# ระบบคอมพิวเตอร์และสถาปัตยกรรม (Computer System and Architecture)



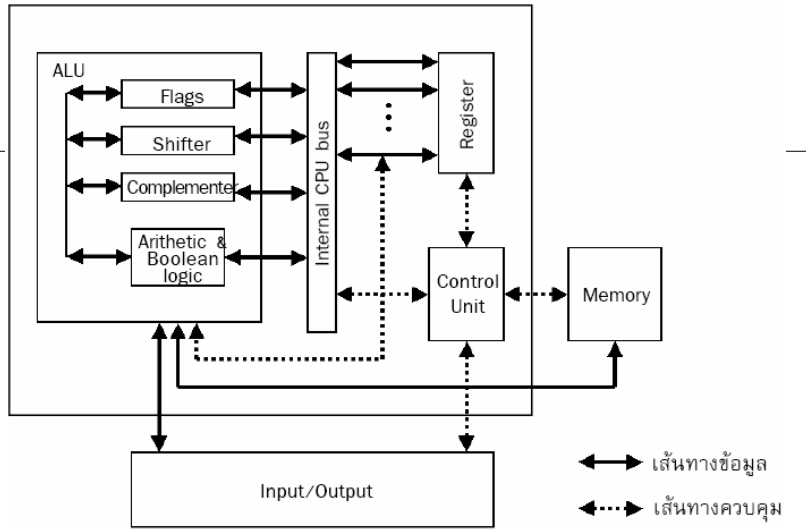
# บทที่ 12
# หน่วยควบคุมและเส้นทางข้อมูล (Control Unit and Data path)



## หน่วยควบคุม (Control Unit)

- ฟังก์ชันของโปรเซสเซอร์
  - Opcode
  - การกำหนดโหมดของแอ็ดเดรส
  - รีจิสเตอร์
  - อินพุต/เอาต์พุตโมดูลอินเทอร์เฟซ (I/O Module Interface)
  - เมมโมรีโมดูลอินเทอร์เฟซ (Memory Module Interface)
  - โครงสร้างการโปรเซสอินเทอร์รัพต์
- ทั้ง 6 ประการนี้เป็นสิ่งที่โปรเซสเซอร์ต้องทำ โดยการควบคุมของ “หน่วยควบคุม” (Control Unit)





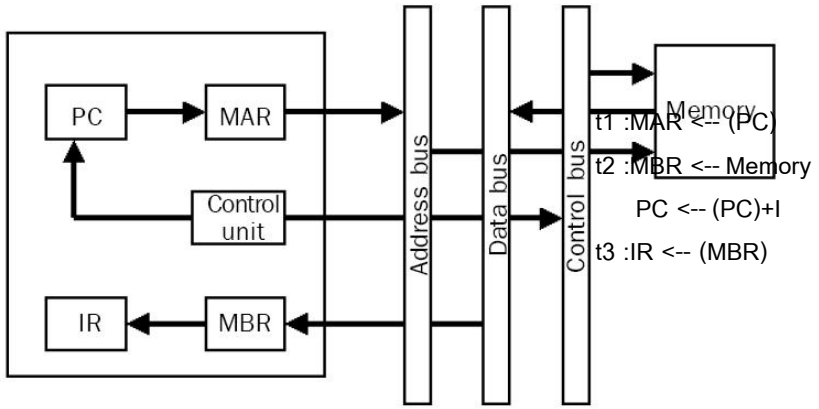

ความสัมพันธ์ของหน่วยควบคุม, เส้นทางข้อมูล หน่วยความจำ และอินพุต/เอาต์พุต

# ไมโครโอเปอเรชัน (Micro-Operation)

- ชุดของวงรอบคำสั่ง (Instruction Cycle) มีวงรอบเฟ็ตช์ (fetch), อินไดเร็กต์ (indirect), เอ็กซิคิวต์ (execute) และอินเทอร์รัพต์ (interrupt) เป็นหน่วยย่อย ๆ ประกอบอยู่ภายใน
- แต่ละหน่วยย่อยจะมีขั้นตอนย่อย ๆ หลายขั้นตอนที่เรียกว่า “ไมโครโอเปอเรชัน” (Micro Operation)
- รีจิสเตอร์ในซีพียู ประกอบด้วย
  - เมมโมรีแอ็ดเดรสรีจิสเตอร์ (Memory Address Register : MAR)
  - เมมโมรีบัฟเฟอร์รีจิสเตอร์ (Memory Buffer Register : MBR)
  - โปรแกรมเคาเตอร์ (Program Counter : PC)
  - อินสตรัคชันรีจิสเตอร์ (Instruction Register : IR)

