1,Row2,Row3** ดังรูปที่ 6a จากนั้นนำค่าเอาต์พุตของแต่ละสเตตทั้ง 4 พอร์ตมาเขียนในตาราง Action condition ดังรูปที่ 6b

Action conditions:

	Output Port	Output Value	In State	Additional Conditions
1	Row3	1	S1	<< condition >>
2	Row2	1	S1	<< condition >>
3	Row1	1	S1	<< condition >>
4	Row0	0	S1	<< condition >>
5	Row3	1	S2	<< condition >>
6	Row2	1	S2	<< condition >>
7	Row1	0	S2	<< condition >>
8	Row0	1	S2	<< condition >>
9	Row3	1	S3	<< condition >>
10	Row2	0	S3	<< condition >>
11	Row1	1	S3	<< condition >>
12	Row0	1	S3	<< condition >>
13	Row3	0	S4	<< condition >>
14	Row2	1	S4	<< condition >>
15	Row1	1	S4	<< condition >>
16	Row0	1	S4	<< condition >>

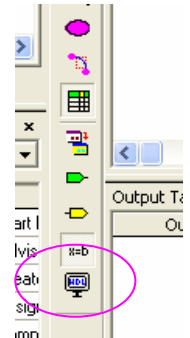
State Machine Wizard: Actions [page 3 of 4]



รูปที่ 6a

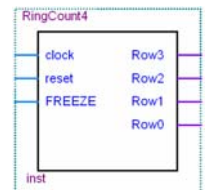
รูปที่ 6b

รูปที่ 6c



2.5 เสร็จแล้วหากไม่มีขั้นตอนใดผิดพลาด โปรแกรมก็จะทำการสร้างรูปแผนภาพของ state Diagram ให้ จากนั้นทำการบันทึกไฟล์และสร้างไฟล์ VHDL โดยเลือกที่แถบเครื่องมือ Generate VHDL File ดังรูปที่ 6c

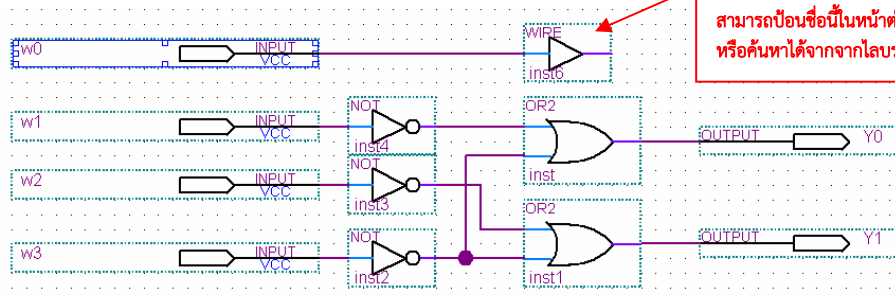
- เมื่อได้ไฟล์แบบ VHDL แล้วให้ทำการคอมไพล์ และสร้าง **symbol file** ของขึ้นมาเพื่อเตรียมไว้ใช้งานในขั้นตอนต่อไป เมื่อเสร็จขั้นตอนนี้เราจะได้อุปกรณ์ชื่อ **RingCount4** ดังรูปที่ 7 ถูกจัดเตรียมไว้ในไลบรารีรอให้นำไปใช้ใช้งานได้



รูปที่ 7

ขั้นที่ 2 เตรียมการสร้างวงจร(ระบบงาน) 4-to-2 Encoder

- ปิดโปรเจกต์ที่ทำมาในขั้นตอนที่ 2 – 3 ก่อนที่จะทำขั้นตอนต่อไป
- ทำการสร้างอุปกรณ์ 4-to-2 Encoder
 - สร้างโปรเจกต์ชื่อ **"Encoder4to2"** ให้ใช้ชิพเบอร์ EP3C10E144C8
 - สร้างไฟล์สำหรับเขียนแผนภาพวงจรในรูปที่ 8



