

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

Experiment Adv DSD-04: Register and Time Delay

วัตถุประสงค์:

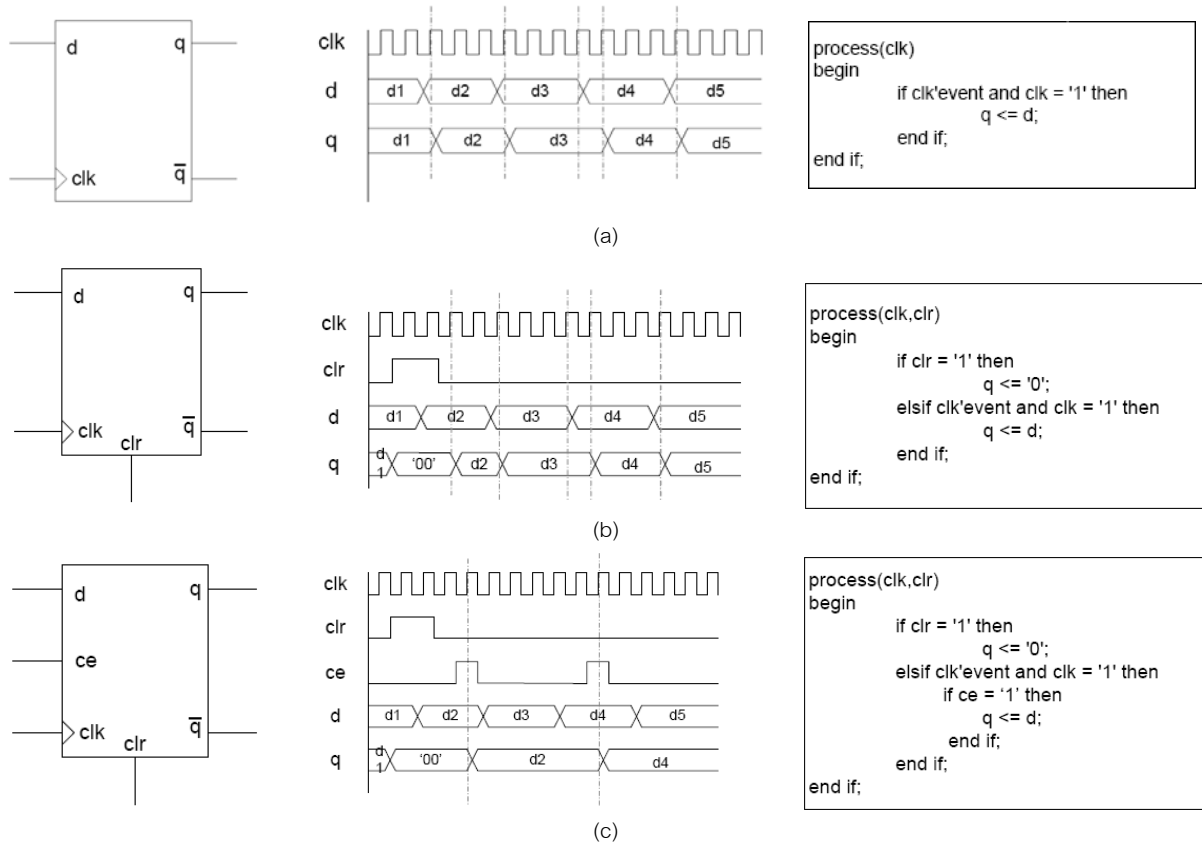
1. เข้าใจพฤติกรรมของวงจรรีจิสเตอร์
2. สามารถเขียน VHDL Code เพื่อสร้างวงจรรีจิสเตอร์ได้
3. เข้าใจความแตกต่างระหว่าง Synchronous Reset และ Asynchronous Reset

บทนำ

ในการออกแบบวงจรดิจิทัล นอกจากวงจร Combinational ที่เราได้ทำการทดลองออกแบบมาแล้ว วงจรที่สำคัญมากวงจรมองหนึ่งคือ วงจรประเภท Storage Element หรือวงจรที่สามารถเก็บข้อมูลไว้ในระยะเวลาหนึ่งที่ต้องการ เช่น รีจิสเตอร์, ฟลิปฟลอป, แลทช์