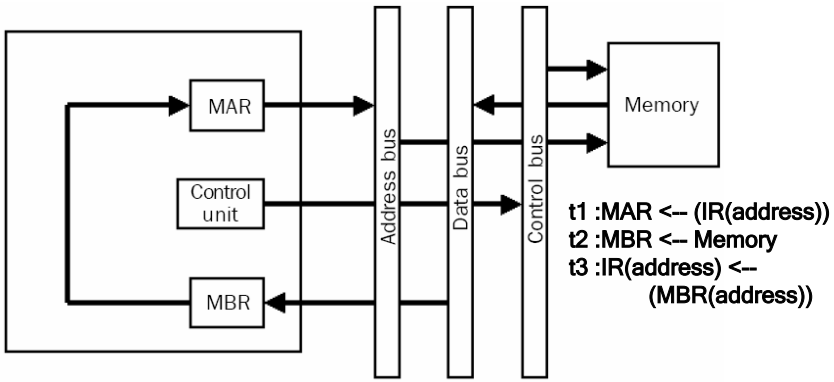


# Fetch Cycle Flow



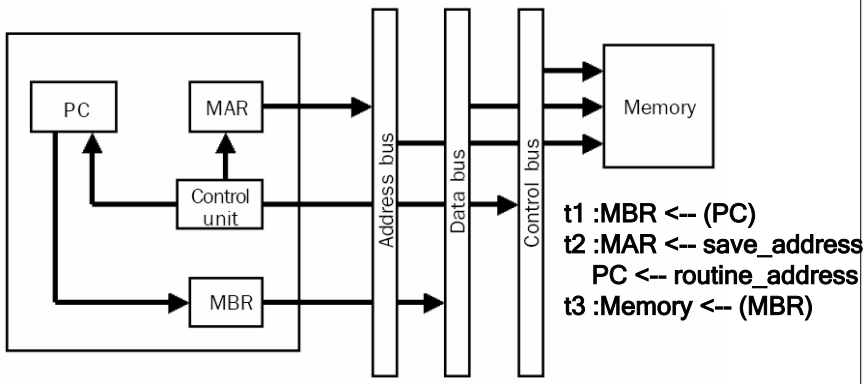



# Data flow in Indirect Circuit






# Data flow in Interrupt Circuit



# Execute Cycle

- Micro Operation สำหรับ Execute Cycleจะแตกต่างจาก Micro Operation ใน fetch cycle, indirect และ interrupt
- ใน Execute Cycle จะเปลี่ยนไปตาม Operation ที่โปรเซสเซอร์ทำงาน

