บทที่ 4

การออกแบบชิปซีเรียลเทเลเท็กซต์เดต้าชิฟเตอร์

4.1 เหตุผลในการใช้ชิป FPGA (Field Programmable Gate Array)

ในการทำวิทยานิพนธ์นี้ เป็นการนำแบบจำลองภาษาวีเอชดีแอลมาใช้ในการออกแบบวงจร ซึ่งมีประโยชน์ดังที่กล่าวไว้ในบทที่ 2 คือ สามารถสร้างวงจรและจำลองการทำงานได้โดยไม่ต้องทำการต่อหรือประกอบวงจรขึ้นจริง ดังนั้นเมื่อนำแบบจำลองภาษาวีเอชดีแอลมาใช้ จึงพยายามนำไปใช้ออกแบบวงจรเชิงเลขให้ได้มากที่สุดสำหรับงานวิจัยนี้ โดยใช้ซอฟต์แวร์ชื่อ WVPLUS 5.11 ของบริษัท ViewLogic เป็นอุปกรณ์ในการพัฒนาแบบจำลองภาษาวีเอชดีแอลบนเครื่องคอมพิวเตอร์ส่วนบุคคล ซึ่งเมื่อออกแบบและจำลองการทำงานได้ตามความต้องการแล้วก็จะนำไปสังเคราะห์ลงในชิป FPGA ของบริษัท XILINX โดยใช้ซอฟต์แวร์ที่ชื่อว่า XACT ซึ่งเป็นซอฟต์แวร์ของบริษัท XILINX เอง ตัวชิป FPGA ของบริษัท XILINX มีด้วยกันหลายเทคโนโลยี หลายขนาด และหลายแบบ จึงทำให้สามารถเลือกใช้ชิป FPGA ที่เหมาะสมกับวงจรเชิงเลขที่ออกแบบได้ เมื่อนำชิป FPGA ที่สังเคราะห์ขึ้นมาประกอบลงแผ่นวงจรก็จะช่วยทำให้แผ่นวงจรมีขนาดเล็กลงกว่าการนำการนำอุปกรณ์หลาย ๆ ชิ้นมาต่อประกอบกัน เพราะว่าการทำงานของวงจรต่าง ๆ ได้ถูกออกแบบไว้ภายในชิป FPGA เพียงตัวเดียว

4.2 หน้าที่และการทำงานของชิปซีเรียลเทเลเท็กซต์เดต้าชิฟเตอร์

หน้าที่หลักของซีเรียลเทเลเท็กซต์เดต้าชิฟเตอร์นี้คือ ทำการอ่านข้อมูลจากหน่วยความจำแรมหรือรอมแล้วทำการเลื่อนหรือชิฟต์ข้อมูลออกไปแบบอนุกรม ด้วยอัตราเร็ว 6.9375 เมกกะบิตต่อวินาที หรือด้วยความถี่นาฬิกา 6.9375 เมกกะเฮิรตซ์ และจะต้องสามารถเลือกเส้นสแกนในการแทรกข้อมูลได้ด้วย

การทำงานของชิปดังที่เคยกล่าวบ้างแล้วในบทที่ 3 คือ จะเริ่มจากการที่ชิปอ่านข้อมูลตั้งแต่แอดเดรส F0000H ถึงแอดเดรส F0002H ได้ข้อมูลมาจำนวน 3 ไบต์ ซึ่งข้อมูลที่ได้มานี้จะเป็นแอดเดรสเริ่มต้นของหน้าข้อมูลเทเลเท็กซต์ที่ต้องการส่ง และจะทำการส่งข้อมูลออกไป 25 แถวหรือ 25 เส้นสแกนหรือครบ 1,050 ไบต์ ก็จะกลับมา อ่านข้อมูลที่อยู่ถัดไปจากแอดเดรส F0002H อีก 3 แอดเดรส เพื่อใช้เป็นแอดเดรสเริ่มต้นที่จะทำการส่งข้อมูลอีก และเมื่อทำการส่งข้อมูลออกไปครบ 25 เส้นสแกนก็จะกลับมาอ่านข้อมูลจากแอดเดรสที่ถัดไปอีก 3 แอดเดรส เพื่อใช้เป็นแอดเดรสเริ่มต้นในการส่งอีก

จะเป็นเช่นนี้ต่อไปเรื่อย ๆ แต่เมื่อไรที่อ่านข้อมูลได้เป็น 00H ทั้งสามแอดเดรสก็จะกลับไปเริ่มอ่านข้อมูลที่แอดเดรส F0000H ใหม่เป็นอย่างนี้อยู่ตลอดเวลา ดังนั้นที่ตำแหน่งแอดเดรส F0000H เป็นต้นไปจะเป็นข้อมูลของแอดเดรสเริ่มต้นของข้อมูลเทเลเท็กซต์แต่ละหน้าที่ต้องการจะทำการส่งออกไป

