

**การทดลองที่ 10** วงจรจำลองการทำงานของ CPU อย่างง่ายโดยใช้วิธี Schematic บนบอร์ด FPGA

### วัตถุประสงค์

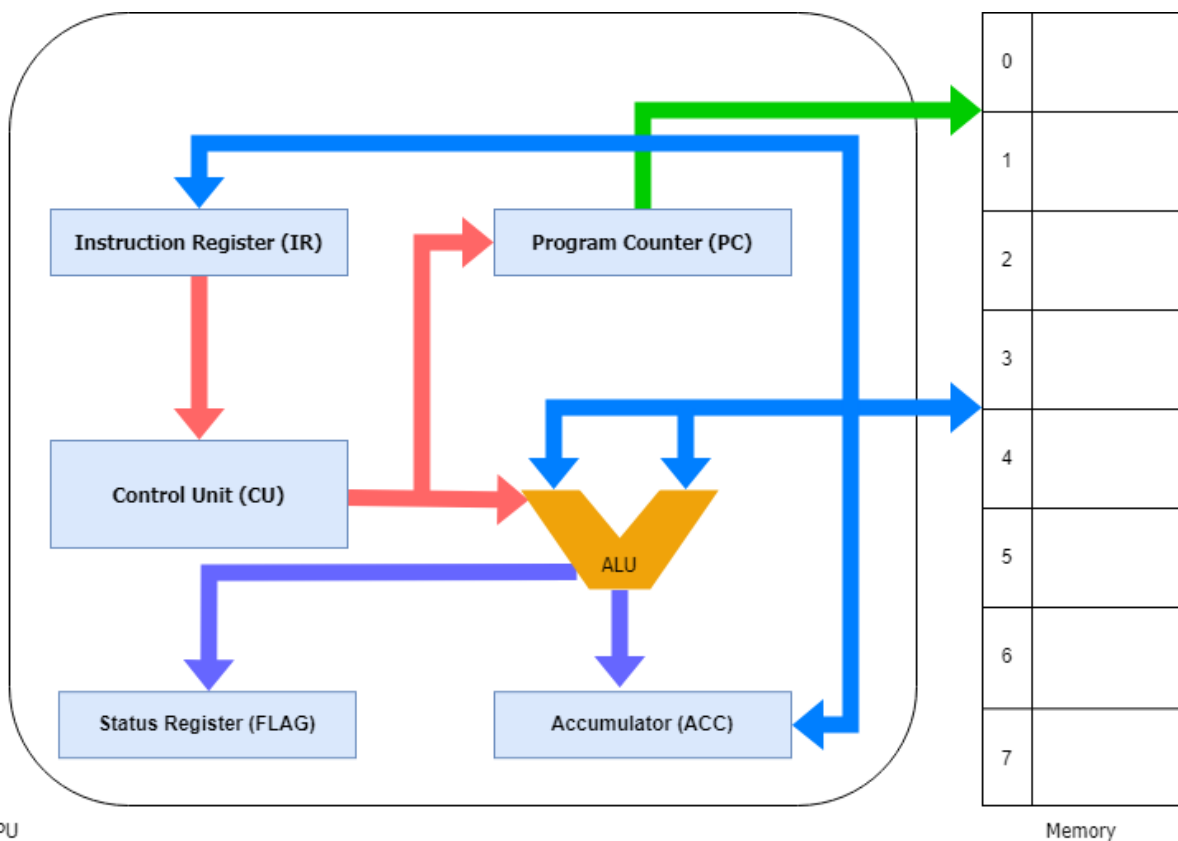
1. เพื่อเสริมสร้างความเข้าใจของนักศึกษาเกี่ยวกับกลไกการประมวลผลของ CPU
2. เพื่อให้นักศึกษาฝึกการใช้งาน FPGA

หมายเหตุ ให้อ่านเอกสารการทดลองและเอกสารประกอบให้ครบก่อนเริ่มทำการทดลอง

### บทนำ

หน่วยประมวลผลกลาง (Central Processing Unit: CPU) เป็นองค์ประกอบสำคัญของระบบคอมพิวเตอร์ ทำหน้าที่ประมวลผลข้อมูลตามชุดคำสั่งที่กำหนด