  Add R1, X
- Micro Operation จะเป็น

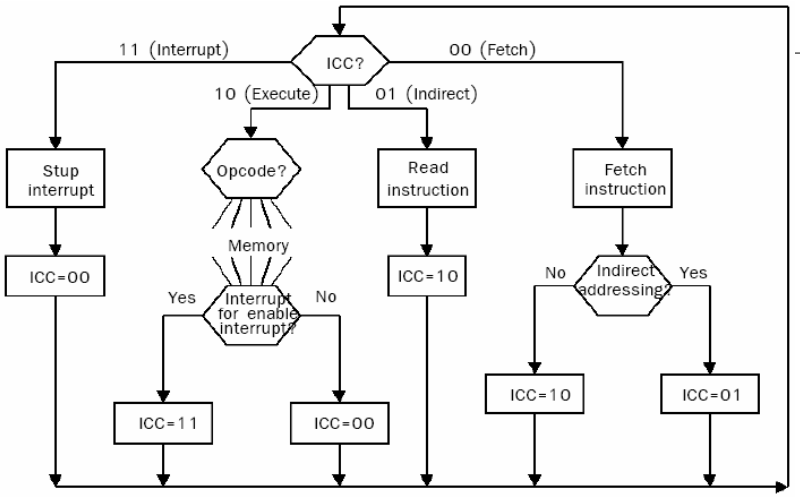
  **t1 : MAR <-- (IR(Address))**

  **t2 : MBR <-- Memory**

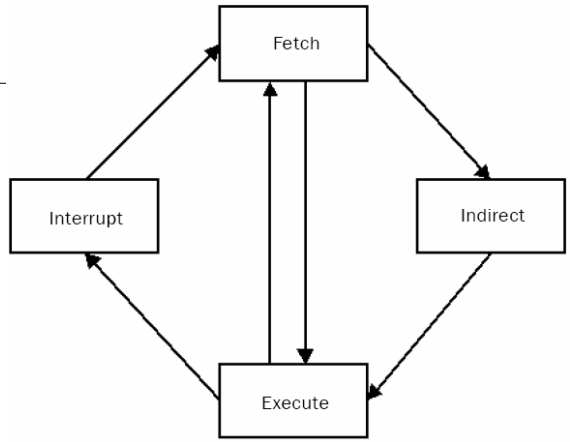
  **t3 : R1 <-- (R1) + (MBR)**



## วงรอบคำสั่ง






ลำดับวงรอบคำสั่ง



# การควบคุมของโปรเซสเซอร์

- ฟังก์ชันพื้นฐานของหน่วยควบคุม
  - กำหนดฟังก์ชันพื้นฐานของโปรเซสเซอร์
  - อธิบาย Micro operation ที่โปรเซสเซอร์ทำ
  - กำหนดฟังก์ชันที่หน่วยควบคุมต้องทำเพื่อให้เกิด Micro Operation
- ฟังก์ชันพื้นฐานของโปรเซสเซอร์
  - ALU
  - รีจิสเตอร์
  - เส้นทางข้อมูลภายใน
  - หน่วยควบคุม

# การควบคุมของโปรเซสเซอร์

- กลุ่มของ Micro operation
  - ย้ายข้อมูลจากรีจิสเตอร์หนึ่งไปยังรีจิสเตอร์อื่น
  - ย้ายข้อมูลจากรีจิสเตอร์ไปยังอินเทอร์เฟซภายนอก เช่น บัสของระบบ
  - ย้ายข้อมูลจากอินเทอร์เฟซภายนอกเข้ามายังรีจิสเตอร์
  - ทำโอเปอเรชันทางด้านคณิตศาสตร์และตรรกะ โดยใช้รีจิสเตอร์สำหรับอินพุตและเอาต์พุต
- งานหลักของหน่วยควบคุม
  - การจัดลำดับ (Sequencing)
  - การเอ็กซิคิวต์ (Execution)