4.3 การออกแบบชิปซีเรียลเทเลเท็กซต์เดต้าชิฟเตอร์

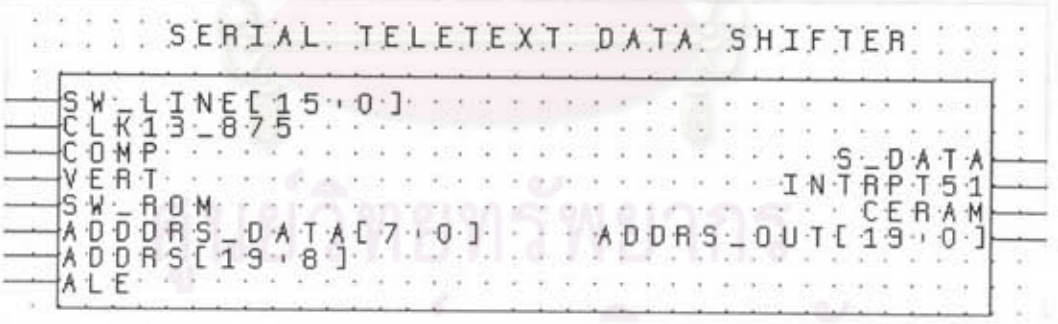
ในการออกแบบชิปเทเลเท็กซต์นี้ได้แบ่งการทำงานตามช่วงเวลาออกเป็น 2 ช่วงเวลา คือ

1.ช่วงเส้นสแกนที่ 7(320) ถึง เส้นที่ 22(335)

2.ช่วงที่อยู่นอกช่วงเวลาของเส้นสแกนที่ 7(320) ถึง 22(335)

เหตุที่ทำการแบ่งออกเป็นสองช่วงเวลา เพราะชิปนี้จะทำหน้าที่ชิฟต์ข้อมูลแบบอนุกรมออกมาในช่วงเวลาที่อยู่ระหว่างเส้นสแกนที่ 7(320) ถึง เส้นที่ 22(335) เท่านั้น เพื่อสามารถจะนำไปรวมกับสัญญาณภาพในระบบเวลาจริงได้เลย นอกเหนือจากช่วงเวลาดังกล่าวชิปซีเรียลเทเลเท็กซต์เดต้าชิฟเตอร์จะถูกใช้เป็นวงจรแลตช์สำหรับพอร์ตแอดเดรสของชิปไมโครคอนโทรลเลอร์ MCS-51 เพื่อใช้ในการเคลื่อนย้ายข้อมูลไปไว้ในหน่วยความจำแรม เท่านั้น

รูปที่ 4.1 เป็นการแสดงให้เห็นอินพุตและเอาต์พุตของชิปซีเรียลเทเลเท็กซต์เดต้าชิฟเตอร์ที่ออกแบบ เพื่อให้เห็นว่าชิปที่ทำการออกแบบมีลักษณะอย่างไร ใช้ทำอะไร ก่อนที่จะอธิบายการออกแบบแผนภาพภายในโดยละเอียดต่อไป



รูปที่ 4.1 อินพุตเอาต์พุตของชิปซีเรียลเทเลเท็กซต์เดต้าชิฟเตอร์

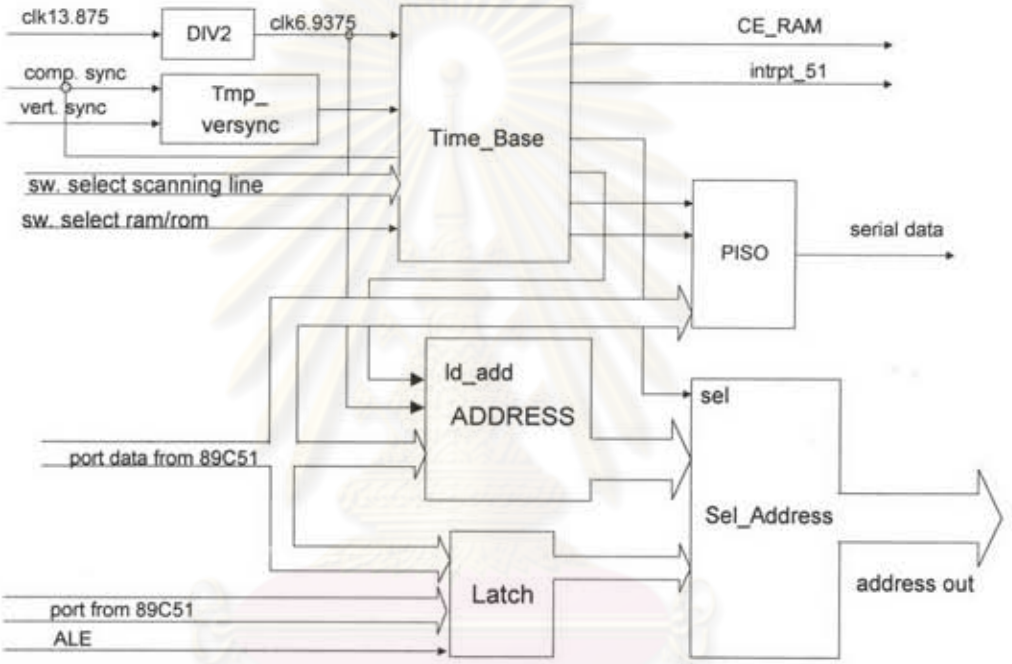
- SW_LINE เป็นขาอินพุตที่ต่อกับสวิตช์ที่ใช้สำหรับเลือกเส้นกวาดหรือเส้นสแกนในการส่งข้อมูลเทเลเท็กซต์ มีทั้งหมดจำนวน 16 อินพุต เพื่อใช้เลือกตั้งแต่เส้นกวาดที่ 7(320) ถึง เส้นที่ 22(335)