หมายเหตุ: สำหรับไลบรารีของอุปกรณ์ที่ชื่อ WIRE น.ศ. สามารถป้อนชื่อในหน้าต่างของเมนู Symbol ได้โดยตรงหรือค้นหาได้จากจากไลบรารีของอุปกรณ์ประเภท buffer

รูปที่ 8

5.3 ทำการคอมไพล์และสร้าง symbol เพื่อเตรียมไว้ใช้งานในขั้นถัดไป

- ปิดโปรเจกต์ไฟล์ในขั้นตอนที่ 5 ก่อนทำขั้นตอนต่อไป



ขั้นที่ 3 สร้างหรือดัดแปลงวงจรแสดงผล 7-Segment ที่เคยสร้างไว้ให้แสดงตัวเลขที่เหมาะสมกับแป้นพิมพ์

A2: การถอดรหัสตำแหน่งของแป้นพิมพ์ Keypad

เมื่อพิจารณาวงจร keypad encoder จากรูปที่ 2 ประกอบกับตำแหน่งคีย์หมายเลข ในรูปที่ 9 จะพบว่า

ตำแหน่ง Row ถูกแสดงด้วยค่าของบิตเอาต์พุต D_3D_2

ตำแหน่ง Column ถูกแสดงด้วยค่าของบิตเอาต์พุต D_1D_0

ดังนั้นหากสมมติว่าเราลองกดแป้นพิมพ์เลข “1” เราจะถอดรหัสค่าของ $D_3D_2D_1D_0$ ได้ดังนี้

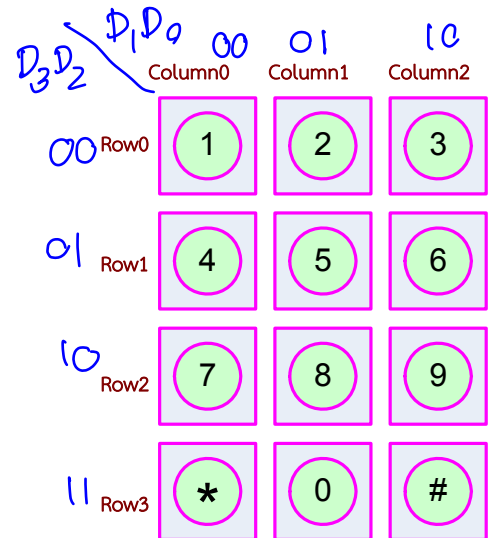
- เลข “1” อยู่ในตำแหน่ง Row0 , Column0

เมื่อ Row0= ‘0’ (ถูกกด) ดังนั้นค่าของ D_3D_2 = “00”

และ Column0= ‘0’ (ถูกกด) ค่าของ D_1D_0 = “00”

ดังนั้น

$D_3D_2D_1D_0$ = “0000” คือค่ารหัสที่ได้จากการกดแป้นเลข “1”

