

รีจิสเตอร์ A

เป็นรีจิสเตอร์หลักของระบบ การทำงานใดๆ ไม่ว่าจะเป็นการบวก ลบ หรือการส่งผลลัพธ์ออก แสดงผลใช้รีจิสเตอร์ A ทำงานเสมอ ลักษณะของรีจิสเตอร์ A ประกอบด้วย

D3 - D0 เป็นบัส 2 ทิศทาง

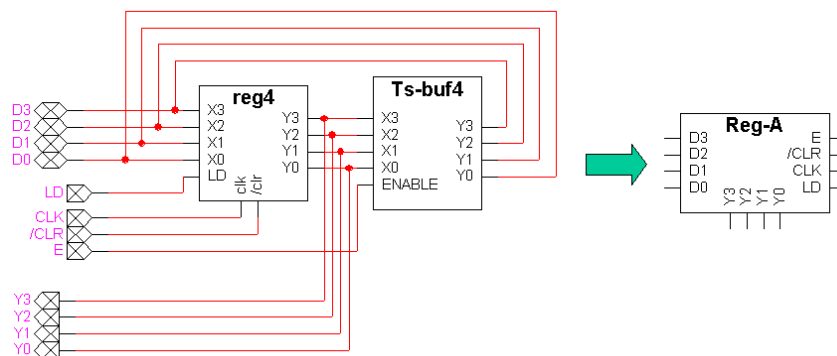
Y3 - Y0 เป็นบัสทิศทางเดียว

CLK เป็นสัญญาณนาฬิกา ใช้กำกับการทำงานของรีจิสเตอร์ เมื่อมีสัญญาณขอบขาขึ้นมา รีจิสเตอร์จึงจะทำงาน ตามการควบคุมของสัญญาณอื่นๆ

LD ใช้ควบคุมการรับข้อมูลจาก D3 - D0 แอคทีฟ 1

E ใช้ควบคุมการส่งข้อมูลที่เก็บอยู่ภายในรีจิสเตอร์ให้ออกมาปรากฏที่ D3-D0 แอคทีฟ 1

/CLR ใช้สำหรับทำให้ข้อมูลภายในรีจิสเตอร์เป็น 0 ทั้งหมด

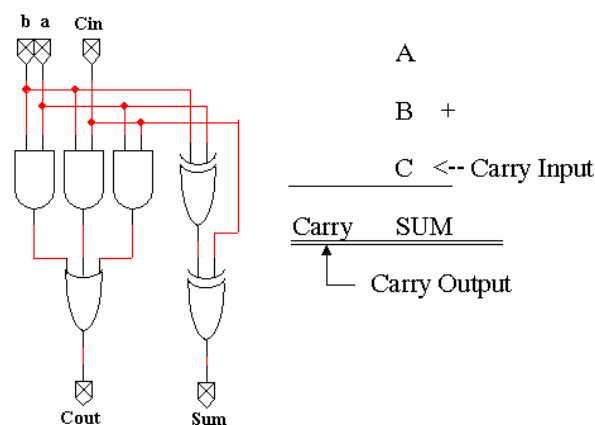


รูปที่ 2.8 บล็อกไดอะแกรมและสัญญาณลักษณะของรีจิสเตอร์ A

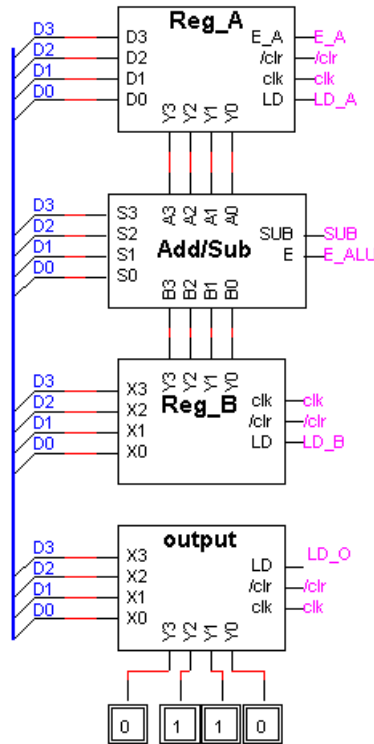
2.4 หน่วยประมวลผล (ALU)