- CLK13_875 เป็นอินพุตที่รับความถี่นาฬิกา 13.875 เมกกะเฮิรตซ์เข้ามา
- COMP เป็นอินพุตที่รับสัญญาณซิงก์รวมที่ได้จากเอาต์พุตของวงจรแยกซิงก์เข้ามาเพื่อนำไปใช้ในการออกแบบวงจรนับเส้นกวาดหรือเส้นสแกน
- VERT เป็นอินพุตที่รับสัญญาณซิงก์แนวตั้งที่ได้จากเอาต์พุตของวงจรแยกซิงก์เพื่อนำไปใช้ในการนับเส้นสแกน เช่นเดียวกัน
- SW_ROM เป็นอินพุตที่ต่อกับสวิตช์ที่ใช้สำหรับเลือกอ่านข้อมูลว่าจะอ่านข้อมูลจากหน่วยความจำแรมหรือรอม
- ADDRS_DATA เป็นอินพุตที่ต่อกับพอร์ตแอดเดรสและข้อมูลของชิปไมโครคอนโทรลเลอร์พอร์ต 0 มีจำนวน 8 ขาหรือ 8 บิต
- ADDRS เป็นอินพุตสำหรับแอดเดรสจำนวน 12 บิต ซึ่งจะต่อกับพอร์ต 2 ของชิปไมโครคอนโทรลเลอร์ซึ่งเป็นพอร์ตแอดเดรสจำนวน 8 บิต และพอร์ต 1 อีก 4 บิตที่เขียนโปรแกรมให้ใช้เป็นแอดเดรส
- ALE เป็นสัญญาณที่ต่อกับขา ALE ของชิปไมโครคอนโทรลเลอร์ เพื่อใช้ในการแลตช์ค่าแอดเดรสก่อนที่จะทำการเขียนข้อมูลลงในแรม
- S_DATA เป็นเอาต์พุตของข้อมูลที่ถูกชิฟต์ออกมาแบบอนุกรม มีสัญญาณลอจิกเป็น "1" หรือ "0" เท่านั้น
- INTRPT51 เป็นเอาต์พุตที่จะให้ลอจิกเป็น "1" ในช่วงของเส้นสแกนที่ 7(320) ถึงเส้นที่ 22(335) เพื่อนำไปใช้ในการสร้างสัญญาณ CTS
- CERAM เป็นเอาต์พุตที่ใช้สำหรับเลือกหน่วยความจำรอมหรือแรม
- ADDRS_OUT เป็นเอาต์พุตที่จะต่อไปยังแอดเดรสของหน่วยความจำรอมและแรม มีจำนวน 20 เอาต์พุต หรือ 20 บิต

แนวความคิดในการออกแบบชิป [2]

วิธีที่ใช้ในการออกแบบจะมี 2 วิธีด้วยกัน คือ การออกแบบจากบนลงล่าง (Top down design) และการออกแบบจากล่างขึ้นบน (Bottom up design) ซึ่งในการออกแบชิปเทเลเท็กซต์เดต้าชิฟเตอร์นี้จะต้องใช้ทั้งสองวิธีประกอบกันเนื่องจากเป็นชิปที่รวมวงจรที่ใหญ่มากเอาไว้ โดยมีแนวความคิดที่จะออกแบบตามหน้าที่หลักของชิป คือ ชิปเทเลเท็กซต์เดต้าชิฟเตอร์นี้จะมีหน้าที่ในการชิฟต์ข้อมูลแบบขนาน 8 บิต ออกเป็นข้อมูลแบบอนุกรมทีละบิต จึงเริ่มทดลองออกแบบบล็อก PISO (Parallel In Serial Out) ซึ่งบล็อก PISO นี้จะต้องมีสัญญาณนาฬิกาความถี่ 6.9375 เมกกะเฮิรตซ์ เพื่อใช้ในการชิฟต์

ข้อมูล ตามจังหวะของเวลาที่สามารถนำข้อมูลที่สามารถนำไปรวมกับสัญญาณภาพวิดีโอได้เลย จึงทำให้ต้องสร้างบล็อก Time_Base ขึ้น เพื่อกำเนิดสัญญาณต่างที่ใช้ในการควบคุมหรือสัญญาณทริกก์ให้ถูกต้องของจังหวะเวลา และเนื่องจากหน้าที่ของชิปเทเลเท็กซต์เดต้าชิฟเตอร์มีด้วยกันหลายอย่าง เช่น มีการโหลดค่าแอดเดรส, การนับแอดเดรส, การแลตช์ค่าของพอร์ตแอดเดรส เป็นต้น จึงได้ทำการออกแบบวงจรแต่ละส่วนแล้วทำการจำลองการทำงาน (Simulation) จนได้วงจรที่ทำงานได้ตามต้องการแล้วจึงนำมาประกอบเข้าไปในการออกแบบ ซึ่งบล็อกต่าง ๆ แสดงไว้ดังรูปที่ 4.2