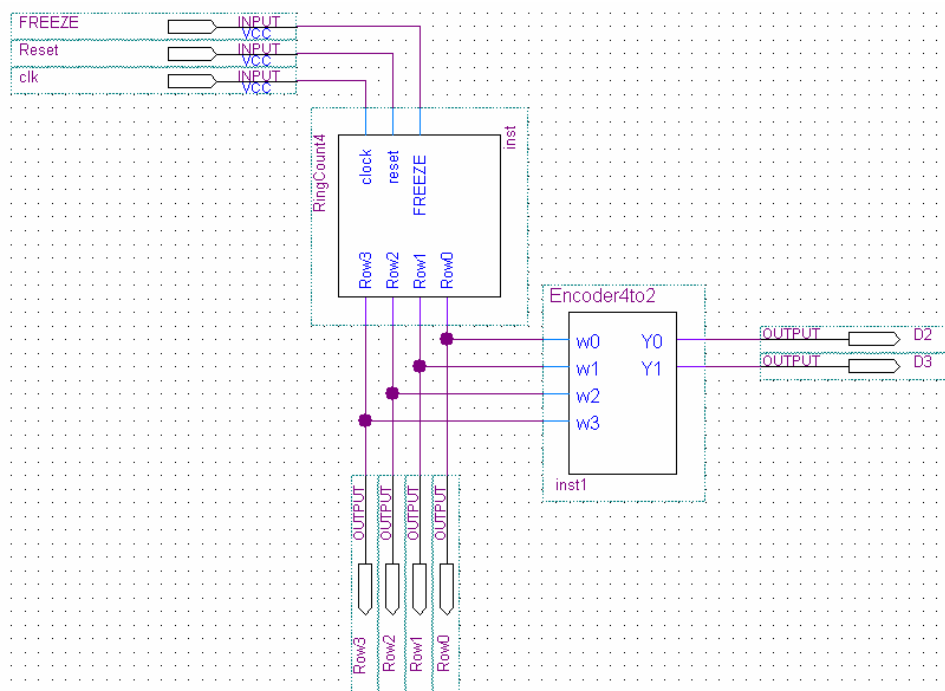


รูปที่ 9

A3: การทดลอง การถอดรหัสตำแหน่งของแป้นพิมพ์ ในแนว Row

7. ให้ทำการสร้างโปรเจกต์ขึ้นใหม่ (ให้ น.ศ. ตั้งชื่อโปรเจกต์เอง) และใช้ชิพเบอร์ EP3C10E144C8

a) เปิดไฟล์ขึ้นมาใหม่ (ใช้ชื่อเดียวกันกับโปรเจก) เพื่อเขียนวงจรในรูปที่ 10 แล้วทำการคอมไพล์

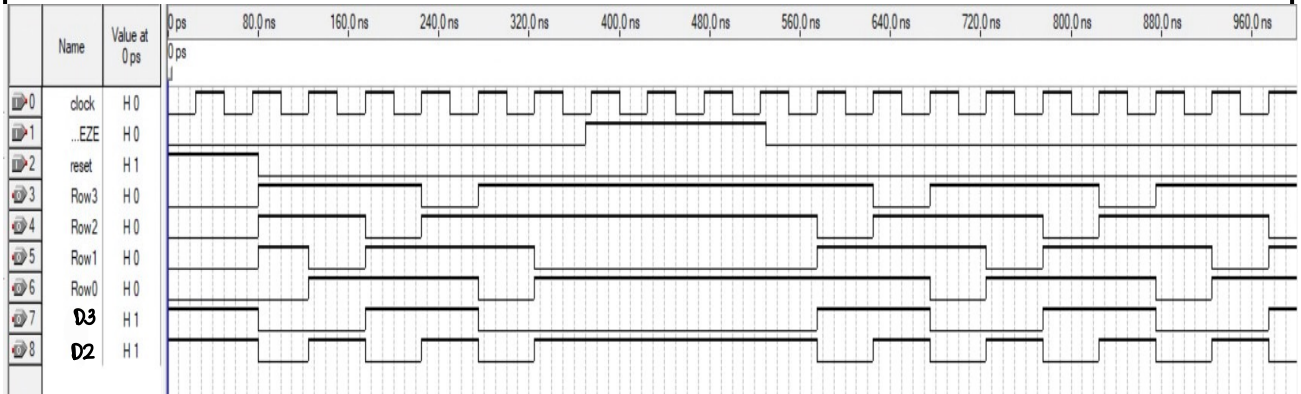


รูปที่ 10

b) สร้างไฟล์แสดงแผนภาพทางเวลา (Timing diagram) กำหนดพารามิเตอร์สำหรับแสดงผลจำลองการทำงานโดยให้มีค่า
End Time = 0.8 us Grid Size = 1 ns



- กำหนดสัญญาณอินพุต clk ให้เป็นแบบนาฬิกาที่มีความยาวคาบ (period) เป็น 50 ns
- กำหนดสัญญาณอินพุต Reset ให้เป็นลอจิก '1' ในช่วง 0 – 80 ns นอกนั้นให้เป็น '0'
- กำหนดสัญญาณอินพุต FREEZE ให้เป็นลอจิก '1' ในช่วง 370 – 530 ns นอกนั้นให้เป็น '0'
- จำลองการทำงานโหมด Functional mode บันทึกผลที่ได้ลงในรูปที่ 11