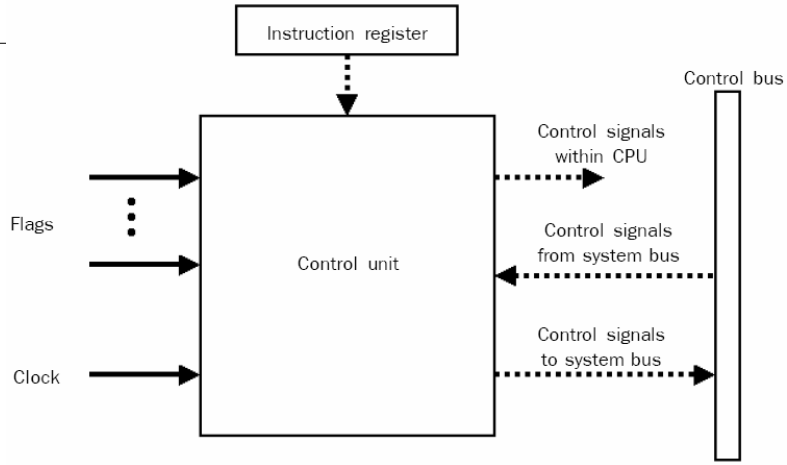


# สัญญาณควบคุม (Control Signal)

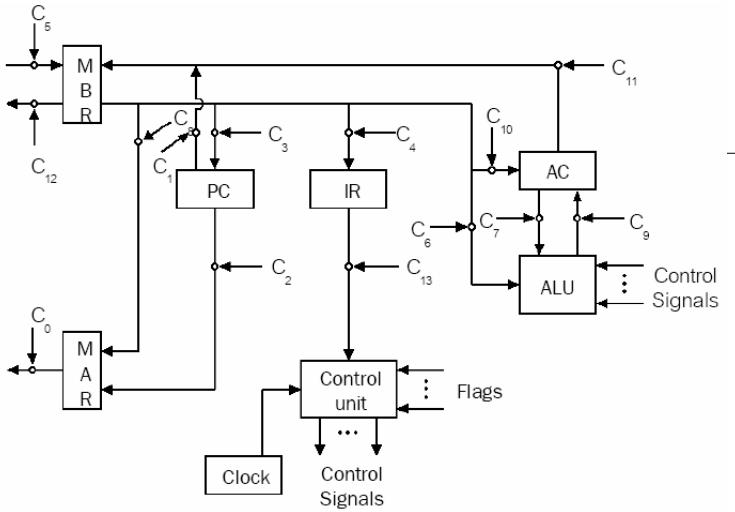
- **อินพุตของหน่วยควบคุม**
  - สัญญาณนาฬิกา (Clock)
  - อินสตรัคชันรีจิสเตอร์ (Instruction register)
  - แฟ็ก (Flag)
  - สัญญาณควบคุมจากคอนโทรลบัส (Control signals from control bus)
- **เอาต์พุตของหน่วย**
  - สัญญาณควบคุมภายในโปรเซสเซอร์ (Control signals within the processor)
  - สัญญาณควบคุมไปยังคอนโทรลบัส (Control signals to control bus)



# โมเดลของหน่วยควบคุม






ตัวอย่างเส้นทางข้อมูลและสัญญาณควบคุมที่มี AC เพียงตัวเดียว

# ตาราง Micro Operation และ Control Signal

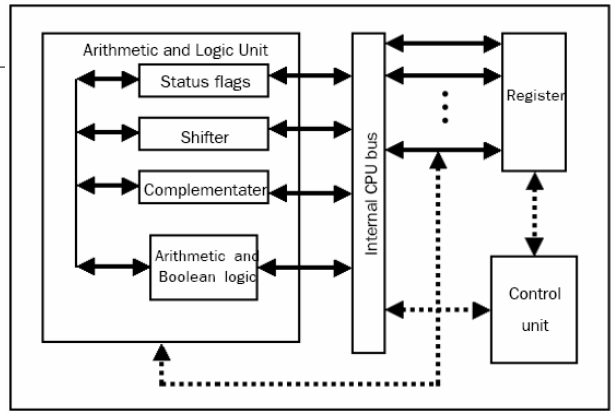
| วงรอบคำสั่ง | ไมโครโอเปอเรชัน | สัญญาณควบคุมที่แอ็กทีฟ |
|---|---|---|
| เฟ็ตช์ | t1 : MAR ⟵ (PC) | $C_2$ |
| | t2 : MBR ⟵ Memory | |
| | PC ⟵ (PC)+1 | $C_5$, $C_R$ |
| | t3 : IR ⟵ (MBR) | $C_4$ |
| อินไดเร็กต์ | t1 : MAR ⟵ (IR(Address)) | $C_8$ |
| | t2 : MBR ⟵ Memory | $C_5$, $C_R$ |
| | t3 : IR (Address) ⟵ (MBR (Address)) | $C_4$ |
| อินเทอร์รัพต์ | t1 : MBR ⟵ (PC) | $C_1$ |
| | t2 : MAR ⟵ Save_Address | |
| | PC ⟵ Routine_Address | |
| | t3 : Memory ⟵ (MBR) | $C_{12}$, $C_W$ |

$C_R$ คือสัญญาณควบคุมสำหรับการอ่านไปยังบัสระบบ

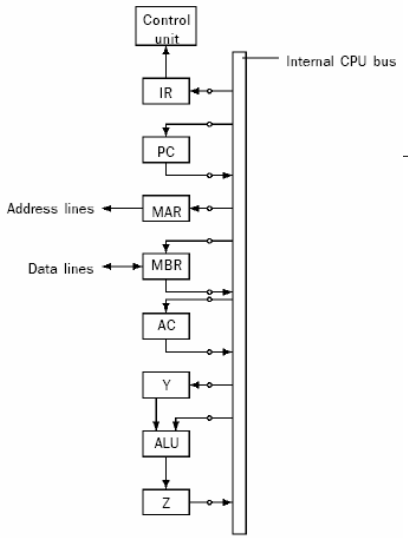
$C_W$ คือสัญญาณควบคุมสำหรับการเขียนไปยังบัสระบบ