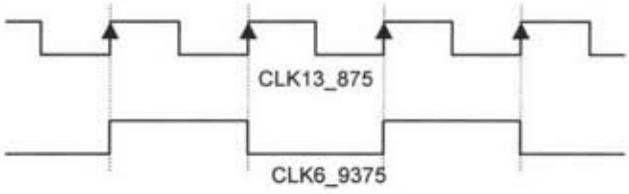
รูปที่ 4.2 แผนภาพบล็อกภายในชิปซีเรียลเทเลเท็กซต์เดต้าชิฟเตอร์

1. DIV2 จะเป็นบล็อกที่จะนำความถี่ 13.875 เมกกะเฮิรตซ์มาหารสอง เพื่อให้ได้ความถี่ 6.9375 เมกกะเฮิรตซ์ ไปใช้ในบล็อกต่าง ๆ ต่อไป

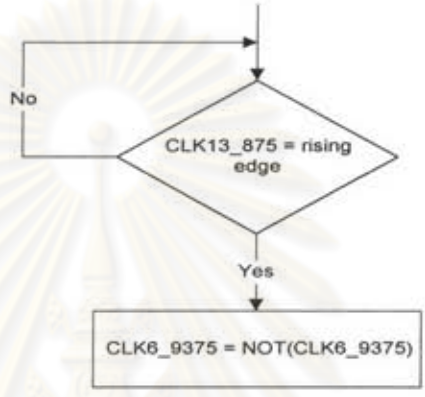
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

แนวทางในการออกแบบ DIV2

ในการออกแบบวงจรหารสองจะออกแบบได้สองแบบด้วยกันคือ จะใช้ขอบขาขึ้นหรือขอบขาลงของสัญญาณนาฬิกาในการท็อกเกิลสัญญาณ ดังแผนภาพในรูปที่ 4.3 จะใช้ขอบขาขึ้นในการออกแบบ



รูปที่ 4.3 ภาพสัญญาณของ DIV2



รูปที่ 4.4 แผนภาพการออกแบบ DIV2

จากแผนภาพในรูปที่ 4.4 สามารถออกแบบโดยใช้ภาษาวีเอชดีแอลได้ดังรูปที่ 4.5 และเนื่องจากความถี่ 6.9375 เมกกะเฮิรตซ์ จะถูกนำไปใช้เป็นสัญญาณนาฬิกาของฟลิปฟลอปภายในบล็อกต่าง ๆ อีกมากมาย ดังนั้นเพื่อเพิ่มความสามารถในการขับและลดค่าประวิงเวลาลงจึงนำอุปกรณ์ที่ชื่อ ACLK มาต่อเข้ากับเอาต์พุต CLK6_9375 ก่อนที่จะไปจ่ายให้กับฟลิปฟลอปในบล็อกต่าง ๆ ดังแสดงไว้ในภาคผนวก

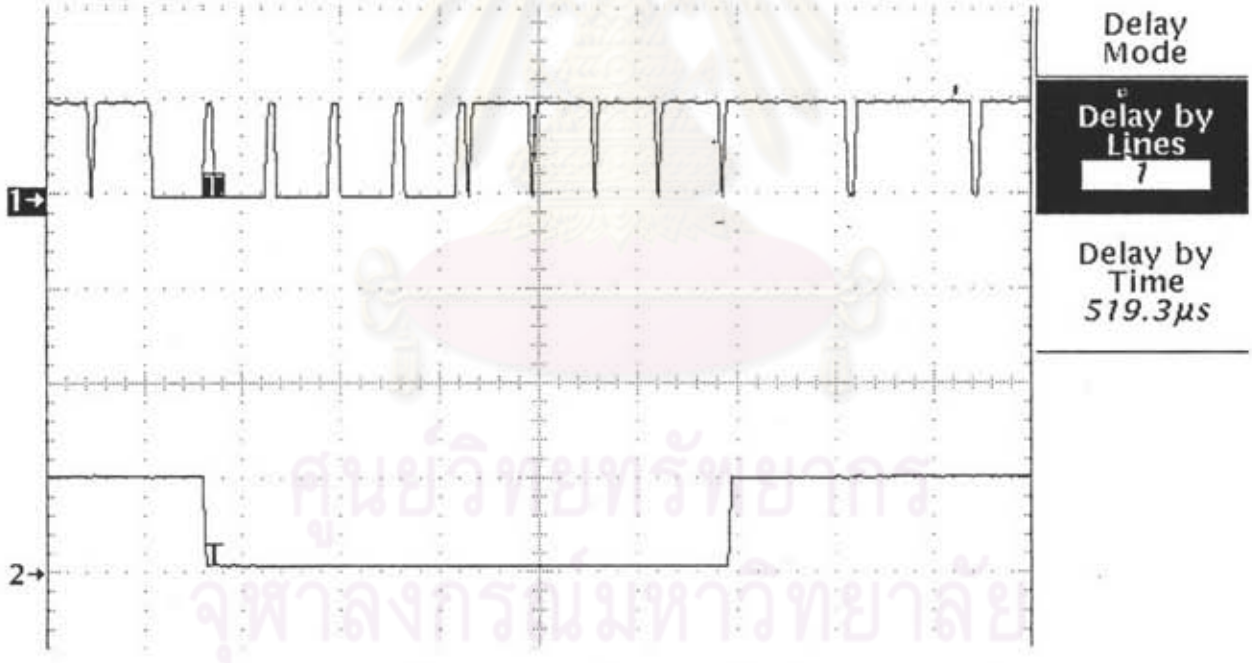
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
ENTITY div2 IS
PORT    (
                clk13_875           : in  vlbit;
                clk6_9375           : out vlbit
        );
END div2;
ARCHITECTURE behavioral OF div2 IS
SIGNAL          tmp_clk             : vlbit;
BEGIN
        PROCESS
        BEGIN
```

