Control Sequence of Fetch Cycle

Step	Action
1	PCout,ARin,Read,Set carry in of ALU,clear y ,Add,Zin
2	Zout,PCin,WMFC
3	DRout,IRin

- ***Ex// Add R1,X***

Step	Action
4	Address of IRout, ARin, Read
5	R1out,yin,WMFC
6	DRout,Add,Zin
7	Zout,R1in
8	End

Execution of Branch Instructions

- A branch instruction replaces the contents of PC with the branch target address, which is usually obtained by adding an offset X given in the branch instruction.
- The offset X is usually the difference between the branch target address and the address immediately following the branch instruction.
- Conditional branch

Control Sequence of Execution Cycle

- ***Ex// Unconditional Branch***
- ***Jmp X***

Step	Action
4	PCout,yin
5	Address of IRout,Add,Zin
6	Zout,PCin
7	End

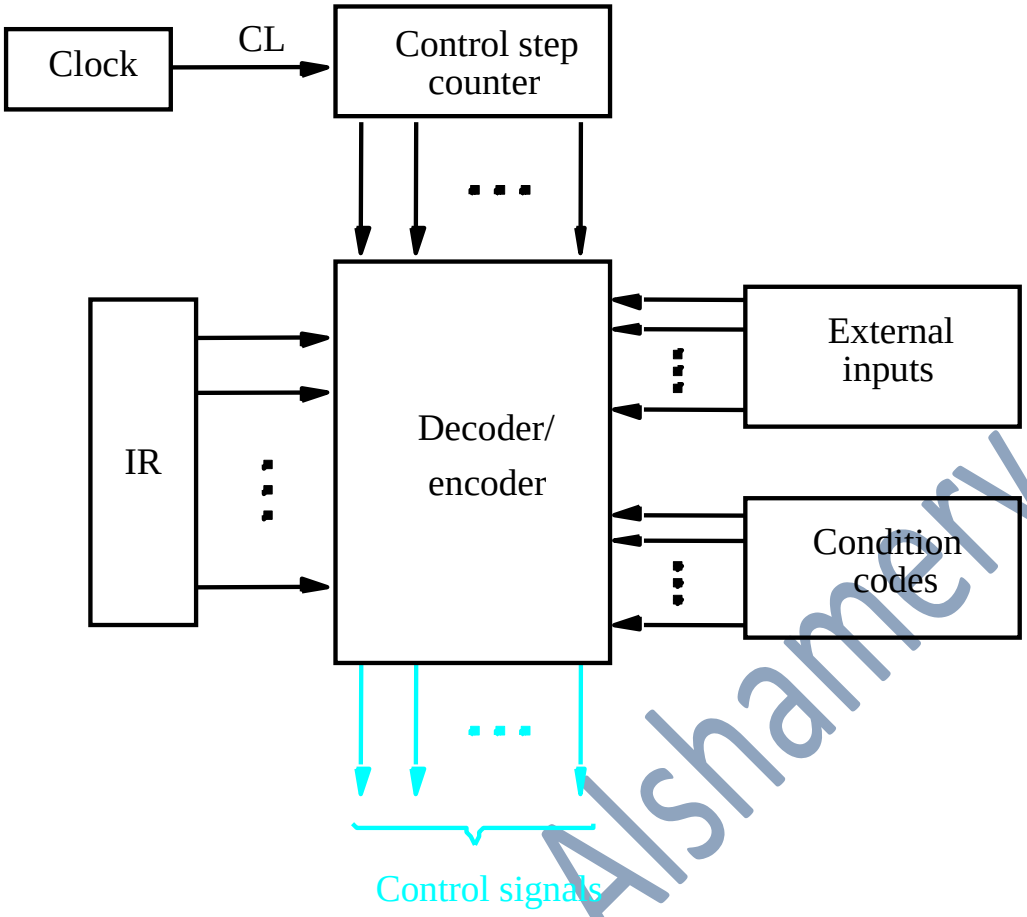
- ***Ex// Conditional Branch***

Step	Action
4	if (N bar) then End
5	PCout,yin
6	Address of IRout,Add,Zin
7	Zout,PCin
8	End

Microprogrammed Control

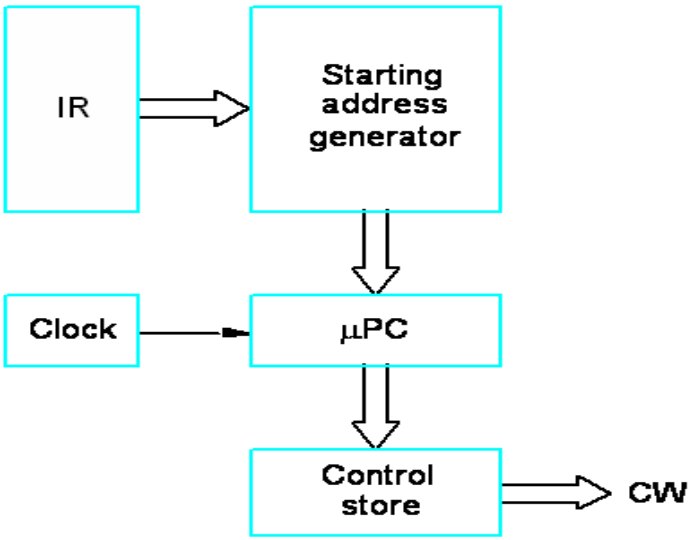
- An instruction is executed by carrying out a sequence of more rudimentary operations.
- All operations and data transfers are controlled by the processor clock.
- The ALU is a combinational circuit that has no internal storage.
- To execute instructions, the processor must have some means of generating the control signals needed in the proper sequence.
- Two categories: hardwired control and microprogrammed control
- Hardwired system can operate at high speed; but with little flexibility.