หน่วยนี้ทำหน้าที่ในการประมวลผล สำหรับ SiCo ทำได้เพียงการบวกและลบเลขไบนารีขนาด 4 บิตเท่านั้น โครงสร้างประกอบขึ้นจากวงจร Full Adder จำนวน 4 ชุด

วงจรวก Full Adder



รูปที่ 2.9 โลจิกไดอะแกรมของ Full adder



รูปที่ 2.11 บล็อกไออะแกรมแสดงการต่อส่วนประกอบต่างเข้าด้วยกันโดยใช้บัส

ตารางที่ 2.1 แสดงค่าลอจิกที่ให้กับสัญญาณต่างๆเมื่อส่งข้อมูลจาก A ให้แก่ Output

E_A	LD_A	SUB	E_ALU	LD_B	LD_O	/CLR
1	0	0	0	0	1	1

หมายเหตุ ในขณะที่ต้องมีสัญญาณนาฬิกาอยู่ตลอดเวลา

2.6 Program Counter (PC)

PC เป็นวงจรรนับ ใช้สำหรับระบุตำแหน่งของคำสั่งที่จะนำไปทำงานในลำดับถัดไป เมื่อมีการนำคำสั่งออกจากหน่วยความจำแล้ว ค่าของ PC จะเพิ่มค่าไปที่ละ 1 โดยอัตโนมัติ หน้าที่และการทำงานของสัญญาณต่างๆมีดังนี้