องค์ประกอบหลักของ CPU ประกอบด้วย 6 ส่วนหลักที่ทำงานประสานกันเพื่อประมวลผลคำสั่งและข้อมูล



### 1. รีจิสเตอร์คำสั่ง (Instruction Register: IR)

- ทำหน้าที่จัดเก็บคำสั่งที่กำลังประมวลผลในขณะนั้น
- เป็นจุดเชื่อมต่อระหว่างหน่วยความจำและวงจรการประมวลผลของ CPU

### 2. ตัวนับโปรแกรม (Program Counter: PC)

- บันทึกที่อยู่ของคำสั่งถัดไปที่จะถูกเรียกมาประมวลผล
- มีการปรับค่าโดยอัตโนมัติหลังจากการเรียกคำสั่งแต่ละครั้ง หรือถูกปรับโดยคำสั่งกระโดด (Jump) และการเรียกโปรแกรมย่อย (Subroutine)

### 3. หน่วยควบคุม (Control Unit: CU)

- ทำหน้าที่ควบคุมการทำงานของส่วนประกอบอื่น ๆ ใน CPU
- ถอดรหัสคำสั่งจาก IR และส่งสัญญาณควบคุมไปยังส่วนต่าง ๆ ของ CPU
- ประสานงานระหว่างหน่วยความจำและหน่วยคำนวณและตรรกะ

### 4. รีจิสเตอร์สถานะ (Status Register หรือ FLAG)

- บันทึกสถานะของการประมวลผลล่าสุด เช่น ผลลัพธ์เป็นศูนย์, มีการทด, มีเครื่องหมายลบ
- ใช้ในการตัดสินใจสำหรับคำสั่งแบบมีเงื่อนไข และการจัดการกับข้อยกเว้น (Exceptions)

### 5. ตัวสะสม (Accumulator: ACC)

- เป็นรีจิสเตอร์พิเศษที่ใช้เก็บผลลัพธ์ชั่วคราวจากการคำนวณของหน่วยคำนวณและตรรกะ
- มักเป็นตัวตั้งหรือตัวรับในการดำเนินการทางคณิตศาสตร์หรือตรรกะ

### 6. หน่วยคำนวณและตรรกะ (Arithmetic and Logic Unit: ALU)

- ทำการคำนวณทางคณิตศาสตร์ (บวก, ลบ, คูณ, หาร)
- ดำเนินการทางตรรกะ (AND, OR, NOT, XOR)
- ส่งผลลัพธ์ไปยัง ACC และปรับค่า FLAG ตามผลการคำนวณ

**การทำงานของ CPU ร่วมกับหน่วยความจำ (Memory)** การทำงานของ CPU ร่วมกับ Memory เป็นกระบวนการต่อเนื่องที่เรียกว่า วงจรการเรียกและปฏิบัติตามคำสั่ง (Fetch-Execute Cycle) ประกอบด้วยขั้นตอนดังนี้

1. **การเรียกคำสั่ง (Fetch)** CPU ใช้ค่าใน PC เพื่อระบุตำแหน่งของคำสั่งใน Memory และเรียกคำสั่งมาเก็บใน IR
2. **การถอดรหัสคำสั่ง (Decode)** CU วิเคราะห์คำสั่งใน IR เพื่อกำหนดการดำเนินการที่จำเป็น
3. **การเรียกข้อมูล (Fetch Operands)** หากคำสั่งต้องการข้อมูลเพิ่มเติม CPU จะเรียกข้อมูลจาก Memory หรือรีจิสเตอร์อื่น ๆ
4. **การปฏิบัติตามคำสั่ง (Execute)** ALU ดำเนินการตามคำสั่ง โดยใช้ข้อมูลจาก ACC หรือรีจิสเตอร์อื่น ๆ
5. **การจัดเก็บผลลัพธ์ (Store Result)** ผลลัพธ์จะถูกเขียนกลับไปยัง Memory หรือรีจิสเตอร์ตามที่ระบุในคำสั่ง