รูปที่ 11

สังเกตผลการทดลอง

- ก. ที่ขอบขาขึ้นของ clk ขอบที่ 1 และ 2 ค่าของ $Row_3Row_2Row_1Row_0 = "0000"$ เพราะ reset มีค่า '1' ที่ clk ขอบขาขึ้นที่ 3 ค่าของ $Row_3Row_2Row_1Row_0 = "1101"$ ค่าของ $D_3D_2 = "01"$ ที่ clk ขอบขาขึ้นที่ 4 ค่าของ $Row_3Row_2Row_1Row_0 = "1011"$ ค่าของ $D_3D_2 = "10"$ ที่ clk ขอบขาขึ้นที่ 5 ค่าของ $Row_3Row_2Row_1Row_0 = "0111"$ ค่าของ $D_3D_2 = "11"$ ที่ clk ขอบขาขึ้นที่ 6 ค่าของ $Row_3Row_2Row_1Row_0 = "1110"$ ค่าของ $D_3D_2 = "00"$ ที่ clk ขอบขาขึ้นที่ 7 ค่าของ $Row_3Row_2Row_1Row_0 = "1101"$ ค่าของ $D_3D_2 = "01"$ ที่ clk ขอบขาขึ้นที่ 8 ถึง 11 ค่าของ $Row_3Row_2Row_1Row_0 = "1101"$ ค่าของ $D_3D_2 = "01"$ เพราะ ท FREEZE มี Input เป็น 1 จึงดึงค่า Output ก่อนหน้าไว้

- ข. สัญญาณ FREEZE มีความหมายและความสำคัญอย่างไร FREEZE เป็นสัญญาณที่ตรึงค่าสัญญาณ Row Output ไว้ เพื่อไม่ให้ Output เปลี่ยนแปลง จนกว่า FREEZE เป็น 0 ค่า FREEZE = '1' ตามรูปที่ 11 แสดงว่ามีโอกาสที่เป็นหมายเลข 4, 5 หรือ 6 ถูกกด

- ค. ถ้าต้องการสร้างวงจรถอดรหัสแป้นพิมพ์ในแนว column บ้าง น.ศ. ควรจะออกแบบวงจรและวิธีการทดสอบอย่างไร (อธิบายให้อาจารย์สัมผัสได้ว่า น.ศ. มีความรู้จริงๆ)

- ① เปลี่ยน Output ที่ออกมา Ring Counter จาก $Row[3..0] \rightarrow Column[2..0]$
- ② ท $W[2..0]$ ที่ต่อกับ $Row[3..0]$ จึงเปลี่ยนไปต่อกับ $Column[2..0]$ แต่ w_3 ต่อกับ vcc
- ③ Output ของ Encoder เปลี่ยนจาก $D_3, D_2 \rightarrow D_1, D_0$

- ง. จากข้อ ค. และรูปที่ 9 ให้เขียนรหัสค่า $D_3D_2D_1D_0$ ที่แทนเลขต่างๆ ของแป้นพิมพ์ทั้ง 12 คีย์ลงในตารางที่ 1

แป้นคีย์	เลข 1	เลข 2	เลข 3	เลข 4	เลข 5	เลข 6	เลข 7	เลข 8	เลข 9	เลข 0	#	*
$D_3D_2D_1D_0$	0000	0001	0010	0100	0101	0110	1000	1001	1010	1011	1100	1100

ตารางที่ 1

ลายเซ็นอาจารย์ผู้ควบคุม.....



