

# 16 Control Unit

A := A + 1;

(\* Pascal Instruction \*)

Load A

Add 1

Store A

Assembly L instructions  
Machine L instructions

Instruction cycle

Fetch

Indirect

Execute

Interrupt

sub-cycle

MAR ← (PC)

MBR ← Memory  
PC ← (PC) + 1

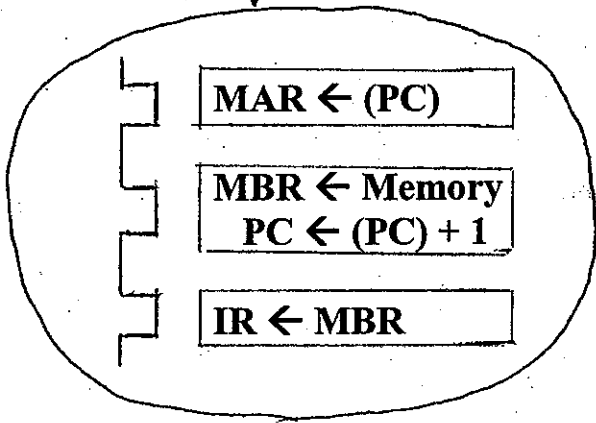
IR ← MBR

micro operation

Control Unit

makes these sequence of micro operations to occur

Note. CU is a combinational circuit



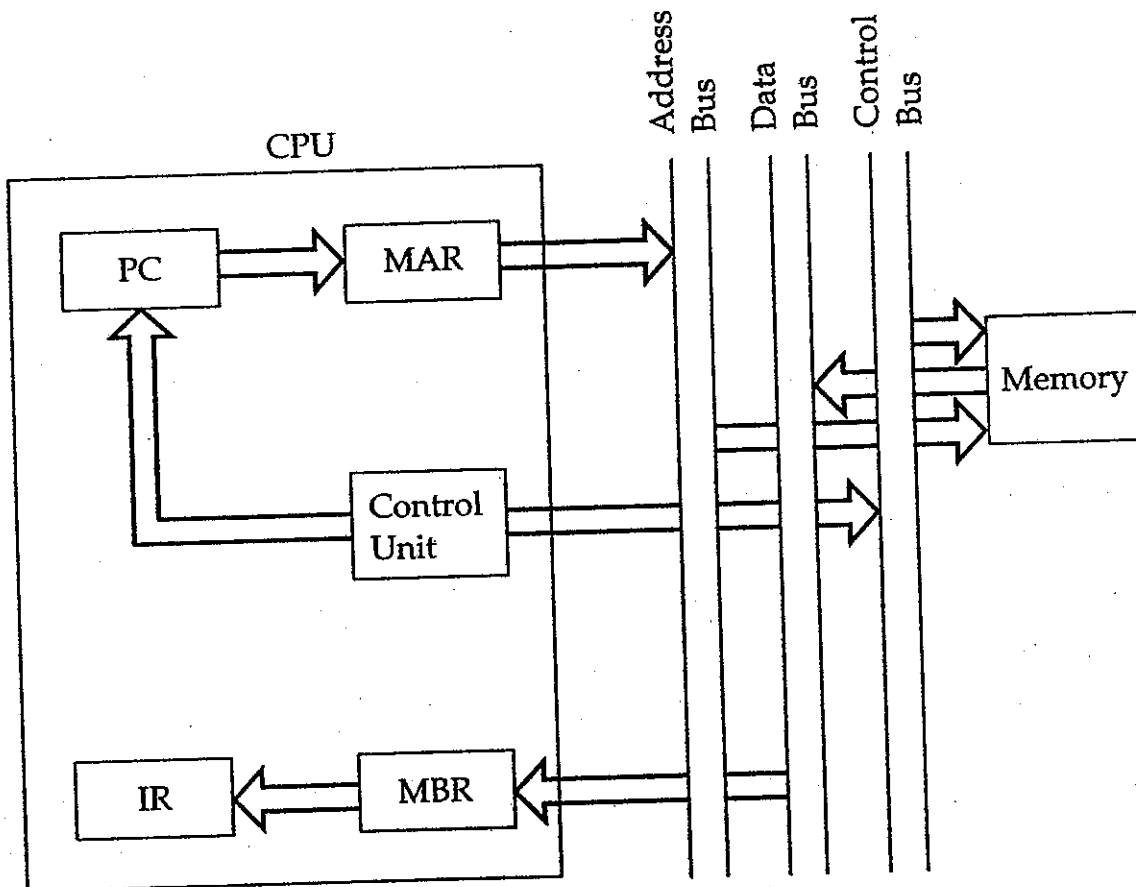


FIGURE Data flow, fetch cycle

$t_1: \text{MAR} \leftarrow (\text{PC})$   
 $t_2: \text{MBR} \leftarrow \text{Memory}$   
 $\text{PC} \leftarrow (\text{PC}) + 1$   
 $t_3: \text{IR} \leftarrow (\text{MBR})$

MAR	
MBR	
PC	0000000001100100
IR	
AC	

(a) Beginning

MAR	0000000001100100
MBR	
PC	0000000001100100
IR	
AC	

$t_1: \text{MAR} \leftarrow (\text{PC})$

(b) First Step

MAR	0000000001100100
MBR	0001000000100000
PC	0000000001100101
IR	
AC	

$t_2: \text{MBR} \leftarrow \text{Memory}$   
 $\text{PC} \leftarrow (\text{PC}) + 1$

(c) Second Step

MAR	0000000001100100
MBR	0001000000100000
PC	0000000001100101
IR	0001000000100000
AC	

$t_3: \text{IR} \leftarrow (\text{MBR})$

(d) Third Step

**FIGURE** Sequence of events, fetch cycle

## The Indirect Cycle

$t_1: \text{MAR} \leftarrow (\text{IR}(\text{Address}))$   
 $t_2: \text{MBR} \leftarrow \text{Memory}$   
 $t_3: \text{IR}(\text{Address}) \leftarrow (\text{MBR}(\text{Address}))$

## The Execute Cycle

ADD R1, X

$t_1: \text{MAR} \leftarrow (\text{IR}(\text{address}))$   
 $t_2: \text{MBR} \leftarrow \text{Memory}$   
 $(*) \quad t_3: \text{R1} \leftarrow (\text{R1}) + (\text{MBR})$

## The Interrupt Cycle

- $t_1$ : MBR  $\leftarrow$  (PC)  
 $t_2$ : MAR  $\leftarrow$  Save-address  
     PC  $\leftarrow$  Routine-address  
 $t_3$ : Memory  $\leftarrow$  (MBR)