การสื่อสารระหว่าง CPU และ RAM ดำเนินการผ่านบัสระบบ (System Bus) ซึ่งประกอบด้วย

- บัสข้อมูล (Data Bus): สำหรับการส่งข้อมูลระหว่าง CPU และ RAM
- บัสที่อยู่ (Address Bus): ใช้ระบุตำแหน่งของข้อมูลใน RAM
- บัสควบคุม (Control Bus): สำหรับส่งสัญญาณควบคุมระหว่าง CPU และอุปกรณ์อื่น ๆ

**คำสั่ง (Instructions) และการประมวลผล** คำสั่งคอมพิวเตอร์เป็นชุดของรหัสที่บอก CPU ว่าต้องดำเนินการอย่างไร โดยทั่วไปคำสั่งประกอบด้วยสองส่วนหลัก

1. **รหัสปฏิบัติการ (Operation Code หรือ Opcode)** ระบุการดำเนินการที่ต้องทำ เช่น บวก, ลบ, หาร, โดด
2. **ตัวถูกดำเนินการ (Operand)** ข้อมูลหรือตำแหน่งของข้อมูลที่จะใช้ในการดำเนินการ

คำสั่งถูกจัดเก็บใน Memory และถูกเรียกมาประมวลผลทีละคำสั่งตามลำดับที่กำหนดโดย PC เว้นแต่จะมีคำสั่งกระโดดหรือเรียกโปรแกรมย่อย

ประเภทของคำสั่งที่พบบ่อยในการทำงานของ CPU ได้แก่

- คำสั่งการโอนข้อมูล (Data Transfer Instructions)
- คำสั่งทางคณิตศาสตร์และตรรกะ (Arithmetic and Logical Instructions)
- คำสั่งควบคุมการทำงาน (Control Flow Instructions)
- คำสั่งการจัดการหน่วยความจำ (Memory Management Instructions)
- คำสั่งการป้อนข้อมูลและแสดงผล (Input/Output Instructions)

## การทดลอง

1. ให้นักศึกษานำเอกสารใบตรวจการทดลองให้อาจารย์ผู้ควบคุมการทดลองเซ็นรับรองเอกสารก่อนเริ่มทำการทดลองภายใน 1 ชั่วโมง
  2. ออกแบบวงจรจำลองการทำงานของ CPU ด้วยมีส่วนประกอบ 3 ส่วนคือ ALU ACC และ FLAG
    - 2.1. มี ACC 8 bit โดยเมื่อเปิดเครื่อง กำหนดให้ ACC=0
    - 2.2. มี ALU 1 ตัว ที่สามารถ บวก (+) ลบ(-) XOR(^) Logical Shift Left(<<) AND OR และMOV โดยกำหนดให้ ACC Operand เสมอ
    - 2.3. INPUT
      - 2.3.1. dip switch [2:0] สำหรับ Operation 3 bits
        - 000 คือ ไม่มีการดำเนินการใดๆ
        - 001 คือ +
        - 010 คือ -
        - 011 คือ ^
        - 100 คือ <<
        - 101 คือ AND
        - 110 คือ OR
        - 111 คือ MOV การกำหนดค่าของ ACC
      - 2.3.2. slide switch [7:0] สำหรับ ค่าคงที่ 8 bits
    - 2.4. OUTPUT
      - 2.4.1. 7segment [1:0] แสดง ACC ที่ได้หลังจากทำ Operation เสมอ ในกรณี Overflow หรือ Carry ให้ตัดส่วนเกินทิ้ง
      - 2.4.2. 7segment [3:2] แสดงค่าคงที่จาก slide switch [7:0]
      - 2.4.3. หลอดไฟ LED แสดง FLAG ที่ได้หลังจาก ALU ทำ Operation
        - L0 สำหรับ Zero Flag ผลลัพธ์เป็น 0
        - L1 สำหรับ Overflow Flag ผลลัพธ์เกินจำนวนbitsที่รองรับได้
        - L2 สำหรับ Carry/Borrow flag ผลลัพธ์มีการทศออกไป หรือยืมเข้ามา
    - 2.5. Operation จะกระทำกับ ACC และเก็บผลลัพธ์ไว้ที่ ACC เมื่อกดปุ่ม PB3 เปรียบเสมือน PB3 เป็น Clock
- Hint** 1 Clock = 1 Operation