```
                WAIT UNTIL PRISING(clk13_875);
                tmp_clk <= NOT(tmp_clk);
                END PROCESS;
                clk6_9375 <= tmp_clk;
        END behavioral;
```

รูปที่ 4.5 โปรแกรมภาษาวีเอชดีแอลของบล็อก DIV2

2. TMP_VERSYN เป็นบล็อกที่จะทำการสร้างสัญญาณที่คล้ายกับซิงก์แนวตั้งขึ้นใหม่ เนื่องจากสัญญาณขอบขาขึ้นของซิงก์แนวตั้งจะค่อนข้างตรงกับสัญญาณขอบขาขึ้นของสัญญาณซิงก์รวมดังแสดงในรูปที่ 4.6 ซึ่งจะทำให้เกิดความผิดพลาดในการนับเส้นสแกนขึ้นได้ จึงนำสัญญาณซิงก์รวมและซิงก์แนวตั้งมาสร้างสัญญาณซิงก์แนวตั้งขึ้นใหม่

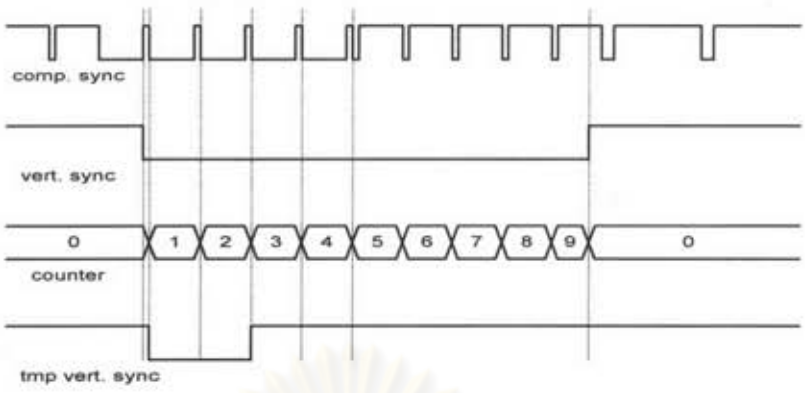


รูปที่ 4.6 สัญญาณซิงก์รวมและซิงก์แนวตั้งจากวงจรแยกซิงก์

แนวทางในการออกแบบ TMP_VERSYN

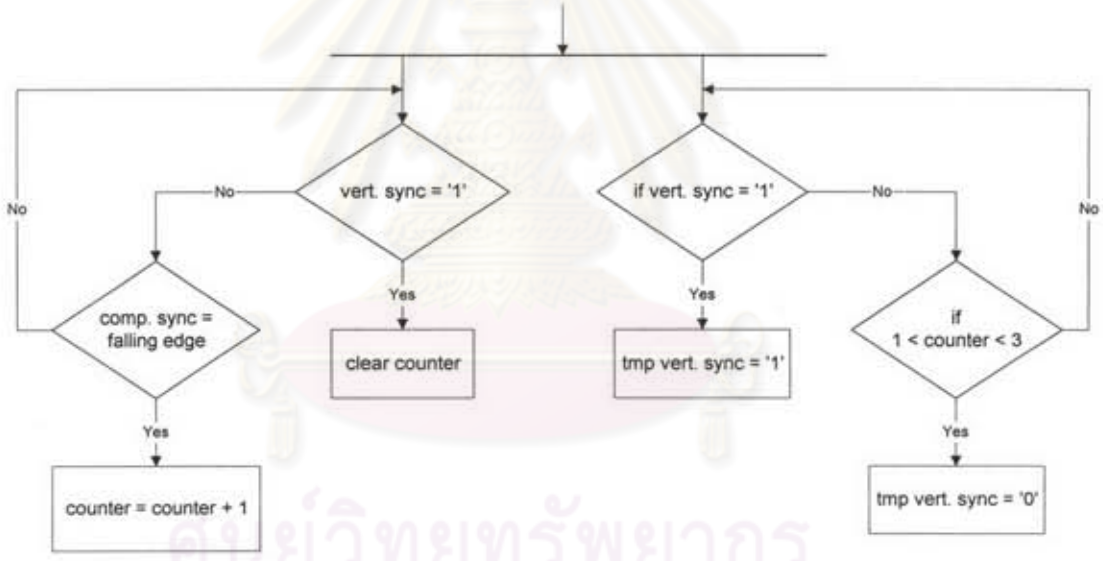
ในการออกแบบได้นำสัญญาณซิงก์รวมมาเป็นสัญญาณนาฬิกา และสัญญาณซิงก์แนวตั้งเป็นสัญญาณลบล้าง(clear) ซึ่งสัญญาณซิงก์ใหม่จะแสดงไว้ดังรูปที่ 4.7

I 17295683



รูปที่ 4.7 ความสัมพันธ์ของสัญญาณที่ใช้ออกแบบ TMP_VERSYNC

จากแผนภาพทางเวลาของสัญญาณในรูปที่ 4.7 สามารถนำไปออกแบบเป็นแผนภาพสำหรับออกแบบภาษาวีเอชดีแอลได้ดังรูปที่ 4.8



รูปที่ 4.8 แผนภาพการออกแบบ TMP_VERSYNC

จากแผนภาพสามารถเขียนเป็นภาษาวีเอชดีแอลได้ดังรูปที่ 4.9