Q3 - Q0 เป็นสัญญาณเอาต์พุต

/clr เป็นสัญญาณควบคุมเพื่อทำให้ค่า Q3 - Q0 เป็นลอจิก 0000 โดย

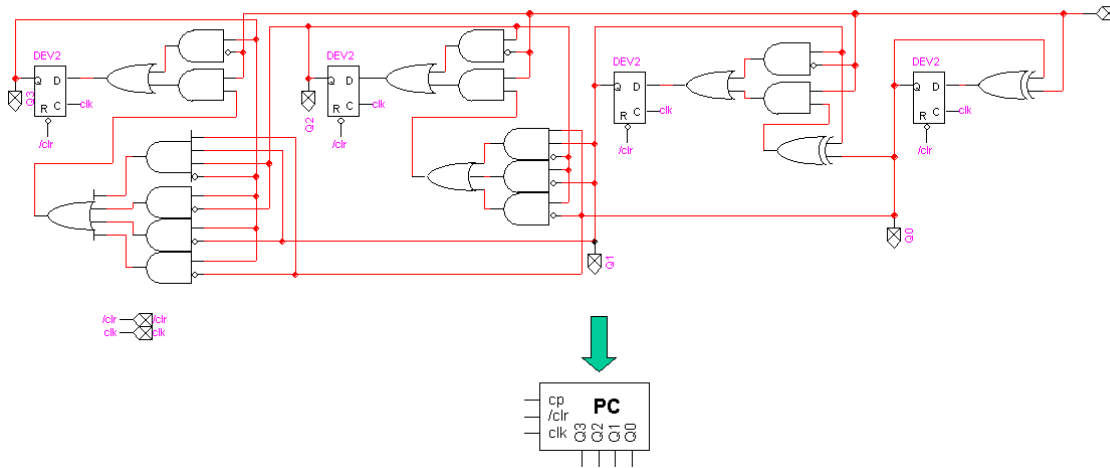
ถ้า /clr = 0 ทำให้ Q3-Q0 = 0000

ถ้า /clr = 1 การทำงานขึ้นอยู่กับ CP และ clk

CP เป็นสัญญาณควบคุม

ถ้า CP = 0 ค่านับคงเดิม

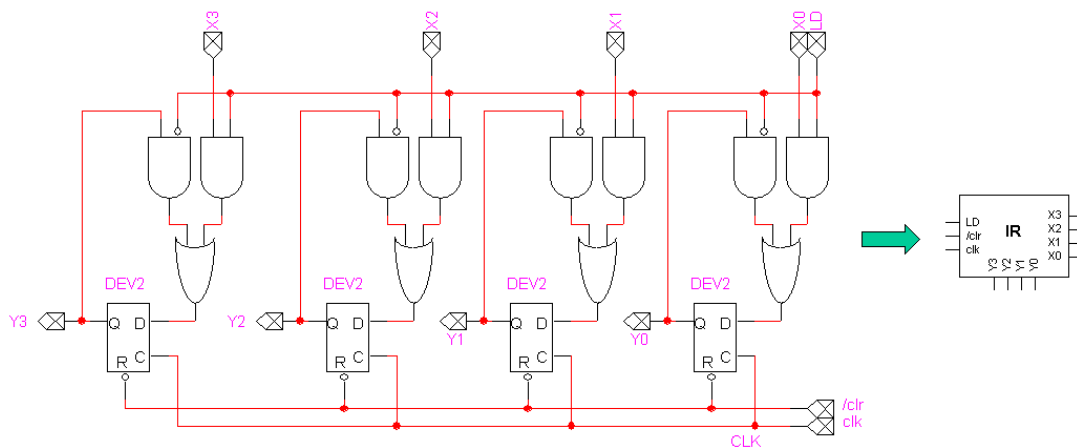
ถ้า CP = 1 นับขึ้น



รูปที่ 2.12 โลจิกไดอะแกรมและสัญลักษณ์ของ PC

2.7 Instruction Register(IR)

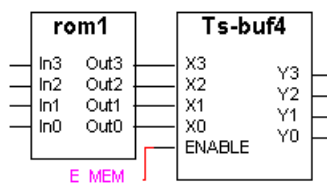
เป็นรีจิสเตอร์ที่ใช้เก็บคำสั่งระหว่างปฏิบัติงาน สร้างจากบัพเฟอร์รีจิสเตอร์ขนาด 4 บิต



รูปที่ 2.13 โลจิกไดอะแกรมและสัญลักษณ์ของ IR

2.8 หน่วยความจำ

ใช้สำหรับเก็บโปรแกรมที่ต้องการให้ SiCo หน่วยความจำนี้มีบัสข้อมูลขนาด 4 บิตและบัสแอดเดรสขนาด 4 บิต สามารถเก็บข้อมูลได้ 16 ตำแหน่ง ตำแหน่งละ 4 บิต เนื่องจากต้องต่อหน่วยความจำ



รูปที่ 2.14 โลจิกไดอะแกรมและสัญลักษณ์ของหน่วยความจำ

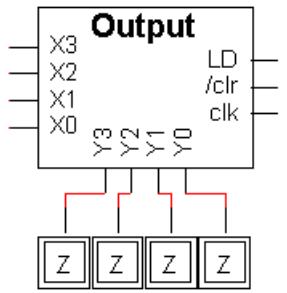
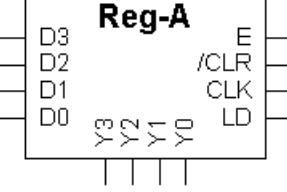
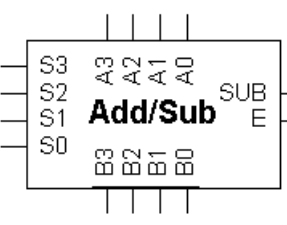
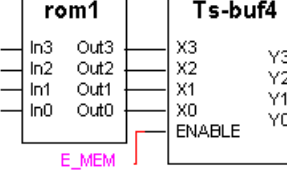
เข้ากับบัสภายใน ดังนั้นสัญญาณเอาต์พุตของหน่วยความจำต้องเป็นแบบ 3 สถานะ จึงใช้บัฟเฟอร์แบบ 3 สถานะมาประกอบ การทำงานเป็นดังนี้

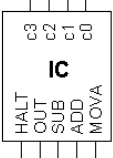
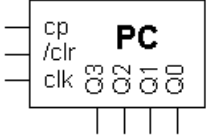
ถ้า E_MEM เป็นลอจิก 1 ข้อมูลจากหน่วยความจำ จะปรากฏที่ Y

ถ้า E_MEM เป็นลอจิก 0 สัญญาณ Y จะเป็น High Impedance

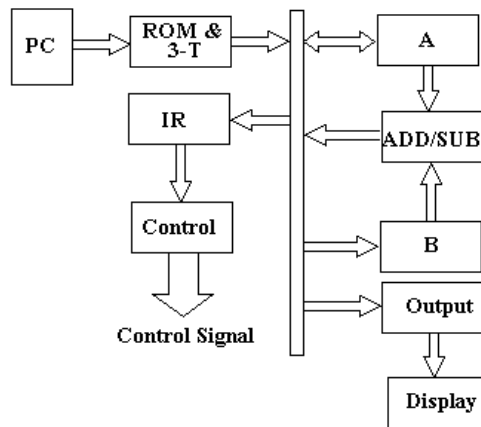
สำหรับข้อมูลในหน่วยความจำนั้นต้องถูกบันทึกไว้ก่อนที่จะนำมาใช้งาน