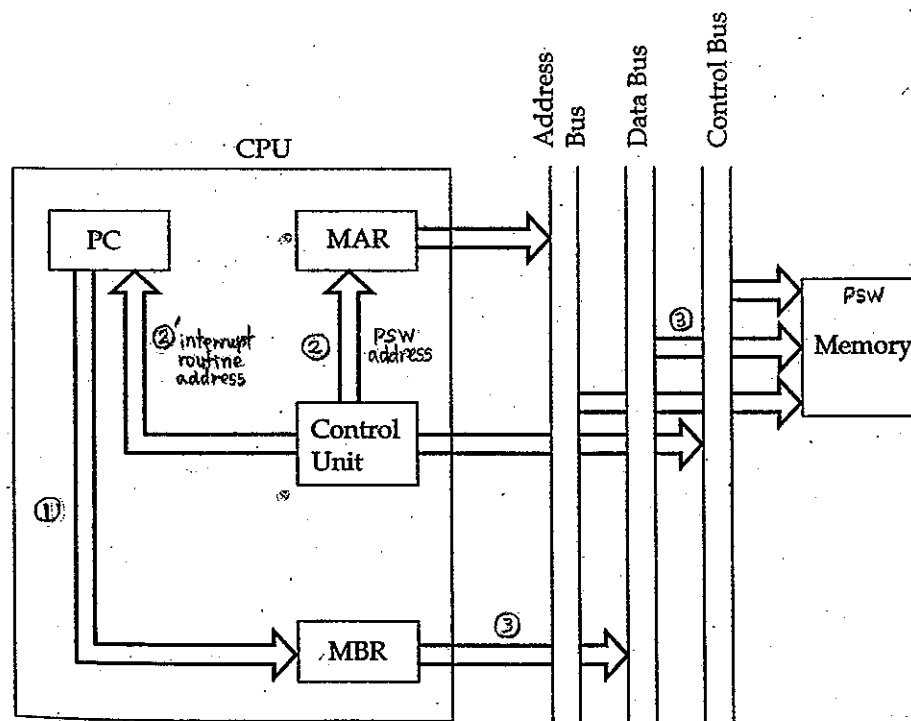


FIGURE Data flow, interrupt cycle.