# การจัดโครงสร้างภายในโปรเซสเซอร์





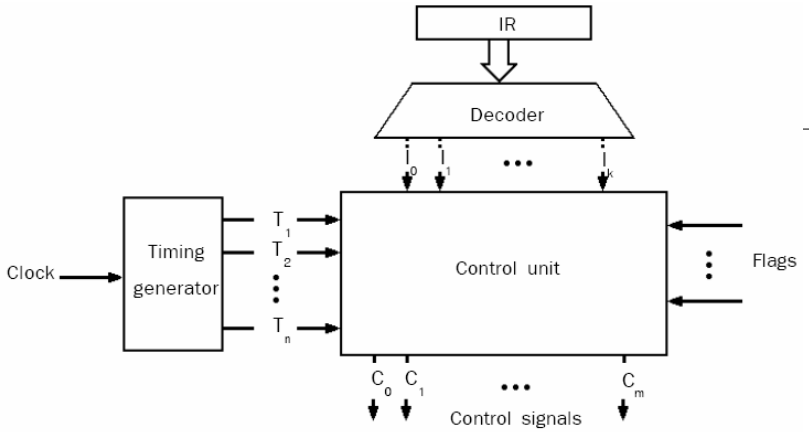




Internal Bus in CPU



# Hardware Structure

- **อินพุตของหน่วยควบคุม**
  - สัญญาณนาฬิกา (Clock)
  - อินสตรัคชันรีจิสเตอร์ (Instruction register)
  - แฟ็ก (Flag)
  - สัญญาณควบคุมจากคอนโทรลบัส (Control signals from control bus)
- หน่วยควบคุมอย่างง่ายจะมีอินพุตเป็นตรรกะเฉพาะอย่างในแต่ละ opcode ซึ่งทำได้โดยใช้ “ตัวถอดรหัส” (Decoder)




หน่วยควบคุมที่มีตัวถอดรหัส



## ตรรกะของหน่วยควบคุม

- พิจารณาตามรูป และตาราง ที่แสดงชุดของไมโครโอเปอเรชันของ 3 วงรอบ (เฟ็ตช์, อินไดเร็กต์ และอินเทอร์รัพท์)
- ที่สัญญาณควบคุม $C_5$ ที่มีใช้ในวงรอบเฟ็ตช์ และอินไดเร็กต์
- สมมุติกำหนดสัญญาณควบคุม 2 สัญญาณคือ P และ Q จะได้

PQ = 00 วงรอบเฟ็ตช์
PQ = 01 วงรอบอินไดเร็กต์
PQ = 10 วงรอบเอ็กซิคิวต์
PQ = 11 วงรอบอินเทอร์รัพต์

ทำให้ได้นิพจน์ $C_5$ เป็น

$$C_5 = P' Q' T_2 + P'Q T_2$$



## ตรรกะของหน่วยควบคุม

- สมมุติ execute Cycle มี 3 คำสั่งที่อ่านเข้ามาคือ LDA, ADD และ ANDทำให้ได้นิพจน์ $C_5$ เป็น

$$C_5 = P' Q' T_2 + P' Q T_2 + P Q' (LDA + ADD + AND) T_2$$

- จะเห็นว่ามีความซับซ้อน และยิ่งซับซ้อนมากยิ่งขึ้นสำหรับระบบคอมพิวเตอร์สมัยใหม่
- “ไมโครโปรแกรมมิ่ง” (Microprogramming) จึงถูกออกแบบขึ้นมาเพื่อลดความซับซ้อนนี้