8. ให้นำโปรแกรมส่วนของการแสดงผล 7-segment ที่ได้เคยสร้างไว้ในการทดลองที่ 8 (ชื่อ **VHD_7SEGM.vhd**) มาทำการดัดแปลงสำหรับแสดงค่าของสวิตช์ Keypad ให้สามารถแสดงผลได้เป็นค่าของปุ่มสวิตช์แต่ละปุ่มโดยอาศัยความสัมพันธ์กัน ระหว่างค่ารหัสของ $D_3D_2D_1D_0$ กับ ค่าตัวเลข ของแป้นคีย์ (ที่ น.ศ.ได้ทำไว้แล้วในตารางที่ 1) และให้บันทึกเป็นไฟล์ใหม่ชื่อ “KeypadDisplay.vhd” ดังรูปที่ 12

```

LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY KeypadDisplay IS
    PORT (
        En : IN STD_LOGIC;
        Key_D : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
        SEGM : OUT STD_LOGIC_VECTOR(1 TO 7)
    );
END KeypadDisplay ;

ARCHITECTURE Behavior OF KeypadDisplay IS
    SIGNAL temp : STD_LOGIC_VECTOR(1 TO 7);
BEGIN
    PROCESS (En, Key_D)
    BEGIN
        IF En = '1' THEN
            CASE Key_D IS
                WHEN "0000" => temp <= "0110000"; --แสดงเลข 1
                WHEN "0001" => temp <= "1101101"; --แสดงเลข 2
                WHEN "0010" => temp <= "1111001"; --แสดงเลข 3
                WHEN "0011" => temp <= "0110011"; --แสดงเลข 4
                WHEN "0100" => temp <= "1011011"; --แสดงเลข 5
                WHEN "0101" => temp <= "1011111"; --แสดงเลข 6
                WHEN "0110" => temp <= "1110000"; --แสดงเลข 7
                WHEN "0111" => temp <= "1111111"; --แสดงเลข 8
                WHEN "1000" => temp <= "1111011"; --แสดงเลข 9
                WHEN "1001" => temp <= "1101011"; --แสดงคีย์ #
                WHEN "1010" => temp <= "1111110"; --แสดงเลข 0
                WHEN "1011" => temp <= "1011101"; --แสดงคีย์ *
                WHEN OTHERS => temp <= "0000000"; --ให้ทุกเช็กเมนต์ดับหมด
            END CASE ;
        ELSE
            temp <= "0000000"; --ให้ทุกเช็กเมนต์ดับหมด
        END IF;
    END PROCESS ;
    SEGM <= NOT temp ;
END Behavior ;
    
```

ให้นำค่าของ $D_3D_2D_1D_0$ (4บิต) จาก
ตารางที่ 1 มาเขียนลงในโปรแกรม

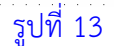
รูปที่ 12

ลายเซ็นอาจารย์ผู้ควบคุม.....

9. ให้ทำการสร้างโปรเจกต์ขึ้นใหม่ชื่อ “KeypadDisplay” และใช้ชิพเบอร์ EP3C10E144C8
- เปิดไฟล์ขึ้นมาใหม่ (ใช้ชื่อเดียวกันกับโปรเจก) เพื่อเขียนโปรแกรมตามในรูปที่ 12
 - ทำการคอมไพล์และสร้าง symbol file ไว้ใช้
 - เมื่อดำเนินการเสร็จแล้วให้ปิดโปรเจกต์ก่อนทำขั้นตอนต่อไป

ขั้นที่ 4 สร้างระบบที่สมบูรณ์ของ วงจรอ่านแป้นพิมพ์ Keypad Matrix4x3 ให้แสดงผลผ่าน 7-Segment

10. สร้างโปรเจกต์ชื่อ “Keypad” ขึ้นมาและให้เก็บไว้ในโฟลเดอร์เดียวกันกับโปรเจกต์ที่สร้างตอนก่อนหน้า จากนั้นสร้างไฟล์ (ใช้ชื่อเดียวกันกับโปรเจก) ขึ้นมาสำหรับเขียนวงจรทดลองดังรูปที่ 13 ใช้ชิพเบอร์ EP3C10E144C8 ทำการคอมไพล์และทดสอบการทำงานของระบบ