Control Unit Organization



- Control signals are generated by a program similar to machine language programs.
- Control Word (CW); microroutine; microinstruction

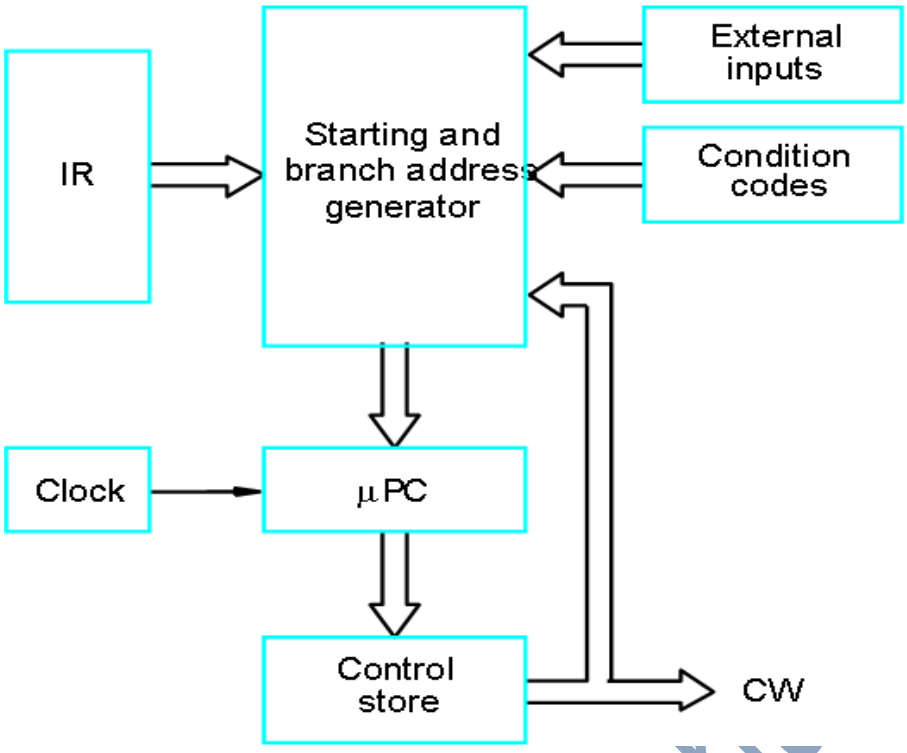
Micro-instruction	..	PC_{in}	PC_{out}	MAR_{in}	Read	MDR_{out}	IR_{in}	Y_{in}	Select	Add	Z_{in}	Z_{out}	$R1_{out}$	$R1_{in}$	$R3_{out}$	WMFC	End	:
1		0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	
2		1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	
3		0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	
4		0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	
5		0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	
6		0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	
7		0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	



Basic organization of a microprogrammed control

- The previous organization cannot handle the situation when the control unit is required to check the status of the condition codes or external inputs to choose between alternative courses of action.
- Use conditional branch microinstruction.

Organization of the control unit to allow conditional branching in the microprogram



Microinstructions

- A straightforward way to structure microinstructions is to assign one bit position to each control signal.
- However, this is very inefficient.
- The length can be reduced: most signals are not needed simultaneously, and many signals are mutually exclusive.
- All mutually exclusive signals are placed in the same group in binary coding.

Partial Format for the Microinstructions

Microinstruction

F1	F2	F3	F4	F5	

F1 (4 bits)	F2 (3 bits)	F3 (3 bits)	F4 (4 bits)	F5 (2 bits)
0000: No transfer 0001: PC_{out} 0010: MDR_{out} 0011: Z_{out} 0100: $R0_{out}$ 0101: $R1_{out}$ 0110: $R2_{out}$ 0111: $R3_{out}$ 1010: $TEMP_{out}$ 1011: $Offset_{out}$	000: No transfer 001: PC_{in} 010: IR_{in} 011: Z_{in} 100: $R0_{in}$ 101: $R1_{in}$ 110: $R2_{in}$ 111: $R3_{in}$	000: No transfer 001: MAR_{in} 010: MDR_{in} 011: $TEMP_{in}$ 100: Y_{in}	0000: Add 0001: Sub ⋮ 1111: XOR 16 ALU functions	00: No action 01: Read 10: Write

F6	F7	F8	...

F6 (1 bit)	F7 (1 bit)	F8 (1 bit)
0: SelectY 1: Select4	0: No action 1: WMFC	0: Continue 1: End

Figure . An example of a partial format for field-encoded microinstructions