### ตัวอย่าง

No	Operation	7segment [3:2] ค่าคงที่	7segment [1:0] ACC	Carry/Borrow flag L2	Overflow Flag L1	Zero Flag L0
1	100 0110 1001	69	00	0	0	1
2	101 1000 0000	80	00	0	0	1
3	000 0000 0000	00	00	0	0	1
4	111 1111 1111	FF	FF	0	0	0
5	011 1101 0100	D4	2B	0	0	0
6	101 1100 0110	C6	02	0	0	0
7	011 1000 0000	80	82	0	0	0
8	111 0000 0101	05	05	0	0	0

### ความหมาย

No	ACC	Operation	ค่าคงที่	ผลลัพธ์
1	00	<<	69	00
2	00	AND	80	00
3	00	ไม่มีการดำเนินการใดๆ	00	00
4	00	MOV	FF	FF
5	FF	^	D4	2B
6	2B	AND	C6	02
7	02	^	80	82
8	82	MOV	05	05

### การทดลองพิเศษ

ออกแบบวงจรจำลองการทำงานของ CPU ด้วยมีส่วนประกอบ 6 ส่วนคือ CU ALU IR PC FLAG และACC

1. มี Registers 2 ตัว คือ R0 8 bits , R1 8 bits โดยเมื่อเปิดเครื่อง กำหนดให้ R0=0 และR1=0
2. มี ALU 1 ตัว ที่สามารถ บวก (+) ลบ(-) XOR(^) Logical Shift Left(<<) AND OR และMOV
3. มี Instruction ที่เก็บได้ 8 Operations โดยมี Operation ละ 15 bits
4. INPUT
  - 4.1. dip switch[6:4] สำหรับเลือก INPUT Operation (Operation[14:12])
    - 000 คือ R0 Operate R1
    - 001 คือ R1 Operate R0
    - 010 คือ R0 Operate ค่าคงที่
    - 011 คือ ค่าคงที่ Operate R0
    - 100 คือ R1 Operate ค่าคงที่
    - 101 คือ ค่าคงที่ Operate R1
  - 4.2. dip switch[3] สำหรับเลือกว่าจะเก็บผลลัพธ์จากการ Operation ที่ใด (Operation[11])
    - 0 คือเก็บผลลัพธ์ ที่ R0
    - 1 คือเก็บผลลัพธ์ ที่ R1
  - 4.3. dip switch[2:0] สำหรับ Operation 3 bits (Operation[10:8])
    - 000 คือ ไม่มีการดำเนินการใดๆ
    - 001 คือ +
    - 010 คือ -
    - 011 คือ ^
    - 100 คือ <<
    - 101 คือ AND
    - 110 คือ OR
    - 111 คือ MOV การกำหนดค่าของ R ในที่กรณี นำ R0 และ R1 มาทำ Operation ให้ เลือก R ที่ไม่ได้ใช้เก็บผลลัพธ์ เป็นตัวกำหนดค่าของ R ที่ใช้เก็บผลลัพธ์
  - 4.4. slide switch[7:0] สำหรับ ค่าคงที่ 8 bits (Operation[7:0])
  - 4.5. ปุ่ม PB3 เพื่อ save Operation ลง Instruction จำนวน 8 Operations ถ้าเพิ่มมากกว่า 8 Operations ให้นำ Operation แรกสุดออก และ Operation ล่าสุดต่อท้าย (QUEUE)
  - 4.6. ปุ่ม PB4 เพื่อเริ่มนับค่า PC โดยมีความเร็วการนับ PC อยู่ 1 Hz เพื่อที่จะทำ Operations ต่าง ๆ ที่อยู่ใน Instruction แล้วหยุดที่ Operation สุดท้าย และกดอีกครั้งเพื่อเริ่มต้นการทำงานใหม่ PC = 0 โดยไม่รีเซ็ตค่า R0 และR1 (เริ่มต้น PC = 0 และOperationแรกจะอยู่ที่แอดเดรส 1 แอดเดรส 0 จะไม่ทำOperationใดๆ)