1. ให้ น.ศ. ทดลองกดปุ่มแป้นคีย์ หมายเลขต่างๆ พร้อมสังเกตตัวเลขที่ปรากฏ ของ 7-Segment ว่าค่าถูกต้องตรงกันหรือไม่

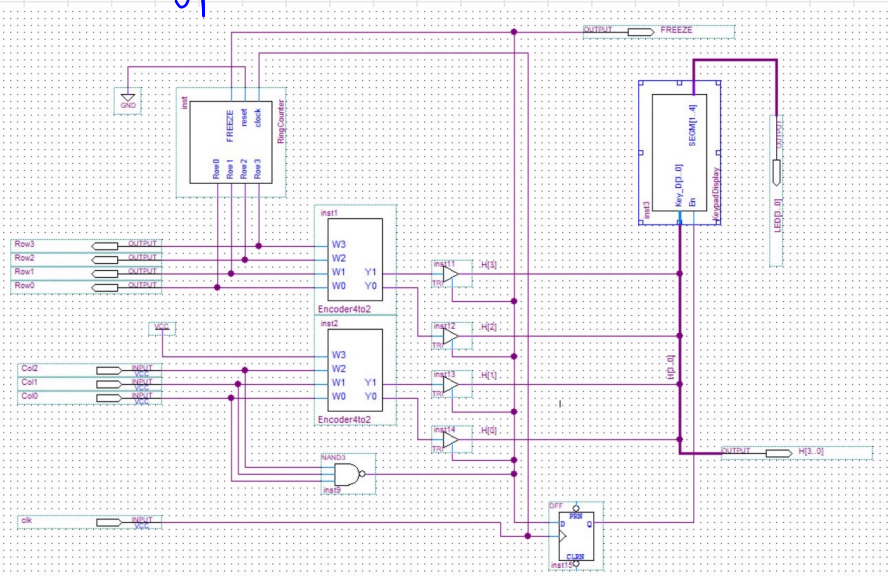
1. ถ้าต้องการให้ตัวเลขที่ปรากฏนั้นยังคงติดค้างค่าสุดท้ายไว้แม้ว่าจะเลิกกดปุ่มแล้ว จะต้องดัดแปลงหรือสร้างเพิ่มเติมวงจรอย่างไรบ้าง (แนะนำให้ใช้อุปกรณ์ประเภท Shift-Register หรือ DFF มาช่วย)

(ให้เขียนลงบนกระดาษ A4 ที่มีเส้นบรรทัดและรวมใส่ท้ายเอกสารการทดลอง)

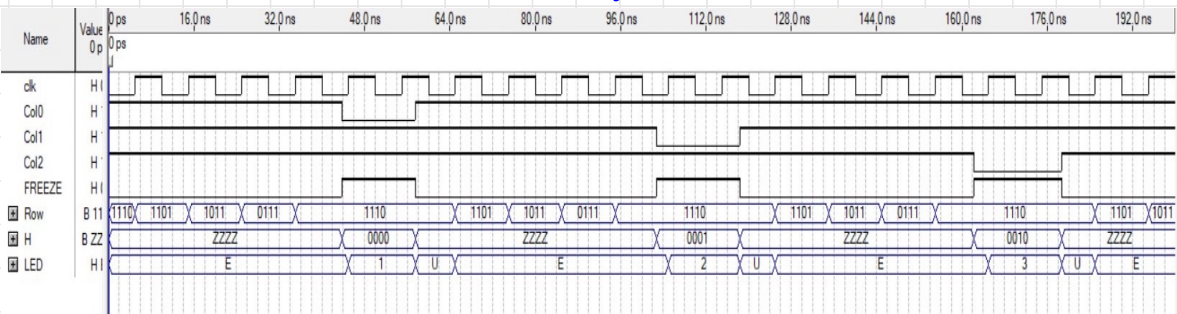
1. ให้ทำรายงาน อธิบายคำสั่งภาษา VHDL ของวงจรที่ได้จากการแปลงจาก state diagram ในการทดลองรูปที่ 5 ทุกคำสั่ง
2. ให้สรุปความรู้ที่ได้จากการทดลองนี้

