จุฬาลงกรณ์มหาวิทยาลัย	ชื่อ_____
คณะวิศวกรรมศาสตร์	เลขประจำตัว_____
ภาควิชาวิศวกรรมคอมพิวเตอร์	หมายเลขเครื่อง_____
2110-361 HARDWARE SYN LAB.	วันที่_____

3. Stack Machine

วัตถุประสงค์

1. ศึกษาการใช้ Memory ใน Spartan-3

การปฏิบัติการ

ออกแบบและสร้างวงจรที่ทำงานแบบ Stack (LIFO) โดยใช้ Push Button Switches 2 อัน เพื่อ PUSH และ POP เมื่อผู้ใช้กดปุ่ม PUSH วงจรตะนำค่าจาก Slider Switches 8 อันมาเก็บลงใน Stack และเมื่อผู้ใช้กดปุ่ม POP วงจรจะนำค่า (8 bits) จาก Top ของ Stack มาแสดง เป็น Hex 2 digits ที่ 7 Segments 2 อันทางด้านซ้ายมือ

ส่วน 7 Segments อีก 2 อันทางด้านขวามือใช้แสดงจำวนของข้อมูลที่เก็บอยู่ใน Stack เป็นเลข Hex 2 digits และ Stack สามารถเก็บข้อมูลได้ 256 ข้อมูล

ควรใช้ Push button 1 อันเพื่อใช้กับสัญญาณ reset

ควรออกแบบและทดสอบด้วยการ simulate ก่อนจะ program ลง board

ข้อแนะนำ

ในอุปกรณ์ FPGA ของ Xilinx รุ่น Spartan-3 มี RAM สร้างไว้ภายในโดยจะแบ่งเป็น 2 ชนิด คือ

- Block RAMs
- Distributed RAMs

Block RAMs หรือ bRAMs เป็น memory device ที่สร้างไว้ใน FPGA แยกต่างห่างจาก CLB (Configurable Logic Block) หรือส่วนที่ใช้ Synthesis วงจรทั่วไป (gates, flip-flops,…) โดยจะจัดเรียงเป็น columns ของวงจร RAM เหมาะสำหรับงานที่ต้องการหน่วยความจำขนาดใหญ่ภายใน chip โปรแกรม Synthesis ของ Webpack (XST) จะนำหน่วยความจำเหล่านี้มาต่อเป็น RAM ขนาดต่างๆกันได้ ถ้าส่วน synthesis สามารถ infer ได้ว่า Verilog code ที่เขียนขึ้นต้องการใช้ RAM ซึ่งการใช้ template จะทำให้การ infer ง่ายขึ้น

ตัวอย่าง ไปที่ Edit→Language Templates→Verilog→Synthesis Constructs→Code Examples→RAM→BlockRAM→Single Port→Write First Mode แล้วเลือก Edit→ Use in File จะได้ template ดังนี้

```
parameter RAM_WIDTH = <ram_width>;
parameter RAM_ADDR_BITS = <ram_addr_bits>;

reg [RAM_WIDTH-1:0] <ram_name> [(2**RAM_ADDR_BITS)- 1:0];
reg [RAM_WIDTH-1:0] <output_data>;

<reg_or_wire> [RAM_ADDR_BITS-1:0] <address>;
<reg_or_wire> [RAM_WIDTH-1:0] <input_data>;

always @(posedge <clock>)
  if (<ram_enable>) begin
    if (<write_enable>) begin
      <ram_name>[<address>] <= <input_data>;
      <output_data> <= <input_data>;
    end
    else
      <output_data> <= <ram_name>[<address>];
  end
```

Block RAM จะแยกระหว่าง input data port กับ output data port (ไม่ต้องทำเป็น bidirectional)

<ram_width> คือ จำนวน bit ของข้อมูลที่จะเก็บใน RAM (หรือขนาดของ data)

<ram_addr_bits> คือ จำนวน bit ของ address ของ RAM (กำหนดขนาดของ “ความจุ” ของ RAM)

<ram_name> คือ “ชื่อ” หรือ identifier ที่ใช้กำหนด RAM

<ram_enable> เทียบได้กับสัญญาณ Chip Select

Distributed RAMs เป็น memory device ที่สร้างไว้ใน CLB มีการจัดเรียงในระดับ bit มีจำนวนน้อยกว่า block RAM มีความเร็วสูงกว่า แต่ถ้าใช้เป็นจำนวนมากจะมี overhead สำหรับวงจรในการกำหนด address ของ RAM จึงเหมาะกับงานที่ใช้ memory ขนาดเล็ก เช่นเดียวกับ block RAM ควรใช้ template ในการเขียน code เพื่อ XST จะได้ infer เป็น Distributed RAM ได้ถูกต้อง

ถึงแม้ว่า RAM ทั้งสองแบบจะเรียกว่า RAM แต่อาจใช้เป็น ROM หรือ Register File ได้
RAM ทั้งสองแบบทำงานแบบ Write Synchronous คือ write พร้อมกับ positive edge ของ clock

ในการปฏิบัติการนี้ให้ใช้ Block RAM