```
ENTITY t_versyn IS
PORT (
        composite           : in  vlbit;
        vertical            : in  vlbit;
        tmp_vert_sync       : out vlbit
    );
END t_versyn;
```

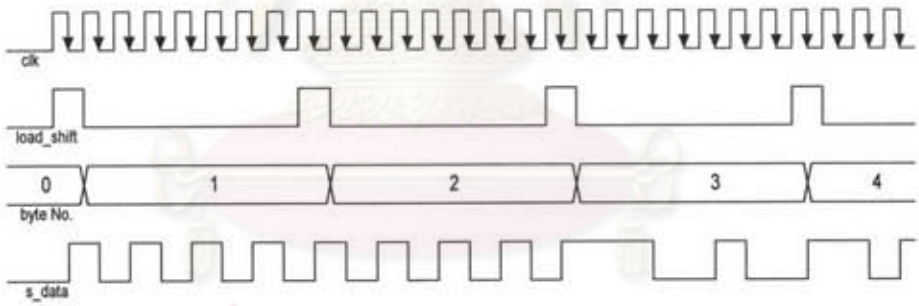
```
ARCHITECTURE behavioral OF t_versyn IS
SIGNAL   cnt                          : vlbit_1d(4 downto 0);
BEGIN
        PROCESS
        BEGIN
        WAIT UNTIL (PFALLING(composite) OR (vertical = '1'));
        IF vertical = '1' THEN
          cnt <= "00000";
        ELSE
          cnt <= ADDUM(cnt(3 downto 0),"001");
        END IF;
        END PROCESS;

        PROCESS(vertical,cnt)
        BEGIN
        IF vertical = '1' THEN
          tmp_vert_sync <= '1';
        ELSE
          IF cnt(3 downto 0) < B"0001" THEN
            tmp_vert_sync <= '1';
          ELSE
            IF cnt(3 downto 0) <= B"0010" THEN
                tmp_vert_sync <= '0';
            ELSE
                tmp_vert_sync <= '1';
            END IF;
          END IF;
        END IF;
        END PROCESS;