ถ้าต้องการสร้างเครื่องคิดเลขอย่างง่าย มีเพียง การบวกและลบเลขจำนวนเต็ม ที่ให้คำตอบไม่เกิน 4 หลัก โดยใช้ความรู้ทั้งหมดที่ได้เรียนมาในห้องปฏิบัติการ เราควรจะออกแบบระบบงานของเครื่องคิดเลขดังที่ได้กล่าวข้างต้น
นี้อย่างไร ?

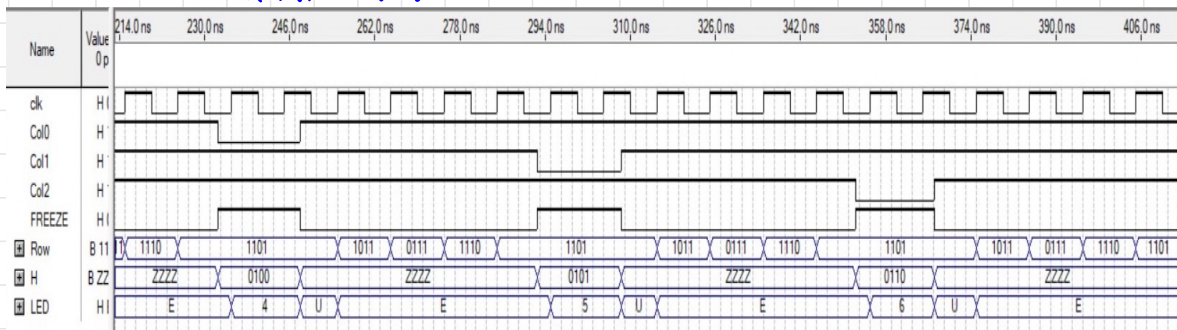
Keypad



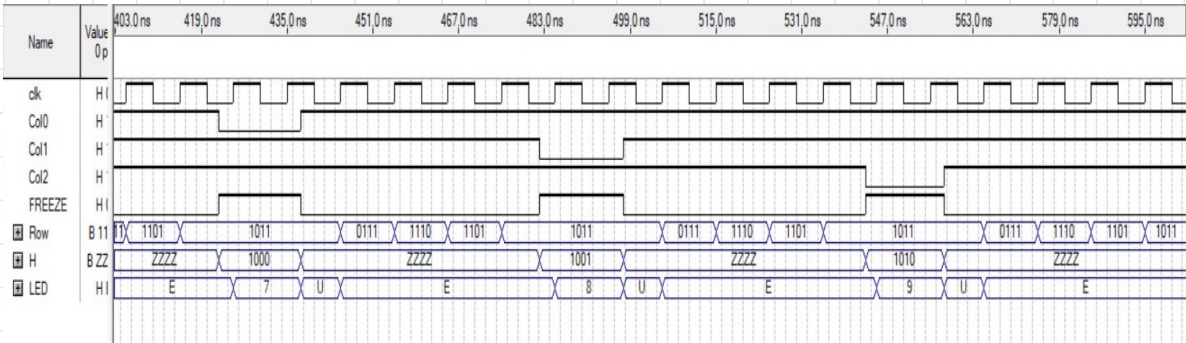
warns Simulate 1000 0-200 ns



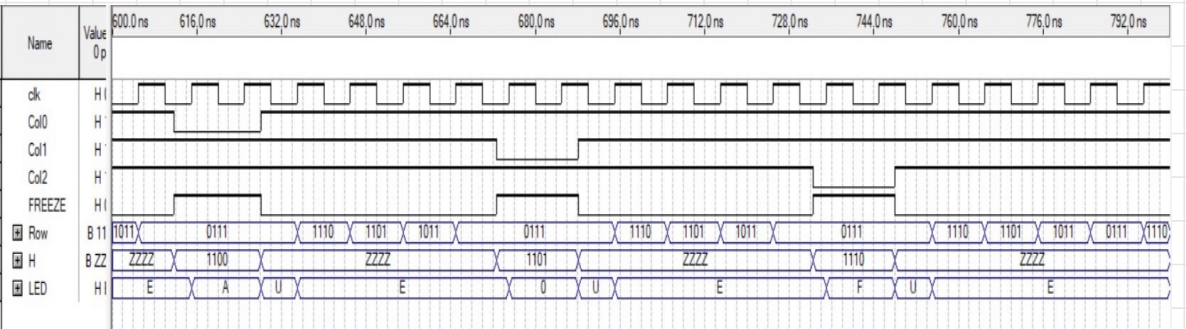
warns Simulate 9600 200-400 ns



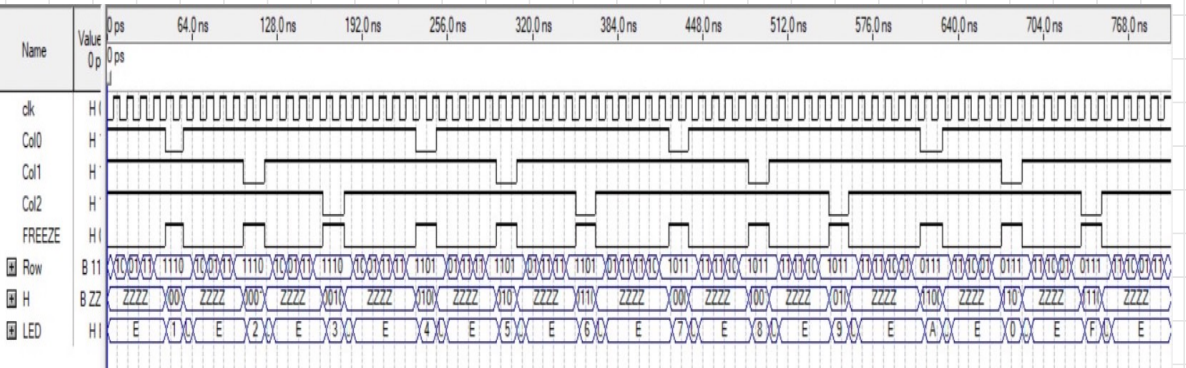
warns Simulate 400-600 ns



warns Simulate 600-800 ns



ภาพรวมของวงจรทั้งหมด



1. ถ้าต้องการให้ตัวเลขที่ปรากฏนั้นยังคงติดค้างค่าสุดท้ายไว้แม้ว่าจะเลิกกดปุ่มแล้ว จะต้องดัดแปลงหรือสร้างเพิ่มเติมวงจรอย่างไรบ้าง (แนะนำให้ใช้อุปกรณ์ประเภท Shift-Register หรือ DFF มาช่วย)

```

15 IF En = '1' THEN
16 CASE Key_D IS --???????? Segment: ?€?Digit?€
17 WHEN "0000" => temp <= "0001" ; --???????? 1
18 WHEN "0001" => temp <= "0010" ; --???????? 2
19 WHEN "0010" => temp <= "0011" ; --???????? 3
20 WHEN "0100" => temp <= "0100" ; --???????? 4
21 WHEN "0101" => temp <= "0101" ; --???????? 5
22 WHEN "0110" => temp <= "0110" ; --???????? 6
23 WHEN "1000" => temp <= "0111" ; --???????? 7
24 WHEN "1001" => temp <= "1000" ; --???????? 8
25 WHEN "1010" => temp <= "1001" ; --???????? 9
26 WHEN "1100" => temp <= "1010" ; --???????? #
27 WHEN "1101" => temp <= "0000" ; --???????? 0
28 WHEN "1110" => temp <= "1111" ; --???????? *
29 WHEN OTHERS => temp <= "1110" ; --.????????????????????
30 END CASE ;
31
32
33 END IF;
34 END PROCESS ;

```


wamy Stimulate