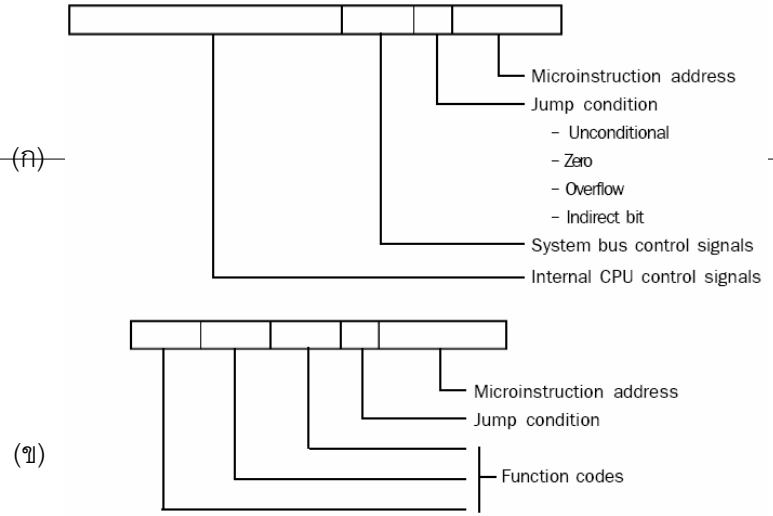
## ไมโครโปรแกรม (Microprogram)

- กำหนดขึ้นครั้งแรกในช่วงปี 1950 โดย M.V.Wilk
- ไม่ค่อยเกิดผลมากนัก เนื่องจากต้องใช้หน่วยความจำที่เร็วแต่ราคาไม่สูงมากนัก
- ข้อมูลใน Datamation ในปี 1964 เขียนไว้ว่าในขณะนั้นยังไม่มีระบบ ไมโครโปรแกรมใช้งานเลย
- System/360 ของ IBM ได้เปิดตัวโดยมีโมดูลใหญ่เป็นไมโครโปรแกรม ในช่วงไม่นานมานี้
- สิ่งหนึ่งที่มีการนำไมโครโปรแกรมมาใช้ประโยชน์ก็คือหน่วยควบคุมของโปรเซสเซอร์

# ไมโครโปรแกรม (Microprogram)

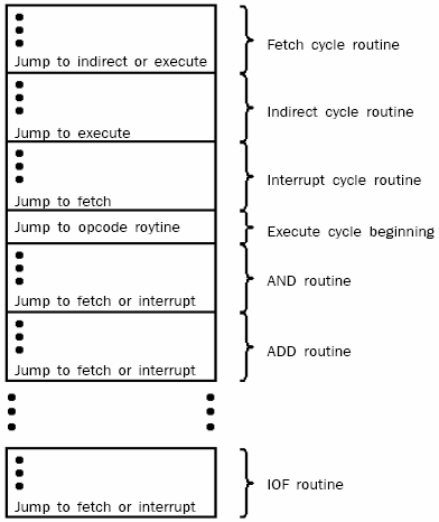
- โปรเซสเซอร์ CICS จะแสดงโครงสร้างของหน่วยควบคุมแบบไมโครโปรแกรม
- แต่ละไมโครโอเปอเรชันจะอธิบายในรูปแบบของสัญลักษณ์
- สัญลักษณ์คือภาษาที่เรียกว่า “ภาษาไมโครโปรแกรม” (Microprogramming language)
- แต่ละบรรทัดเป็นชุดของไมโครโอเปอเรชันที่เกิดขึ้นใน 1 เวลา รู้จักในชื่อ “ไมโครอินสตรัคชัน” (Microinstruction)
- ลำดับของคำสั่งรู้จักในชื่อของ “ไมโครโปรแกรม” (Microprogram) หรือ “เฟิร์มแวร์” (Firmware)