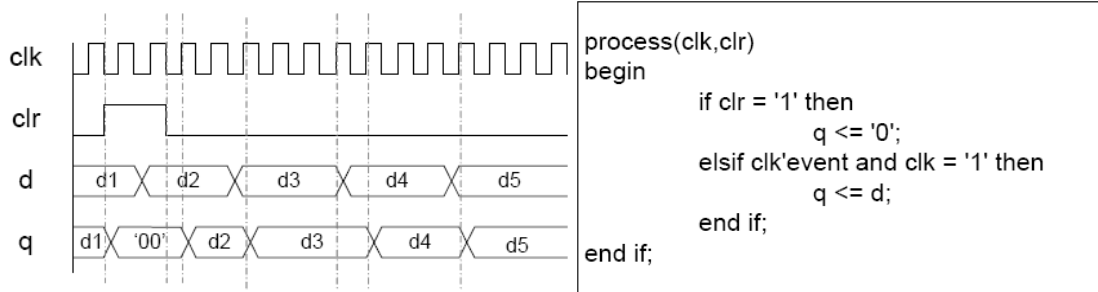
Register

วงจรรีจิสเตอร์ คือวงจรที่สัญญาณเอาต์พุต จะเปลี่ยนแปลงตามสัญญาณอินพุต เมื่อสัญญาณนาฬิกา (clk) มีการเปลี่ยนแปลงที่ขอบขึ้นหรือลง แล้วแต่ประเภทของรีจิสเตอร์ เป็นพฤติกรรมเดียวกับ D Flip-Flop ในที่นี้ใช้คำว่ารีจิสเตอร์

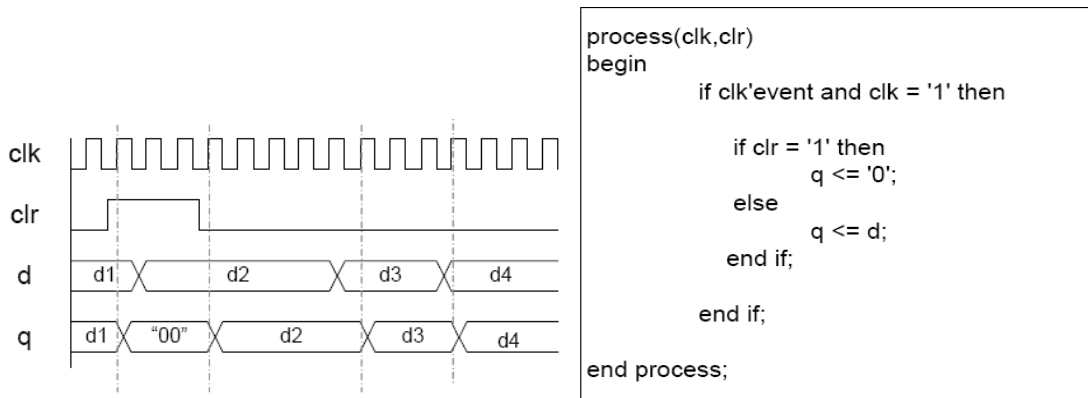


รูปที่ 4.1

รูปที่ 4.1 แสดงลักษณะของรีจิสเตอร์, ไทม์มิงไดอะแกรมและ VHDL code แบบต่างๆ รูป a นั้น จะมีเอาต์พุตเปลี่ยนแปลงทุกครั้งที่ขอบขาขึ้นของสัญญาณนาฬิกา ส่วน b จะมีการรีเซ็ตโดยสัญญาณ clr ส่วนรูป c จะมีสัญญาณ ce (clock enable) เป็นส่วนที่กำหนดตำแหน่งเวลาที่จะให้เอาต์พุตเปลี่ยนแปลงตามอินพุต โดยการทำงานทั้งหมดขึ้นกับตำแหน่งขอบสัญญาณนาฬิกา



(a)



(b)

รูปที่ 4.2

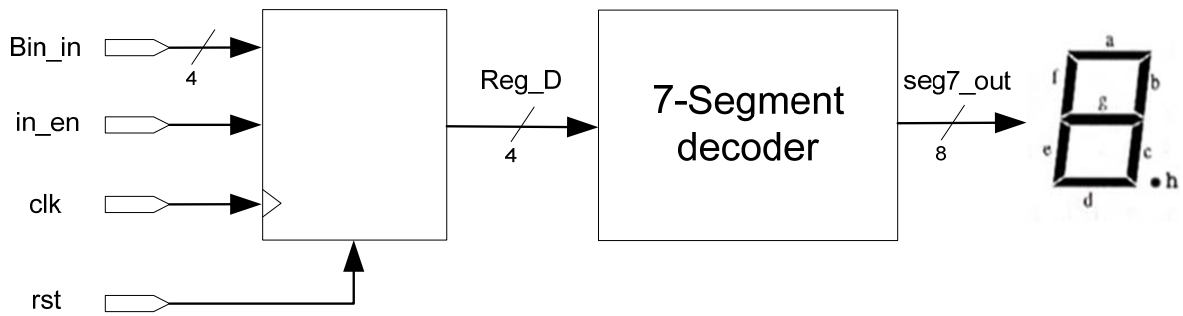
รูป a เป็นการรีเซ็ตแบบ Asynchronous reset คือ การเขียนรีเซ็ตขึ้นกับสัญญาณรีเซ็ต โดยไม่ขึ้นกับขอบสัญญาณของสัญญาณนาฬิกา

รูป b เป็นการรีเซ็ตแบบ Synchronous reset คือ การเขียนรีเซ็ตโดยจะต้องขึ้นกับเงื่อนไขขอบสัญญาณนาฬิกา