# The Instruction Cycle

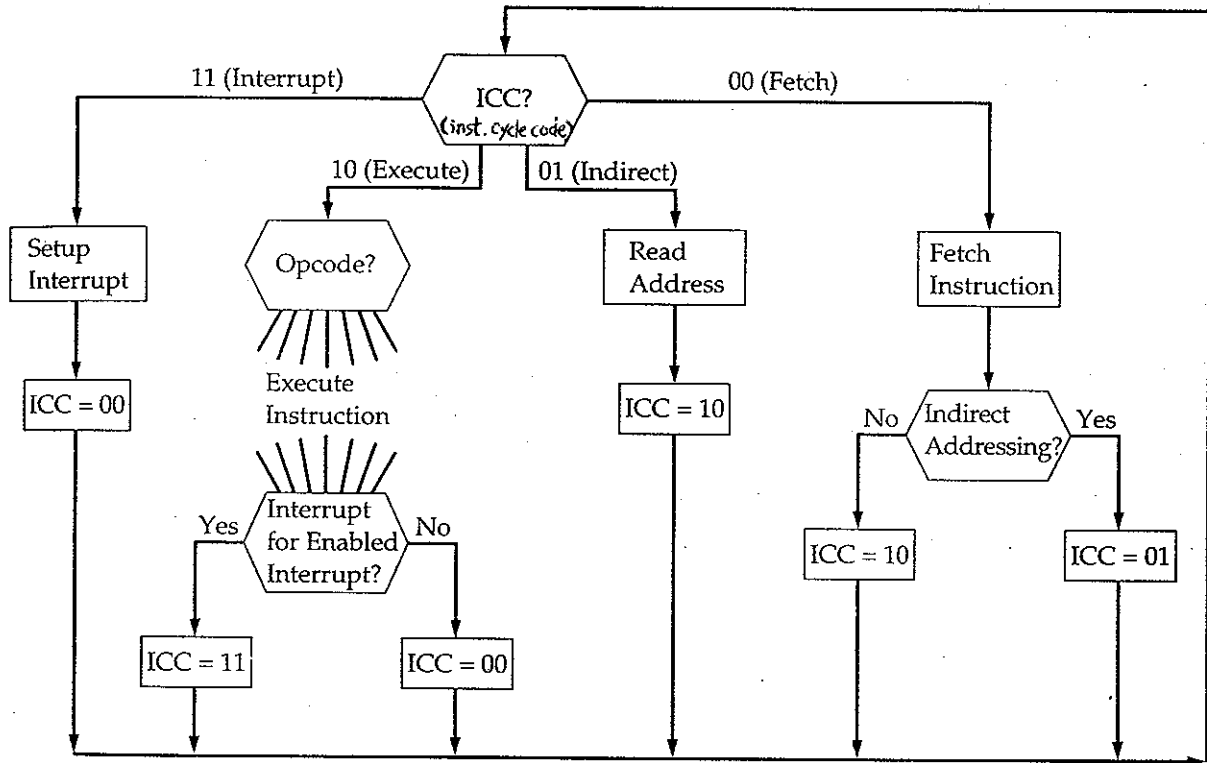


FIGURE 16.3. Flowchart for instruction cycle

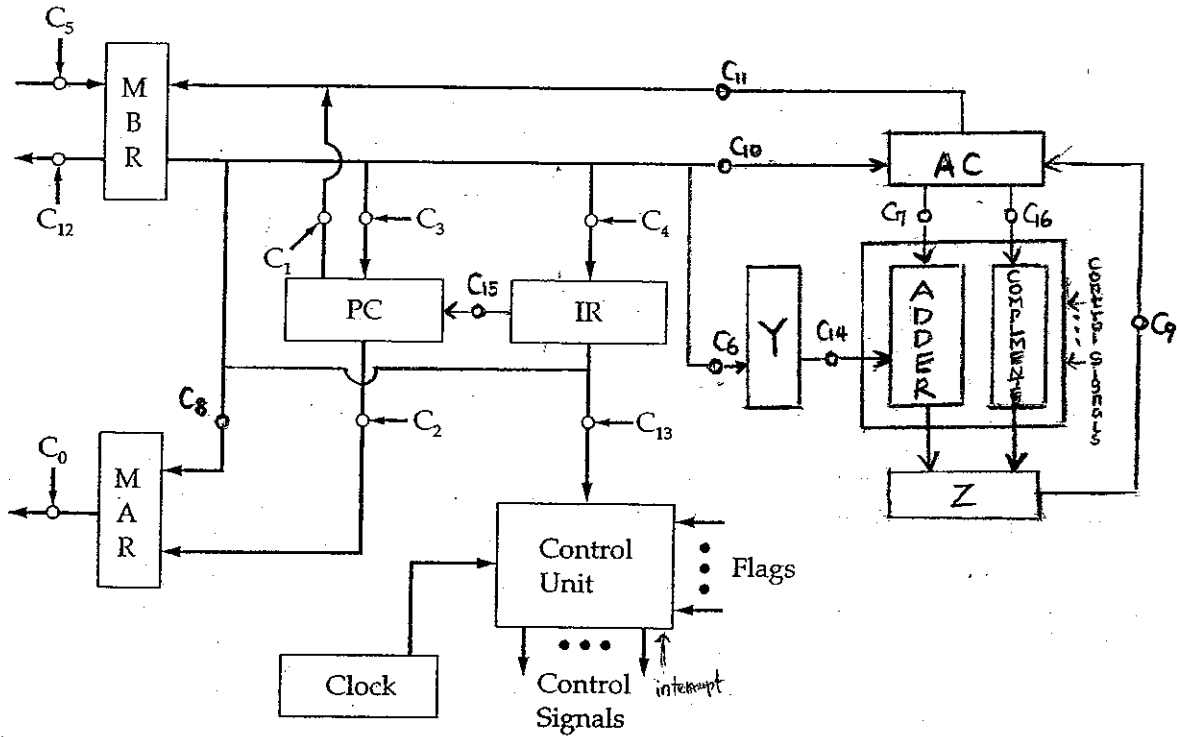


FIGURE 16.5. Data paths and control signals

microprogramming (firmware)  
microinstruction

TABLE 16.1 Micro-operations and Control Signals

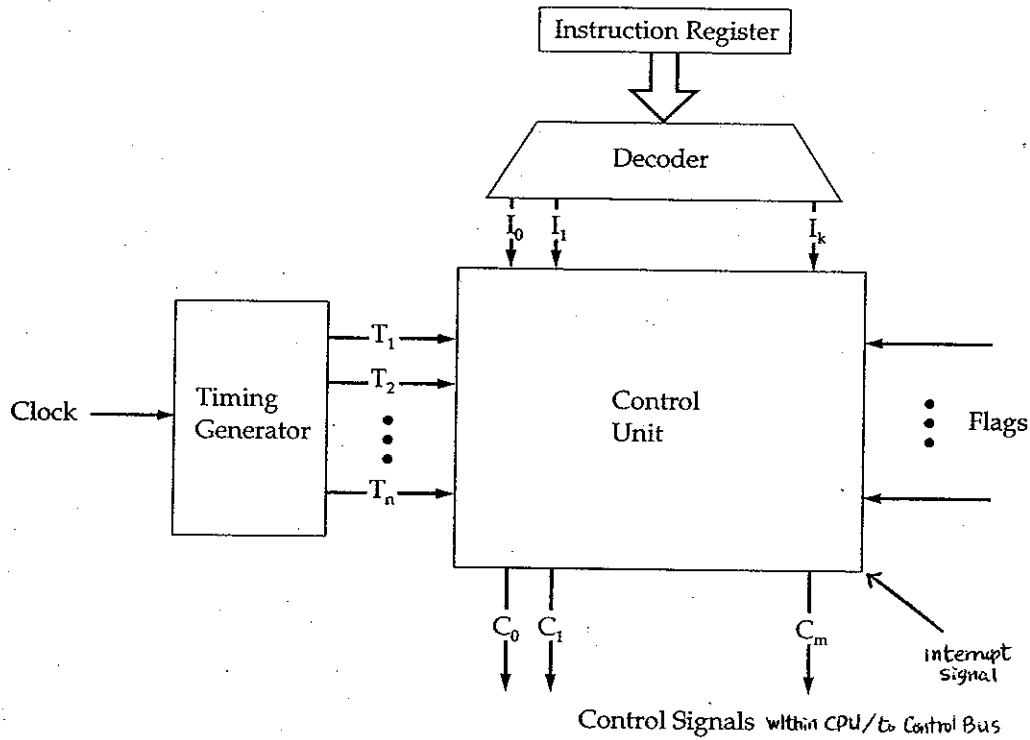
Micro-operations		Active Control Signals
Fetch:	$t_1: MAR \leftarrow (PC)$	$C_2$
	$t_2: MBR \leftarrow \text{Memory}$ $PC \leftarrow (PC) + 1$	$C_5, C_R$
	$t_3: IR \leftarrow (MBR)$	$C_4$
Indirect:	$t_1: MAR \leftarrow (IR(\text{Address}))$	$C_8$