(ก) ไมโครอินสตรัคชันแนวนอน (ข) ไมโครอินสตรัคชันแนวตั้ง

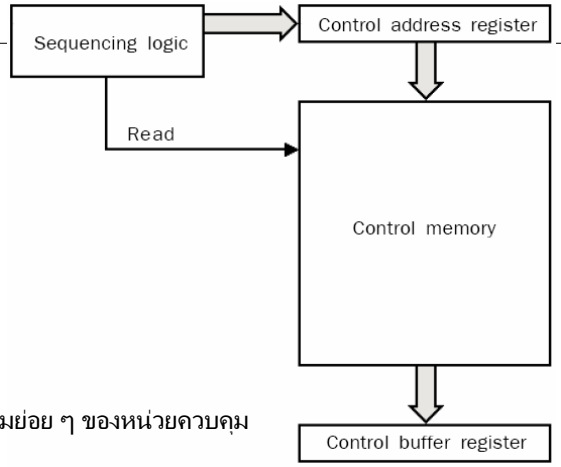




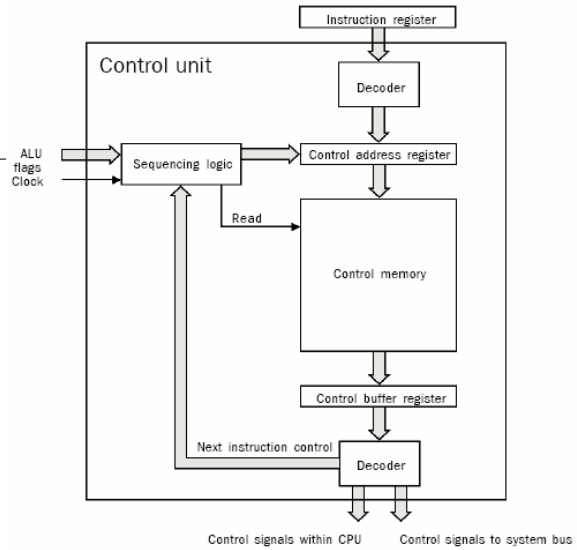

โครงสร้างของ control memory



# หน่วยควบคุมแบบไมโครโปรแกรม




สถาปัตยกรรมย่อย ๆ ของหน่วยควบคุม



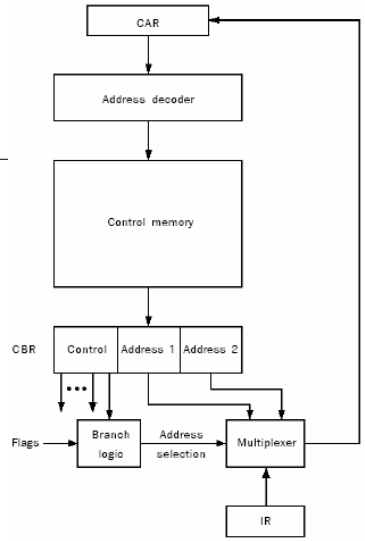

ฟังก์ชันของหน่วยควบคุมแบบไมโครโปรแกรม



# การจัดลำดับ Micro Instruction

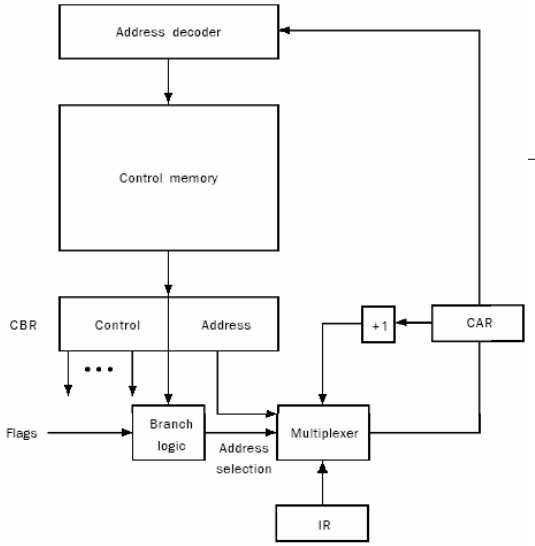
- ข้อพิจารณาการออกแบบ
  - ขนาดของไมโครอินสตรัคชัน
  - เวลาในการสร้างแอ็ดเดรส
- เทคนิคการจัดลำดับ
  - ใช้ 2 แอ็ดเดรสฟิลด์ (two address fields)
  - ใช้แอ็ดเดรสฟิลด์เดียว (single address field)
  - รูปแบบที่เป็นตัวแปร (variable format)