2.9 สรุปหน้าที่และการทำงานของหน่วยต่างๆของ SiCo

	<p>รีจิสเตอร์เอาต์พุต (Output Register) รีจิสเตอร์ B และ รีจิสเตอร์คำสั่ง (IR)</p> <p>อินพุต X0 - X3</p> <p>เอาต์พุต Y0 - Y3</p> <p>การทำงาน /clr ทำให้เอาต์พุตเป็น 0</p> <p>LD = 0 Y คงค่าเดิม</p> <p>LD = 1 Y = X</p>
	<p>รีจิสเตอร์ A</p> <p>อินพุต/เอาต์พุต D0 - D3 เป็นสัญญาณ 2 ทิศทาง แบบ 3 สถานะ</p> <p>เอาต์พุต Y0 - Y3</p> <p>การทำงาน /clr ทำให้เอาต์พุตเป็น 0</p> <p>LD = 0 Y คงค่าเดิม</p> <p>LD = 1 Y = X</p> <p>E = 0 D0 - D3 ทำหน้าที่เป็นเอาต์พุต</p> <p>E = 1 D0 - D3 ทำหน้าที่เป็นอินพุต ส่วนเอาต์พุตอยู่ในสถานะความต้านทานสูง</p>
	<p>อินพุต A0 - A3 B0 - B3</p> <p>เอาต์พุต S0 - S3 เป็นเอาต์พุตแบบ 3 สถานะ</p> <p>การทำงาน SUB = 0 Output = A + B</p> <p>SUB = 1 Output = A - B</p> <p>E = 0 ข้อมูลเอาต์พุตปรากฏที่ S0 - S3</p> <p>E = 1 S0 - S3 อยู่ในสถานะความต้านทานสูง</p>
	<p>หน่วยความจำ</p> <p>อินพุต In0 - In3</p> <p>เอาต์พุต Y0 - Y3 เป็นเอาต์พุตแบบ 3 สถานะ</p> <p>การทำงาน ENABLE = 0 ข้อมูลจาก ROM จะปรากฏที่ Y</p>

	<p>ถอดรหัสคำสั่ง (Instruction Decoder)</p> <p>อินพุต C0 - C3</p> <p>เอาต์พุต HALT, OUT, SUB, ADD, MOV A แอคทีฟ 1</p>
	<p>Program Counter (PC)</p> <p>เอาต์พุต Q0 - Q3</p> <p>การทำงาน /clr ทำให้เอาต์พุตเป็น 0</p> <p> CP = 0 คำนับคงเดิม</p> <p> CP = 1 นับขึ้น</p>

เมื่อนำหน่วยประกอบต่างๆรวมทั้งหน่วยควบคุมมาประกอบเข้าด้วยกันจะมีโครงสร้างดังรูปที่ 2.15



รูปที่ 2.15 บล็อกไดอะแกรมของ SiCo

สำหรับหน่วยควบคุม ซึ่งเป็นหน่วยใหญ่อีกหน่วยหนึ่งนั้น มีความสัมพันธ์กับคำสั่งที่ต้องการให้มีดั่งนั้นต้องกล่าวถึงคำสั่งเสียก่อน

2.10 คำสั่ง

เนื่องจากต้องการให้ SiCo นั้นง่ายต่อการทำความเข้าใจ จึงกำหนดให้มีคำสั่งเพียง 5 คำสั่งดังนี้