END behavioral;
```

รูปที่ 4.9 โปรแกรมภาษาวีเอชดีแอลของบล็อก TMP_VERSYNC

3. PISO เป็นบล็อกที่อ่านข้อมูล 8 บิต แล้วทำทำการเลื่อนออกไปทีละบิตด้วยอัตรา 6.9375 เมกกะบิตต่อวินาที นอกจากนี้ยังออกแบบให้สร้างสัญญาณสำหรับ Clock Run-in และ Framing Code ขึ้น เนื่องจากได้ทำการศึกษาโครงสร้างของข้อมูลเทเลเท็กซต์ดังที่กล่าวไว้ในบทที่ 2 จะเห็นว่าข้อมูลที่เป็น Clock Run-in และ Framing Code ในทุก ๆ แถวข้อมูลจะมีลักษณะเหมือนกันโดยตลอดคือมีข้อมูลเป็น 55H และ 27H ตามลำดับ จึงมีแนวความคิดที่จะลดจำนวนข้อมูลที่จะต้องเก็บในหน่วยความจำแรมและรอม โดยทำการออกแบบเป็นวงจรในการสร้างสัญญาณที่เป็น Clock Run-in และ Framing Code ขึ้นก่อน จากนั้นจึงจะนำข้อมูลจากหน่วยความจำมาทำการเลื่อนหรือสไลซ์ออกไปด้วยความถี่นาฬิกา 6.9375 เมกกะเฮิรตซ์ อีกจำนวน 42 ไบต์ในหนึ่งเส้นกวาด

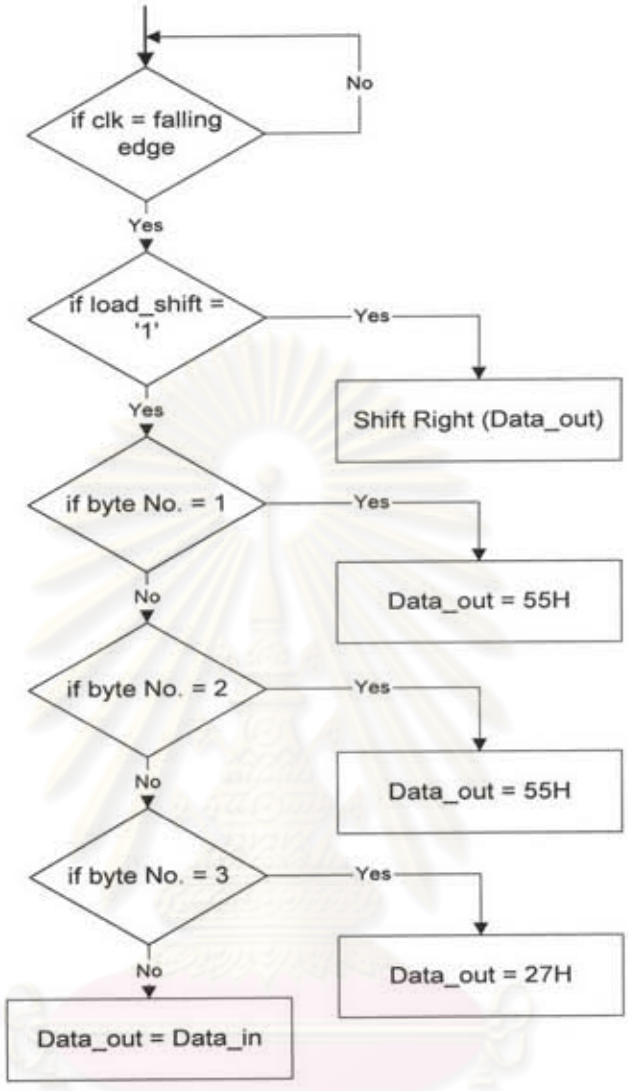
แนวทางในการออกแบบ PISO

วงจรส่วนนี้จะเป็นการออกแบบวงจรโหลดข้อมูลแบบขนานแล้วทำการชิฟต์ข้อมูลออกไปแบบอนุกรม โดยจะเพิ่มวงจรที่สร้างสัญญาณ Clock Run-in และ Framing Code เข้าไปในวงจรส่วนนี้ด้วย สามารถเขียนเป็นแผนภาพได้ดังรูปที่ 4.10



รูปที่ 4.10 แผนภาพเวลาของ PISO

จากแผนภาพเวลาสามารถเขียนเป็นแผนภาพการออกแบบภาษาวีเอชดีแอลได้ดังรูปที่ 4.11



รูปที่ 4.11 แผนภาพการออกแบบ PISO

4. Time_Base เป็นบล็อกที่ใช้ในการสร้างสัญญาณและทำหน้าที่ต่าง ๆ ดังนี้

1. สร้างสัญญาณนาฬิกาความถี่ 6.9375 เมกกะเฮิรตซ์ ในช่วงของเส้นกวาดหรือเส้นสแกนที่ทำการเลือกจาก SW_LINE เพื่อใช้เป็นสัญญาณในการชิฟต์ข้อมูลสำหรับบล็อก PISO
2. สร้างสัญญาณโหลดและเพิ่ม ที่ส่งให้กับบล็อก ADDRESS เพื่อนำไปใช้โหลดข้อมูลจากหน่วยความจำในส่วนที่เป็นแอดเดรสเริ่มต้นของหน้าข้อมูล เพื่อส่งค่าแอดเดรสออกไปทำการอ่านข้อมูลเทเลเท็กซ์
3. สร้างสัญญาณที่ใช้เป็นสัญญาณอินเตอร์รัพต์สำหรับชิปไมโครคอนโทรลเลอร์

4. สร้างสัญญาณที่ใช้เปลี่ยนการทำงานระหว่างหน่วยความจำรอมและแรม เพราะในขณะที่ทำการส่งข้อมูลออกไป ถ้าทำการสับสวิทช์เพื่อเปลี่ยนการทำงานขณะที่ข้อมูลยังส่งไปไม่หมดหน้า ก็จะยังไม่เปลี่ยนการทำจนกว่าจะทำการส่งข้อมูลจนหมดหน้า จึงจะสามารถเปลี่ยนการทำงานกันได้

จากการออกแบบวงจรส่วนนี้จะเป็นวงจรที่ใหญ่มาก ซึ่งจะประกอบด้วยบล็อกย่อย ๆ มาต่อเป็นวงจร ดังที่แสดงไว้ในภาคผนวก คือ บล็อก T_SLICE, บล็อก ROM_RAM, บล็อก T_MULT และบล็อก T_CNTLIN