## 5. OUTPUT

- 5.1. 7segment [1:0] แสดง R0 ที่ได้หลังจากทำ Operation เสมอ ในกรณี Overflow หรือ Carry ให้ตัดส่วนเกินทิ้ง
  - 5.2. 7segment [3:2] แสดง R1 ที่ได้หลังจากทำ Operation เสมอ ในกรณี Overflow หรือ Carry ให้ตัดส่วนเกินทิ้ง
  - 5.3. หลอดไฟ LED แสดง FLAG ที่ได้หลังจาก ALU ทำ Operation
    - L0 สำหรับ Zero Flag ผลลัพธ์เป็น 0
    - L1 สำหรับ Overflow Flag ผลลัพธ์เกินจำนวนbitsที่รองรับได้
    - L2 สำหรับ Carry/Borrow flag ผลลัพธ์มีการทศออกไป หรือยืมเข้ามา
6. เมื่อPC = 0 จะไม่มีการแสดงผลใดๆ

**Hint** 1 Operation = 4 Clock

- 1.อ่าน Operation จาก Instruction ตาม PC
- 2.โหลดค่าที่ต้องการใช้
- 3.operate
- 4.บันทึกผลลัพธ์ และแสดงผล

## ตัวอย่าง

### Instruction

Address	Operation	ความหมาย
0	-	ไม่มีการดำเนินการใดๆ
1	000 0 110 1100 0001	$R0 = R0 \text{ OR } R1$
2	001 1 010 1101 1111	$R1 = R1 - R0$
3	000 0 111 0110 1001	$R0 = R1$
4	010 1 001 1010 0101	$R1 = R0 + A5$
5	011 1 100 0011 1100	$R1 = 3C \ll R0$
6	001 1 101 1001 0111	$R1 = R1 \text{ AND } R0$
7	101 1 111 1111 1111	$R1 = FF$
8	100 1 001 0000 0001	$R1 = R1 + 1$

### การทำงาน

PC	7segment [3:2] R1	7segment [1:0] R0	Carry/Borrow flag L2	Overflow Flag L1	Zero Flag L0
0	ไม่แสดงผล	ไม่แสดงผล	ไม่แสดงผล	ไม่แสดงผล	ไม่แสดงผล
1	00	00	0	0	1
2	00	00	0	0	1
3	00	00	0	0	1
4	A5	00	0	0	0
5	3C	00	0	0	0
6	00	00	0	0	1
7	FF	00	0	0	0
8	00	00	1	1	1

\*หมายเหตุ LED 1 หมายถึง ดิจิต 0 หมายถึง ดับ



ใบตรวจการทดลองที่ 10

วัน/เดือน/ปี \_\_\_\_\_ ☐ กลุ่ม 116 ☐ กลุ่ม 117 ☐ กลุ่ม 153

รหัสนักศึกษา \_\_\_\_\_ ชื่อ-นามสกุล \_\_\_\_\_

การตรวจการทดลอง

ชื่ออาจารย์ประจำวิชา \_\_\_\_\_

การทดลองข้อ 1 ลายเซ็นผู้ควบคุมการทดลอง \_\_\_\_\_ วัน/เดือน/ปี \_\_\_\_\_ เวลา \_\_\_\_\_

☐ ไม่หักส่งซ้ำ ☐ หักส่งซ้ำ 50%

การทดลองข้อ 2 ลายเซ็นผู้ควบคุมการทดลอง \_\_\_\_\_

การทดลองพิเศษลายเซ็นผู้ควบคุมการทดลอง \_\_\_\_\_

หมายเหตุ ไม่รับ ใบตรวจการทดลองที่มีร่องรอยการแก้ไข ชูด ลบ ชีดฆ่า เปลี่ยนแปลงทุกชนิด