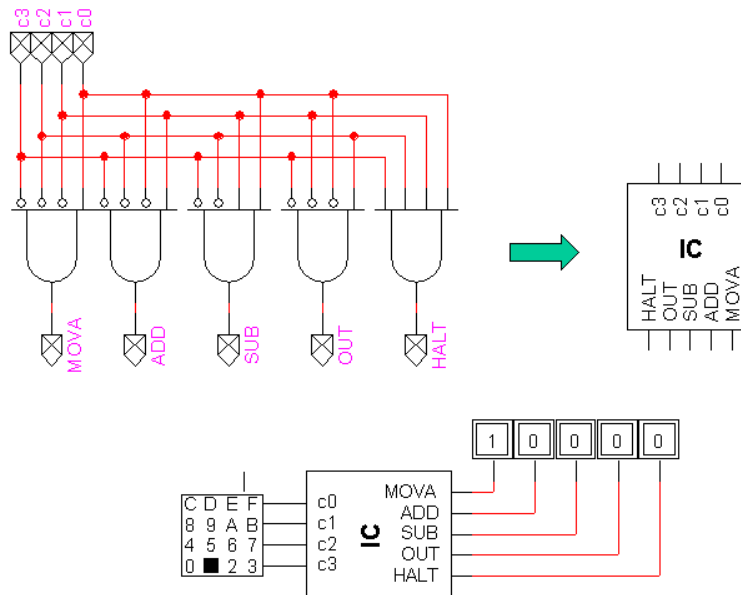
ตารางที่ 2.2 คำสั่งของ SiCo

รหัส	ความหมาย	สัญลักษณ์การทำงาน	รหัสช่วยจำ
1	เก็บข้อมูลในรีจิสเตอร์	$A \leftarrow n$	MOV A,#n
2	บวกข้อมูลในรีจิสเตอร์กับข้อมูลผลลัพธ์เก็บที่ รีจิสเตอร์	$A \leftarrow A + n$	ADD A,#n
3	ลบข้อมูลในรีจิสเตอร์ด้วยข้อมูลผลลัพธ์เก็บที่ รีจิสเตอร์	$A \leftarrow A - n$	SUB A,#n
4	ส่งข้อมูลออกแสดงผล	$OP \leftarrow A$	OUT
F	หยุดทำงาน		HALT

รหัสคำสั่งหมายถึง คำสั่งที่อยู่ในรูปของเลขฐานสิบหก เช่นถ้าเป็นรหัส 1 จะหมายถึงการนำข้อมูลมาเก็บไว้ในรีจิสเตอร์ A โดยข้อมูลที่มาเก็บ จะอยู่ในหน่วยความจำตำแหน่งถัดจากรหัส 1 ไป

วงจรถอดรหัสคำสั่ง (Instruction Decoder)

เป็นวงจรที่นำเอารหัสคำสั่งต่างๆ ตามตารางที่ 2.2 มาสร้างเป็นสัญญาณควบคุมของแต่ละคำสั่ง เช่น ถ้าคำสั่งที่มีรหัสเป็น 1H (เลขฐานสิบหก) หรือ 0001B (เลขฐานสอง) เข้าสู่วงจรถอดรหัส จะมีเพียงสัญญาณ MOVA เท่านั้นที่เป็นลอจิก 1 นอกนั้นเป็นลอจิก 0 ทั้งหมด



รูปที่ 2.16 โลจิกไดอะแกรมและสัญลักษณ์ของวงจรถอดรหัสคำสั่ง

การทำงานของคำสั่ง

การที่จะให้เกิดการทำงานต่างๆได้ คำสั่งต่างๆต้องจัดการทำงานเป็นจังหวะ ในที่นี้เรียกว่า T0 ถึง T4 และเพื่อให้ง่ายต่อการทำความเข้าใจ จึงกำหนดรหัสช่วยจำหรือรูปแบบคำสั่งในแบบคำย่อ ดังนี้

ตารางที่ 2.3 การทำงานในแต่ละจังหวะของคำสั่งต่างๆ

รหัส	รหัสช่วยจำ	การทำงาน	T0	T1	T2	T3	T4
1	MOV A,#n	A ← n	IR ← Memory	PC ← PC+1	A ← Memory	PC ← PC+1	
2	ADD A,#n	A ← A + n	IR ← Memory	PC ← PC+1	B ← Memory	PC ← PC+1	A ← (A+B)
3	SUB A,#n	A ← A - n	IR ← Memory	PC ← PC+1	B ← Memory	PC ← PC+1	A ← (A-B)
4	OUT	O/P ← A	IR ← Memory	PC ← PC+1	O/P ← A		
F	HALT		IR ← Memory	PC ← PC+1	STOP		