บล็อก T_SLICE จะรับอินพุตจาก SW_LINE เพื่อนำไปสร้างวงจรในการนับและเลือกเส้น สแกนที่จะแทรกข้อมูลลงไป เช่น ถ้าเลือกสวิตช์ตัวที่ 1 เปิด ก็จะทำการแทรกข้อมูลลงในเส้นสแกนที่ 7, ถ้าเลือกสวิตช์ตัวที่ 2,3 และ 6 ก็จะทำการแทรกข้อมูลเทเลเท็กซ์ลงในเส้นสแกนที่ 8, 9และ 12 เป็นต้น นอกจากนี้ก็จะนำสัญญาณความถี่ 6.9375 เมกกะเฮิรตซ์ มาสร้างสัญญาณนาฬิกาสำหรับสไลซ์ข้อมูลแล้วยังทำเป็นวงจรนับใช้หน่วงเวลาในการสร้างสัญญาณให้ถูกต้องตามรูปที่ 2.5

บล็อก ROM_RAM จะรับอินพุตจาก ROM_RAM Switch เพื่อสร้างสัญญาณเปิดการทำงานให้หน่วยความจำรอมหรือแรม

บล็อก T_MULT เป็นบล็อกที่กำหนดจังหวะที่จะทำการโหลดข้อมูลหรือจังหวะที่จะทำการชิฟต์ข้อมูลแบบอนุกรม

บล็อก T_CNTLIN ใช้สร้างสัญญาณที่บอกให้รู้ว่าทำการส่งข้อมูลครบ 1 หน้า โดยทำการนับจำนวนเส้นสแกนหรือจำนวนแถวที่ทำการส่ง

5. Address เป็นบล็อกที่จัดการเกี่ยวกับการโหลดและการเพิ่มของแอดเดรส ในการออกแบบตอนแรกวงจรมีขนาดใหญ่มากจนไม่สามารถสร้างเป็นไลบรารีของอุปกรณ์ได้ เนื่องจากซอฟต์แวร์มีความสามารถจำกัด จึงทำให้ต้องแบ่งเป็นบล็อกย่อย ๆ คือ บล็อก T_ADDRS2, T_ADDRS3, T_ADDRS4 และ T_LDADDRแล้วทำการต่อวงจรแต่ละบล็อกเข้าด้วยกัน

บล็อก T_ADDRS2 เป็นบล็อกที่จะทำการส่งค่า F0000H ออกมา และทำการเก็บค่าของแอดเดรสหลังจากที่ทำการเพิ่มไปแอดเดรสไป 3 ตำแหน่ง และเนื่องจากโครงสร้างของฟลิปฟลอปภายในชิป FPGA ที่เลือกใช้ไม่สามารถทำการเซตลอจิกให้เป็น "1" ได้

บล็อก T_ADDRS3 นี้จะกำหนดจังหวะในการอ่านข้อมูลจำนวน 3 ไบต์จากหน่วยความจำ เพื่อเอามาเป็นค่าแอดเดรส

บล็อก T_ADDRS4 จะทำการเลือกแอดเดรสระหว่างแอดเดรสที่อ่านเข้ามากับแอดเดรสที่ทำการเก็บค่าแอดเดรสเริ่มต้นของหน้าข้อมูล

บล็อก T_LDADDR เป็นบล็อกที่จะทำการเพิ่มค่าของแอดเดรสจากแอดเดรสเริ่มต้นของหน้าข้อมูลส่งออกไปเพื่อทำการอ่านข้อมูลทเลเท็กซต์ทั้ง 1,050 ไบต์ดังที่กล่าวมาแล้ว

6. Latch เป็นบล็อกที่ทำการต่อกับพอร์ตและขา ALE ของไมโครคอนโทรลเลอร์ MCS-51 เพื่อทำการแลตช์ค่าพอร์ตขณะที่ทำการเขียนข้อมูลไปยังหน่วยความจำแรม

7. Sel_Address เป็นบล็อกที่จะทำการสวิตช์ค่าแอดเดรสระหว่างแอดเดรสที่ออกมาจากบล็อก ADDRESS กับแอดเดรสที่แลตช์ค่าจากพอร์ตของไมโครคอนโทรลเลอร์ เพื่อต่อออกไปเป็นเอาต์พุตภายนอกชิป โดยในช่วงที่อยู่ระหว่างเส้นกวาดที่ 7(320) ถึงเส้นที่ 22(335) จะให้ค่าแอดเดรสจากบล็อก ADDRESS ส่งออกไปเป็นเอาต์พุต แต่ถ้านอกเหนือจากช่วงเวลาดังกล่าวก็ให้ค่าที่แลตช์ค่าพอร์ตของชิปไมโครคอนโทรลเลอร์ส่งออกไปเป็นเอาต์พุตแทน

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย