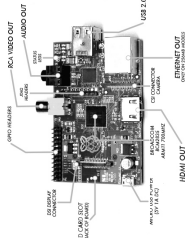
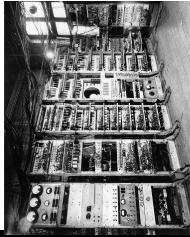
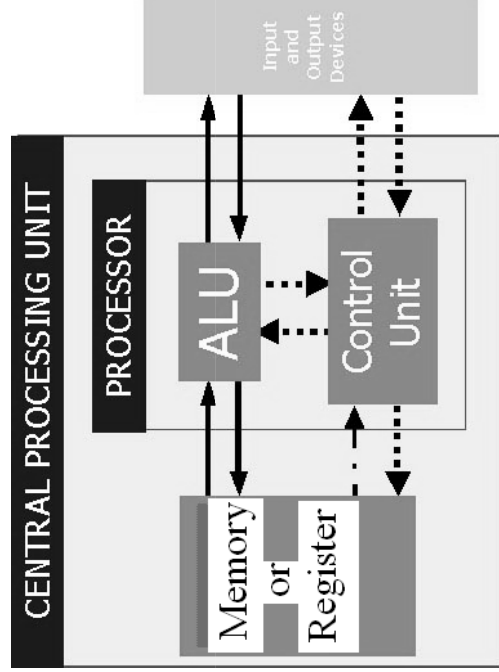


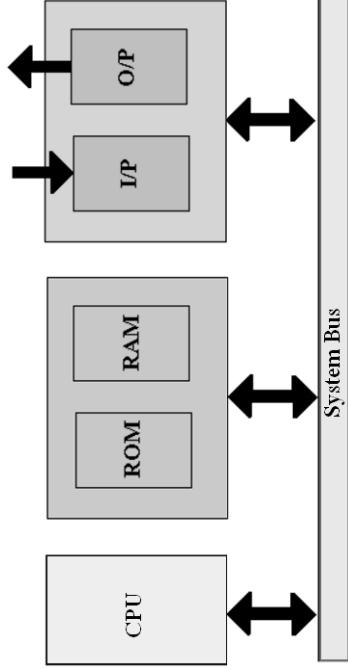
Simple Computer : SiCo



CPU : Central processing unit



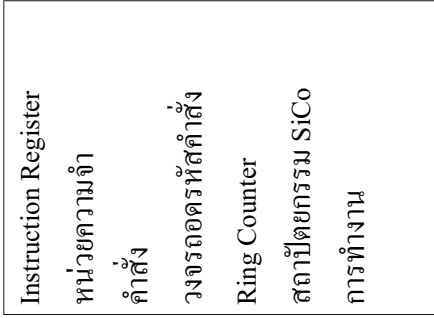
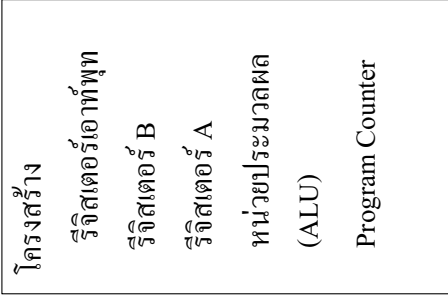
1. บทนำ ระบบคอมพิวเตอร์

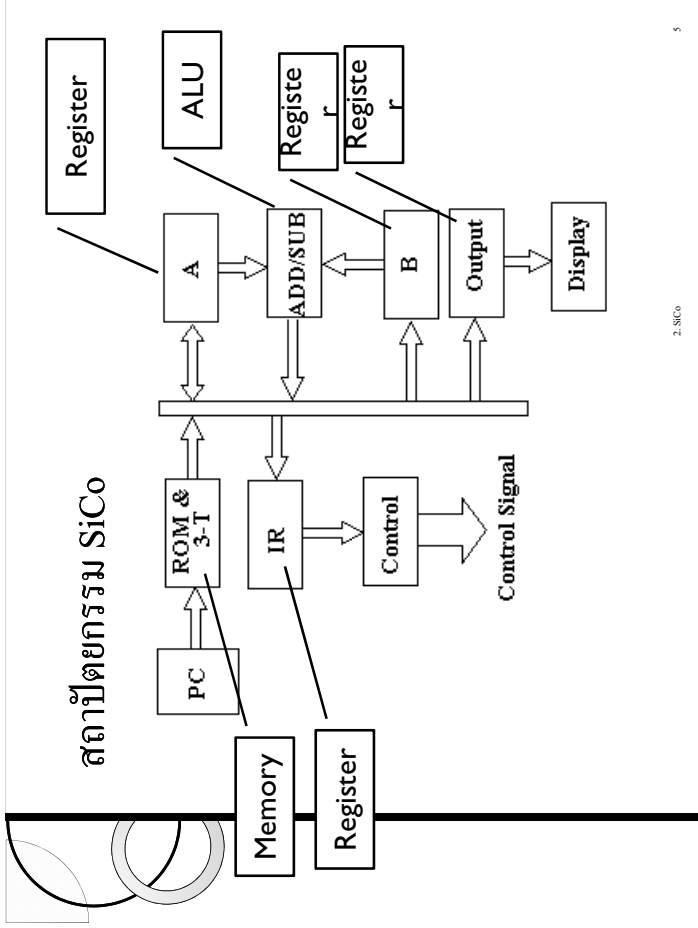


CPU = Central Processing Unit
ROM = Read Only Memory
RAM = Random Access Memory

I/O = Input/Output
I/P = Input
O/P = Output

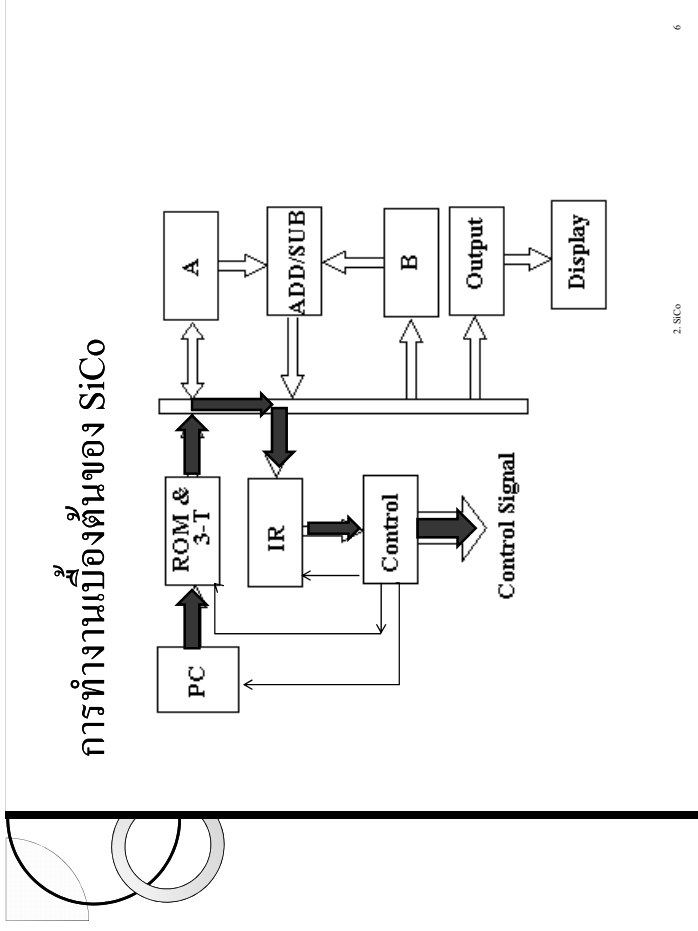
2. สถาปัตยกรรม SiCo คอมพิวเตอร์อย่างง่าย (Simple Computer)





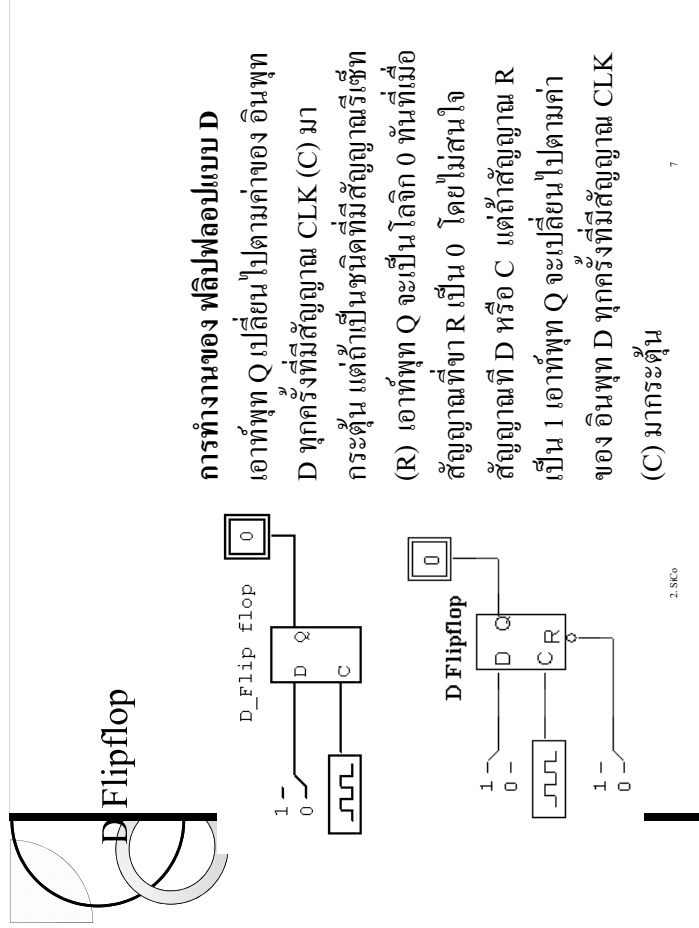
2. SiCo

5



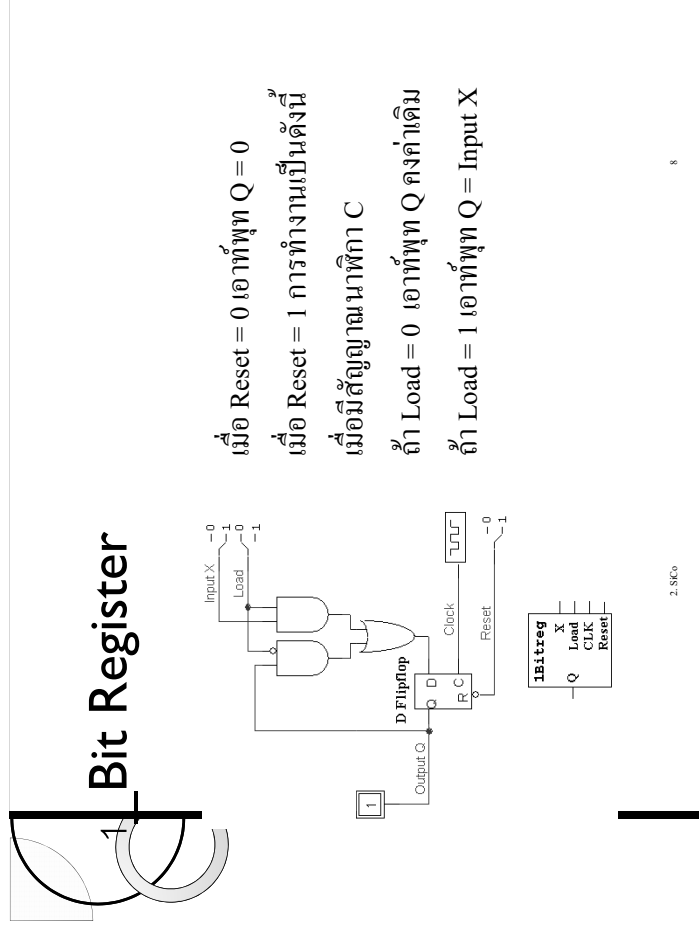
2. SiCo

6



2. SiCo

7



2. SiCo

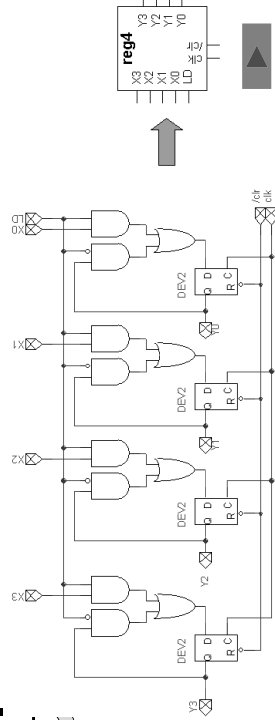
8

การทำงานของ ฟลิปฟล็อปแบบ D

เอาต์พุต Q เปลี่ยนไปตามค่าของ อินพุต D ทุกครั้งที่มีสัญญาณ CLK (C) มา กระตุ้น แต่ถ้าเป็นชนิดที่มีสัญญาณรีเซ็ต (R) เอาต์พุต Q จะเป็น 0 โดยไม่สนใจสัญญาณที่ขา R เป็น 0 โดยไม่สนใจสัญญาณที่ D หรือ C แต่ถ้าสัญญาณ R เป็น 1 เอาต์พุต Q จะเปลี่ยนไปตามค่าของ อินพุต D ทุกครั้งที่มีสัญญาณ CLK (C) มากระตุ้น

เมื่อ Reset = 0 เอาต์พุต Q = 0
เมื่อ Reset = 1 การทำงานเป็นดังนี้
เมื่อมีสัญญาณนาฬิกา C
ถ้า Load = 0 เอาต์พุต Q คงค่าเดิม
ถ้า Load = 1 เอาต์พุต Q = Input X

4 Bit Buffer Register



ถ้า $LD/CLR = 0$ เอาท์พุท $Y_3Y_2Y_1Y_0 = "0000"$

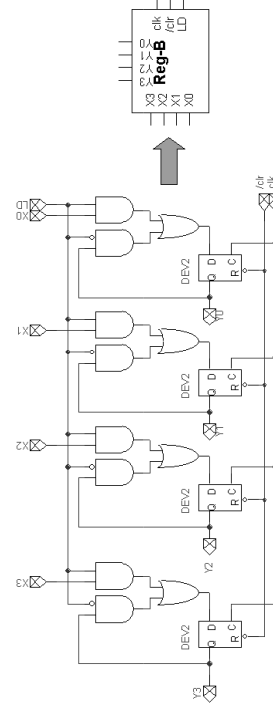
ถ้า $LD/CLR = 1$ การทำงานขึ้นอยู่กับ CLK และ LD โดยมี CLK มากะต้น

ถ้า Load = 0 $Y_3 - Y_0$ จะคงค่าเดิม

ถ้า Load = 1 $Y_3 = X_3, Y_2 = X_2, Y_1 = X_1$ และ $Y_0 = X_0$

รีจิสเตอร์ B

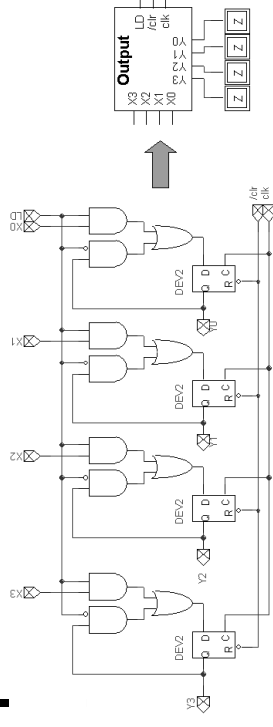
(B Register)



รีจิสเตอร์เอาท์พุท

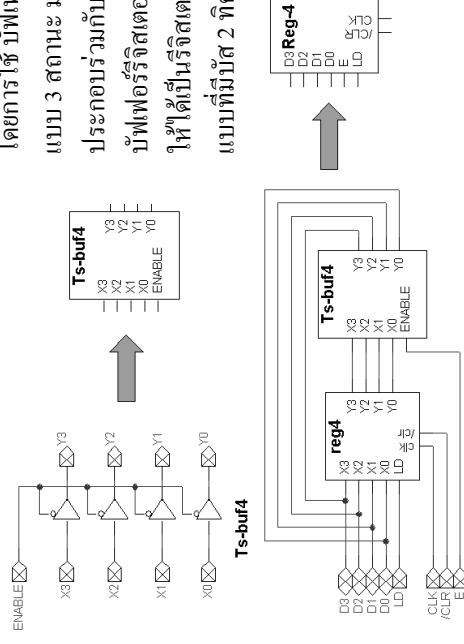
(Output Register)

ให้ Buffer Register ขนาด 4 บิต ทำเป็นรีจิสเตอร์เอาท์พุทสำหรับการแสดงผล

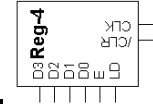


4 Bit Buffer Register with Bidirectional Bus

โดยการรีจิสเตอร์
แบบ 3 สถานะ มา
ประกอบร่วมกับ
รีจิสเตอร์ที่ทำ
ให้เป็นรีจิสเตอร์
แบบที่มีบิต 2 ทิศทาง



การทำงานของ Bidirectional Bus Buffer Register

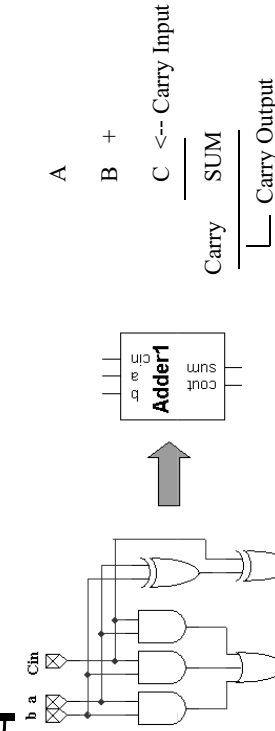


การทำงาน

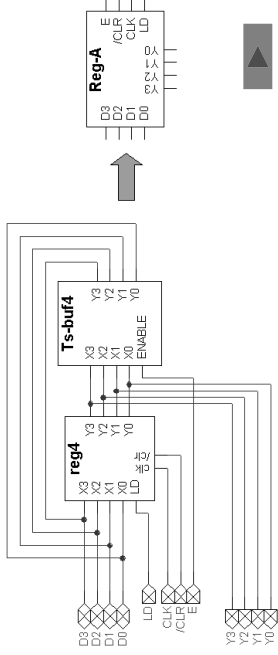
ถ้า E = 1 สัญญาณเอาต์พุตของ D3, D2, D1 และ D0 จะอยู่ในสถานะ High Impedance ใช้ในสถานะรอรับข้อมูลอินพุต โดย ถ้า LD = 1 จะรับข้อมูลที่ D3-D0 ไปเก็บไว้ภายในรีจิสเตอร์ ถ้าเป็น 1 จะคงค่าข้อมูลเดิมไม่เก็บข้อมูลใหม่

ถ้า E = 0 ข้อมูลที่เก็บอยู่ในรีจิสเตอร์จะปรากฏออกทางสัญญาณ D3-D0 ผ่านสัญญาณ /CLR ใช้สำหรับทำให้ข้อมูลภายในรีจิสเตอร์เป็น 0 ทั้งหมด

วงจรบวก Full Adder

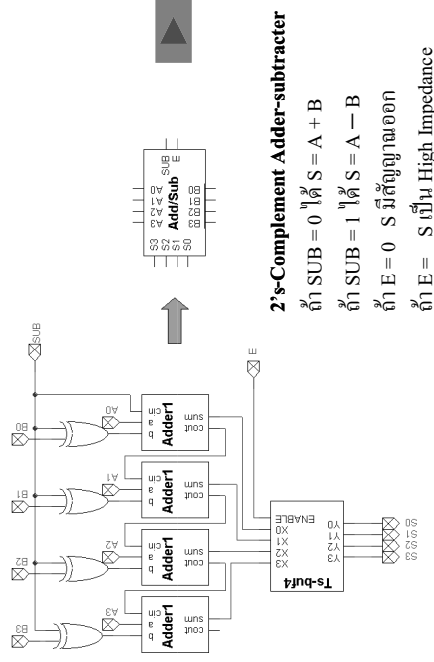


รีจิสเตอร์ A



D0-3 เป็นบัส 2 ทิศทาง
Y0 — 3 เป็นบัสทิศทางเดียว

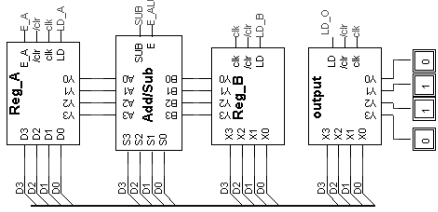
หน่วยประมวลผล (ALU) Arithmetic-Logic Units



2's-Complement Adder-subtractor

- ถ้า SUB = 0 ได้ $S = A + B$
- ถ้า SUB = 1 ได้ $S = A - B$
- ถ้า E = 0 S มีสัญญาณออก
- ถ้า E = S เป็น High Impedance

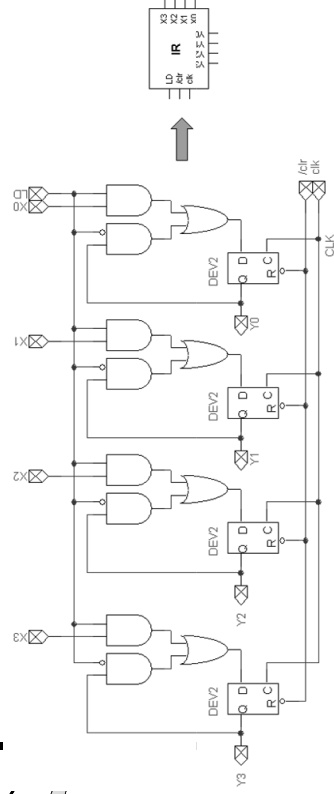
การต่อรีจิสเตอร์เข้ากับบัส



2. SCo

17

Instruction Register

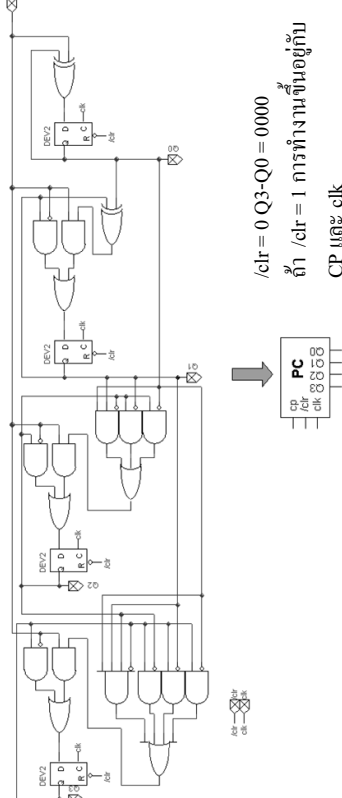


ใช้กับคำสั่งระหว่างปฏิบัติงาน

2. SCo

19

Program Counter (PC)

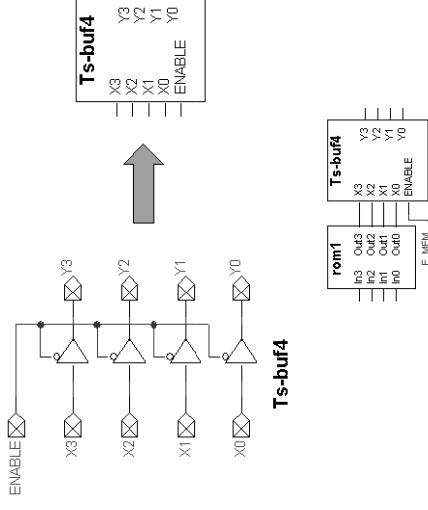


/clr = 0 Q3-Q0 = 0000
ถ้า /clr = 1 การทำงานขึ้นอยู่กับ
CP และ clk
CP = 0 ถ้านับคงเดิม
CP = 1 นับขึ้น

2. SCo

18

หน่วยความจำ



เก็บโปรแกรม

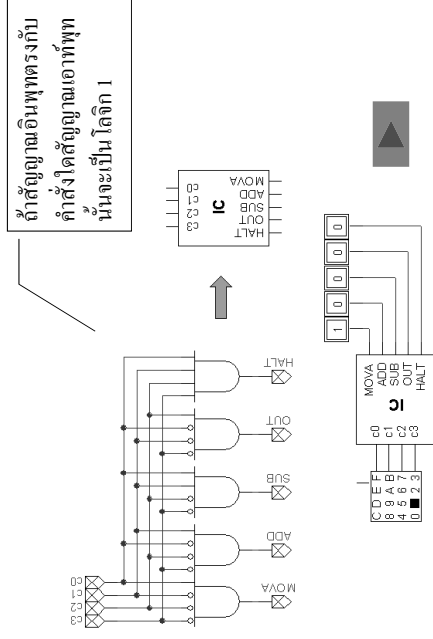
2. SCo

20

คำสั่ง

รหัส	ความหมาย	สัญลักษณ์การทำงาน
1	เก็บข้อมูลในรีจิสเตอร์ A	$A \leftarrow n$
2	บวกข้อมูลในรีจิสเตอร์ A กับข้อมูล n ผลลัพธ์เก็บที่ รีจิสเตอร์ A	$A \leftarrow A + n$
3	ลบข้อมูลในรีจิสเตอร์ A ด้วยข้อมูล n ผลลัพธ์เก็บที่ รีจิสเตอร์ A	$A \leftarrow A - n$
4	ส่งข้อมูลออกแสดงผล	$O/P \leftarrow A$
F	หยุดทำงาน	

วงจรถอดรหัสคำสั่ง (Instruction Decoder)

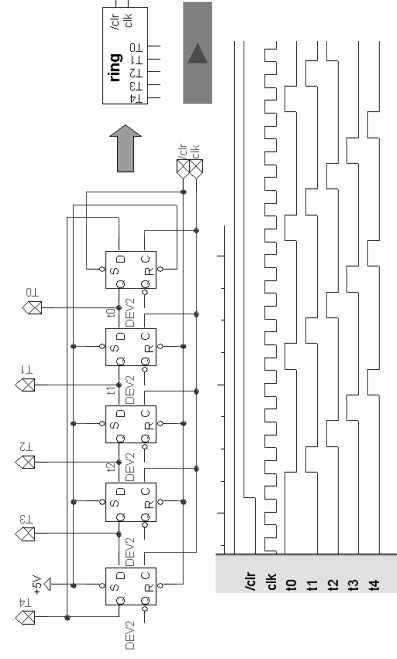


การทำงานของคำสั่ง

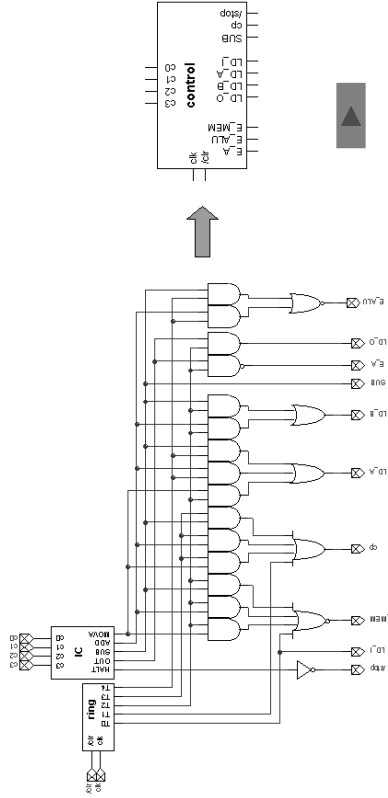
รหัส	รหัสช่วยจำ	การทำงาน	T0	T1	T2	T3	T4
1	MOV A, #n	$A \leftarrow n$	$IR \leftarrow \text{Memory}$	$PC \leftarrow PC+1$	$A \leftarrow \text{Memory}$	$PC \leftarrow PC+1$	
2	ADD A, #n	$A \leftarrow A + n$	$IR \leftarrow \text{Memory}$	$PC \leftarrow PC+1$	$B \leftarrow \text{Memory}$	$PC \leftarrow PC+1$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

T0 ถึง T4 หมายถึงจังหวะการทำงานที่ 0 ถึง 4
ซึ่งทำได้โดยการใช้อย่าง Ring Counter

Ring Counter



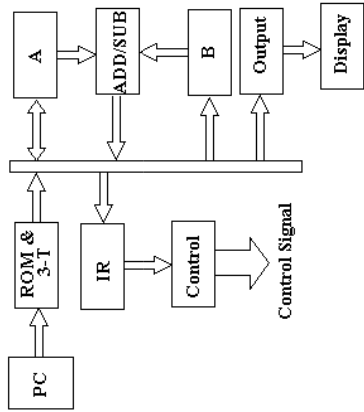
หน่วยควบคุม (Control Unit)



2. SiCo

25

สถาปัตยกรรม SiCo



2. SiCo

27

ตัวอย่างโปรแกรม

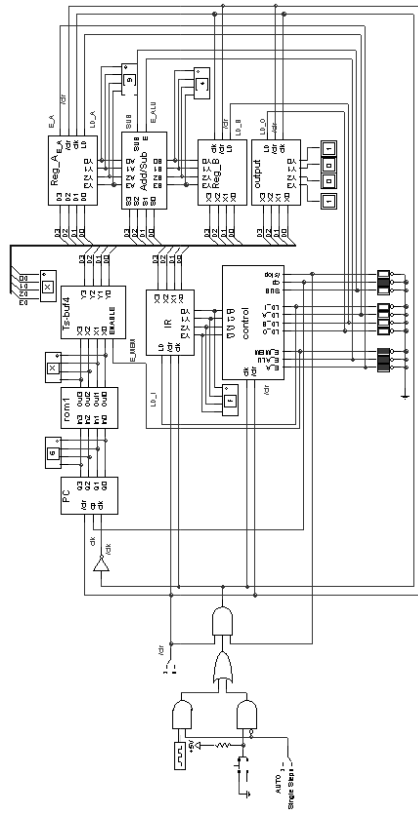
		หน่วยความจำ	
Address	Mnemonic	Address	Data
0	ORG 0	0	1
1	MOV A,#5 ; Set A = 5	1	5
2	ADD A,#4 ; Add A with 4	2	2
3	OUT ; Display result	3	4
4	HALT ; Stop	4	4
5		5	F

ข้อมูลที่จัดเก็บในหน่วยความจำ

2. SiCo

26

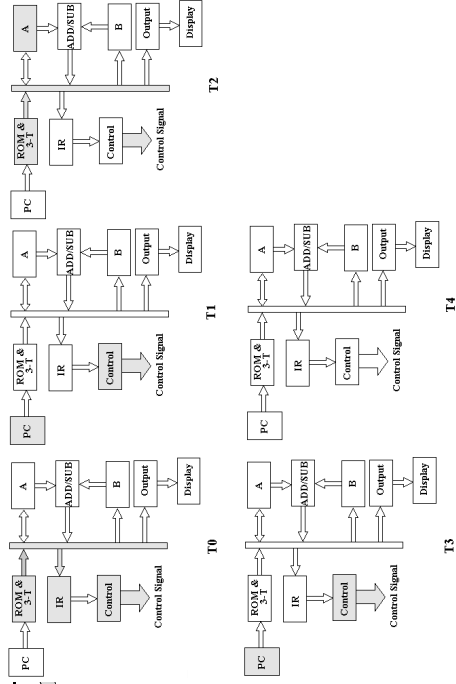
ไดอะแกรมของ SiCo



2. SiCo

28

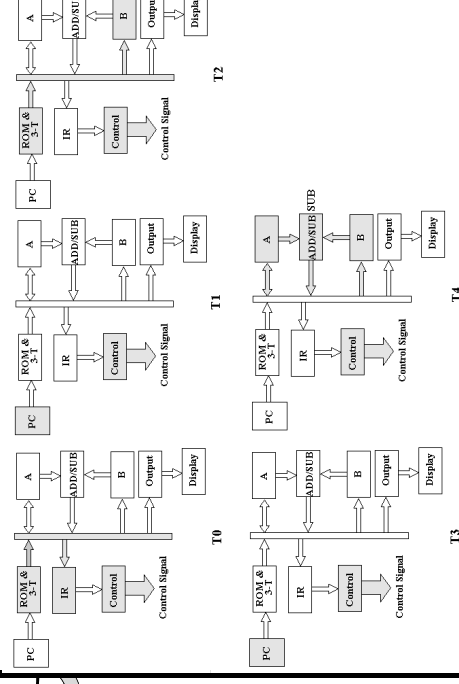
การทำงานของคำสั่ง MOV A,#n



2. SCo

29

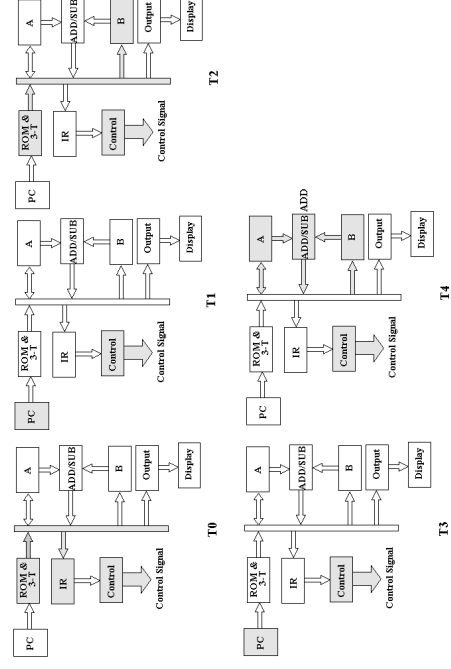
การทำงานของคำสั่ง SUB A,#n



2. SCo

31

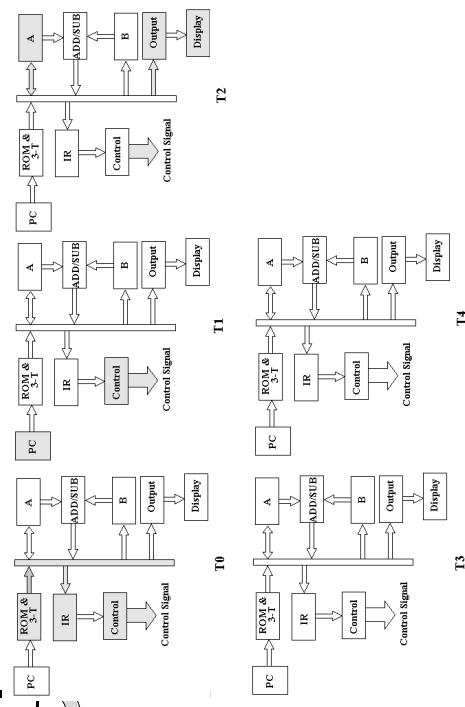
การทำงานของคำสั่ง ADD A,#n



2. SCo

30

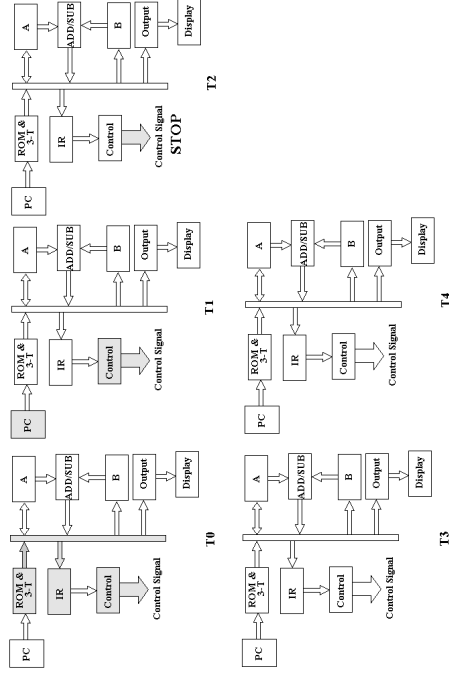
การทำงานของคำสั่ง OUT



2. SCo

32

การทำงานของคำสั่ง HALT



33

2. Sico

ตัวอย่างโปรแกรม

Address	Data	Mnemonic		Comment
		Opcode	Operand	
0		ORG	0	
1	5	MOV	A,#5	; Set A = 5
2	4	ADD	A,#4	; Add A with 4
3	4	OUT		; Display result
4	F	HALT		; Stop

ข้อมูลที่จัดเก็บในหน่วยความจำ

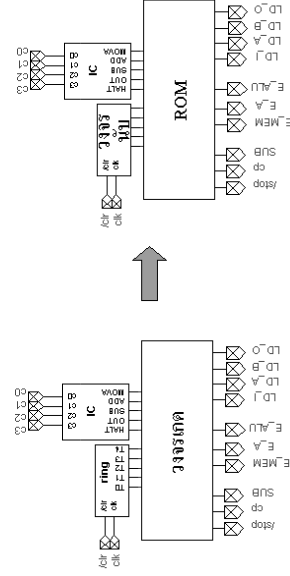


34

2. Sico

Microprogramming

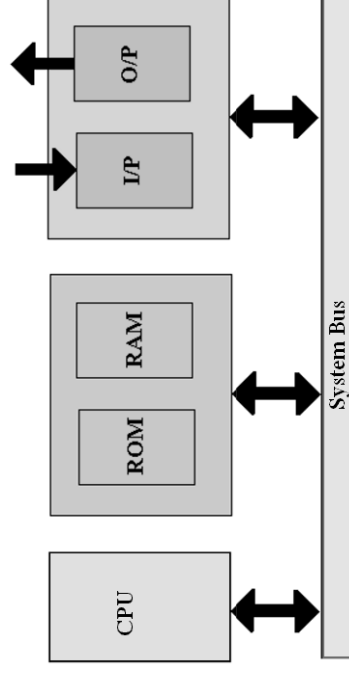
Microprogramming เป็นการ ใช้สัญญาณจาก ROM มาทำหน้าที่แทนวงจรควบคุมทำให้สามารถแก้ไขและเปลี่ยนแปลงการทำงานของวงจรได้ง่ขึ้น



35

2. Sico

การทำงานของคอมพิวเตอร์เบื้องต้น



CPU = Central Processing Unit
 ROM = Read Only Memory
 RAM = Random Access Memory
 I/O = Input/Output
 I/P = Input
 O/P = Output

36

2. Sico