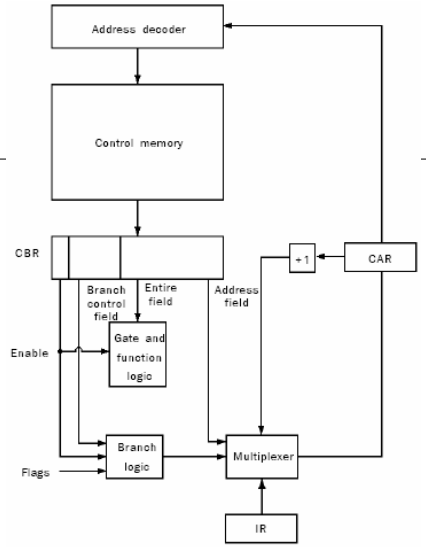



Branch Control Logic : แบบ 2 address fields






Branch Control Logic : แบบ1 address field



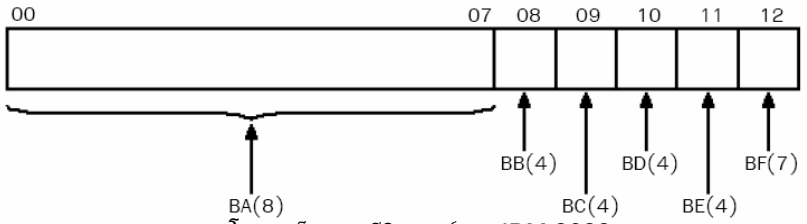

## Branch Control Logic : รูปแบบที่เป็นตัวแปร



## การสร้าง address

| Explicit | Implicit |
|---|---|
| 2 ฟิลด์ (two-fields) | การแม็พ (Mapping) |
| กระโดดแบบไม่มีเงื่อนไข (Conditional branch) | การบวก (Adding) |
| กระโดดแบบไม่มีเงื่อนไข (Unconditional branch) | การควบคุมส่วนที่เหลือ (Residual Control) |

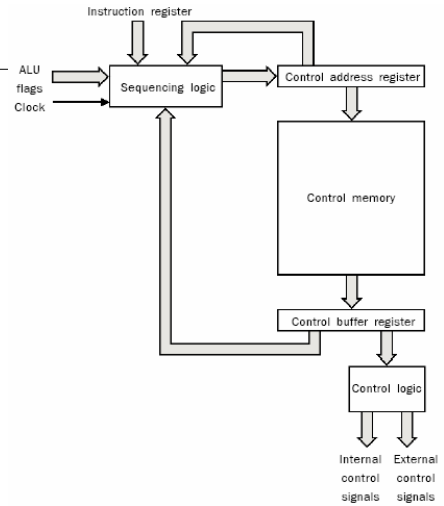
ตาราง เทคนิคการสร้าง address ของ micro instruction




คอนโทรลแอ็ดเดรสรีจิสเตอร์ของ IBM 3033

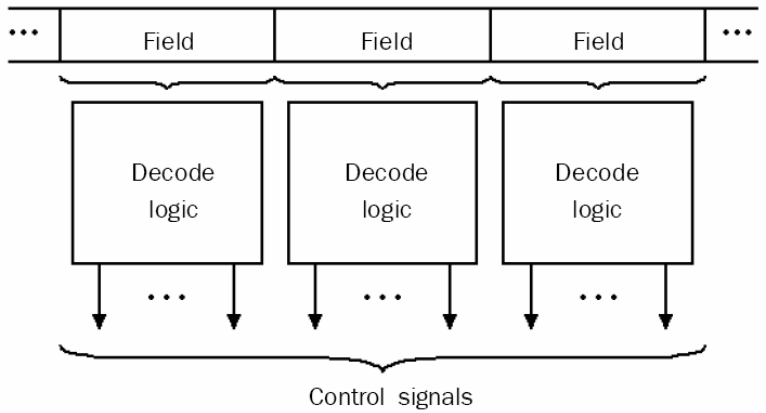


## Micro Instructions Execute



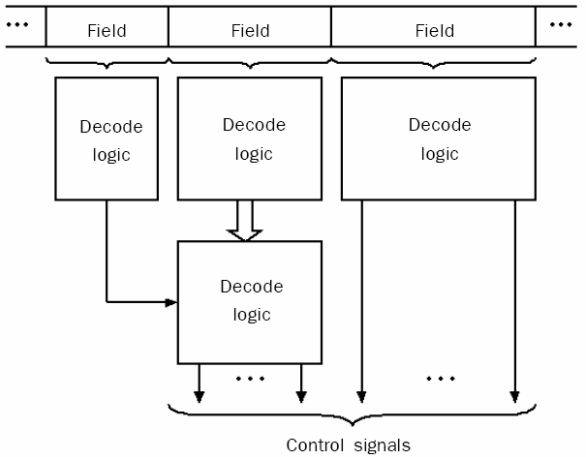



## Micro Instructions Encoding




## การเข้ารหัสโดยตรง (Direct Encoding)

# Micro Instructions Encoding




การเข้ารหัสโดยอ้อม (Indirect Encoding)