ขั้นตอนการทดลอง

วงจรรีจิสเตอร์ 4 บิต

1. ออกแบบวงจรรีจิสเตอร์ขนาด 4 บิต เพื่อรับข้อมูลไบนารีขนาด 4 บิต แล้วนำไปแสดงผลที่ 7-Segment โดยมีไดอะแกรมดังรูปที่ 4.3



รูปที่ 4.3

การทำงานของวงจร คือ สัญญาณ Reg_D[3:0] จะมีค่าเท่ากับ Bin_in[3:0] เมื่อตำแหน่งขอบขาขึ้นของสัญญาณ clk และ in_en อยู่ในสถานะ high และค่าของ Reg_D[3:0] จะถูกรีเซ็ตเมื่อ rst อยู่ในสถานะ high สัญญาณ Reg_D[3:0] จะถูกนำไปถอดรหัสให้เป็นข้อมูลสำหรับขับ 7-Segment display

2. การทดลองครั้งนี้สามารถใช้ module package ของ 7-segment จากการทดลองครั้งที่แล้วได้ ให้สร้าง VHDL module โดยอ้างอิงจากตัวอย่าง

```

Library .....;
use .....;

entity seg7_decoder is
    port ( ..... : .....;
          ..... : .....;
          ..... : .....;
          bin_in: std_logic_vector(3 downto 0;
          seg7_out: std_logic_vector(7 downto 0););
end .....;

architecture ..... of ..... is
    signal reg_D      :std_logic_vector(3 downto 0);
    signal seg7_tmp   :std_logic_vector(7 downto 0);
begin
    -- process register
    register: process .....
    begin
        if .....
            ..... <= .....
        elsif .....
            if .....
                ..... <= .....
            end if;
        end process register;

    -- process 7-segment decoder
    seg7_decoder: process .....
    begin
        -- decode 7-segment pattern to signal seg7_tmp
        .....
        .....
    end process .....;

    -- seg7_tmp to output
    .....
end .....;
    
```

หมายเหตุ: การกำหนด เหตุการณ์ของสัญญาณนาฬิกาสามารถดูได้จาก Language Templates ในหมวดการใช้งาน Process

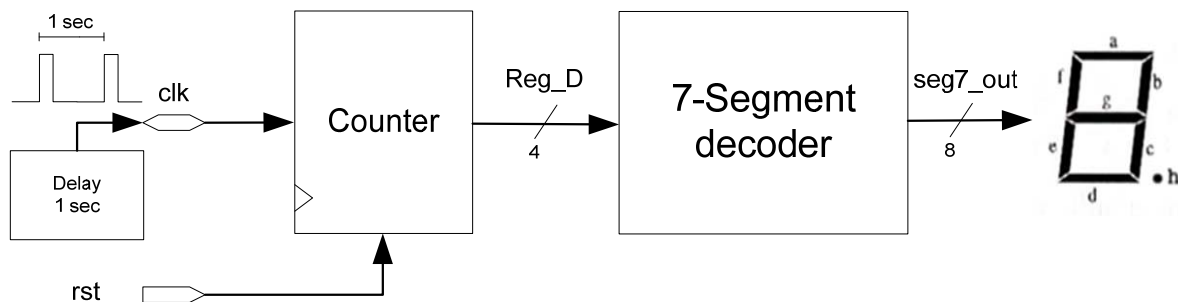
3. Simulate วงจรด้วย VHDL Test Bench โดยกำหนดกรณีทดสอบให้ครบทุกกรณี
4. Synthesis, Implementation โดยให้ทำการระบุตำแหน่ง อินพุต/เอาต์พุต ด้วยตนเอง โดยกำหนดดังนี้

Port	Device
Bin_in[3:0]	Dip Switch
In_en	Push Button
rst	Push Button
clk	Push Button
seg7_out	7-Segment

5. ดาวนีโหลดลงบอร์ดทดลอง แล้วทำการทดสอบ สังเกตผลที่เกิดขึ้น

Counter with Delay time

1. ให้ทำการออกแบบวงจร Counter นับวนรอบเลข 0-9 แล้วแสดงผลที่ 7-segment โดยในการนับเลข จะมีการเพิ่มค่าทุกๆขาขึ้นของสัญญาณ clk ทดสอบด้วย Simulator แล้วดาวน์โหลดบนบอร์ดทดลอง



2. พัฒนาให้มีส่วนสำหรับหน่วงเวลา 1 วินาทีโดยรับอินพุตจาก Oscillator บนบอร์ดทดลอง โดยทุกๆ 1 วินาทีจะมีเอาต์พุตเป็นสัญญาณ high pulse
3. นำเอาต์พุตของส่วนหน่วงเวลา 1 วินาทีไปใช้แทนสัญญาณ clk ของ Counter ทดสอบด้วย Simulator แล้วดาวน์โหลดบนบอร์ดทดลอง