	$t_2: MBR \leftarrow \text{Memory}$	$C_5, C_R$
	$t_3: IR(\text{Address}) \leftarrow (MBR(\text{Address}))$	$C_4$
Interrupt:	$t_1: MBR \leftarrow (PC)$	$C_1$
	$t_2: MAR \leftarrow \text{Save-address}$ $PC \leftarrow \text{Routine-address}$	not shown
	$t_3: \text{Memory} \leftarrow (MBR)$	$C_{12}, C_W$

$C_R$  = Read control signal to system bus.  
 $C_W$  = Write control signal to system bus.

control word

	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$	$C_6$	...	$C_R$	$C_W$	Imp resolution
Fetch $t_1$	0	1	0	0	0	0	...	0	0	
$t_2$	0	0	0	0	1	0	...	1	0	

## Hardwired Implementation



**FIGURE 16.4** Control unit with decoded inputs

*control signal*

PQ = 00	Fetch Cycle
PQ = 01	Indirect Cycle
PQ = 10	Execute Cycle
PQ = 11	Interrupt Cycle

$$C_5 = \bar{P} \cdot \bar{Q} \cdot T_2 + \bar{P} \cdot Q \cdot T_2 + P \cdot \bar{Q} \cdot (LDA + ADD + AND) \cdot T_2$$

Load Accumulator

t1: MAR	← (IR(address))	C8
t2: MBR	← Memory	C5, CR
t3: AC	← (MBR)	C10

Add to Accumulator

t1: MAR	← (IR(address))	C8
t2: MBR	← Memory	C5, CR
t3: Y	← (MBR)	C6
t4: Z	← (AC)+(Y)	C7, C14
t5: AC	← (Z)	C9

Branch

t1: PC	← (IR(address))	C15
--------	-----------------	-----