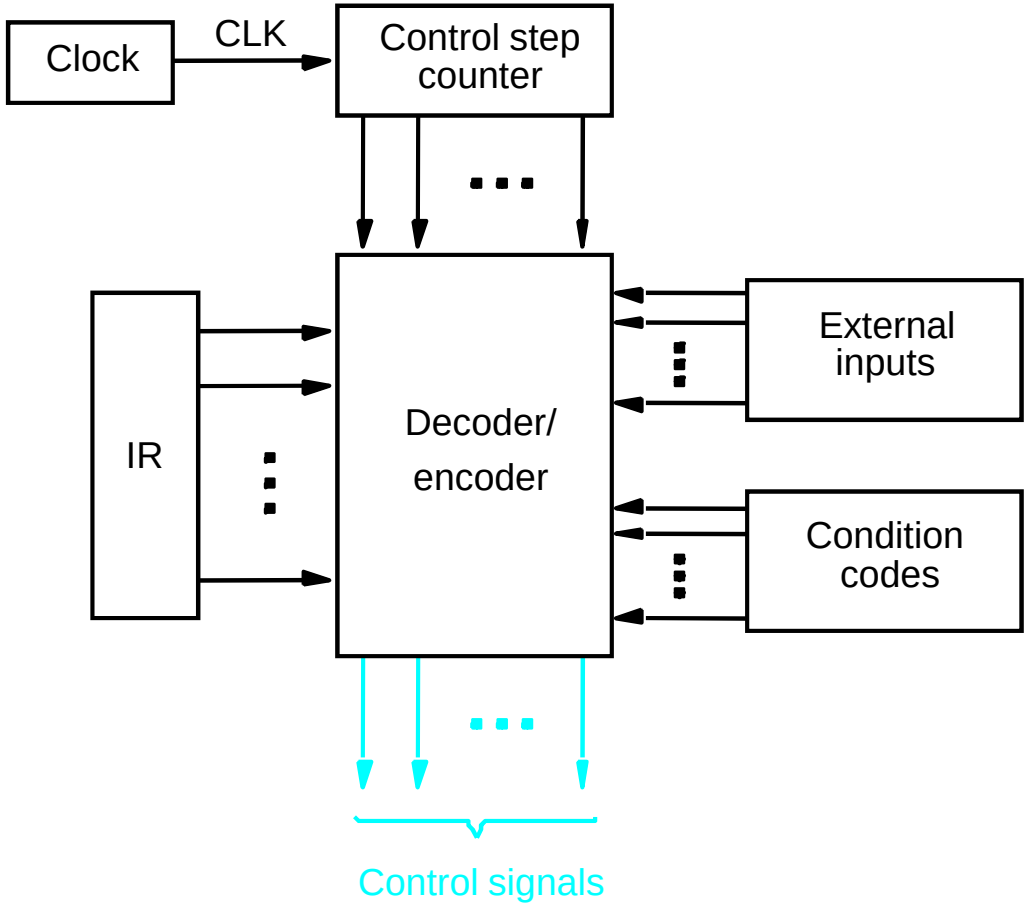
Hardwired Control Unit

Overview

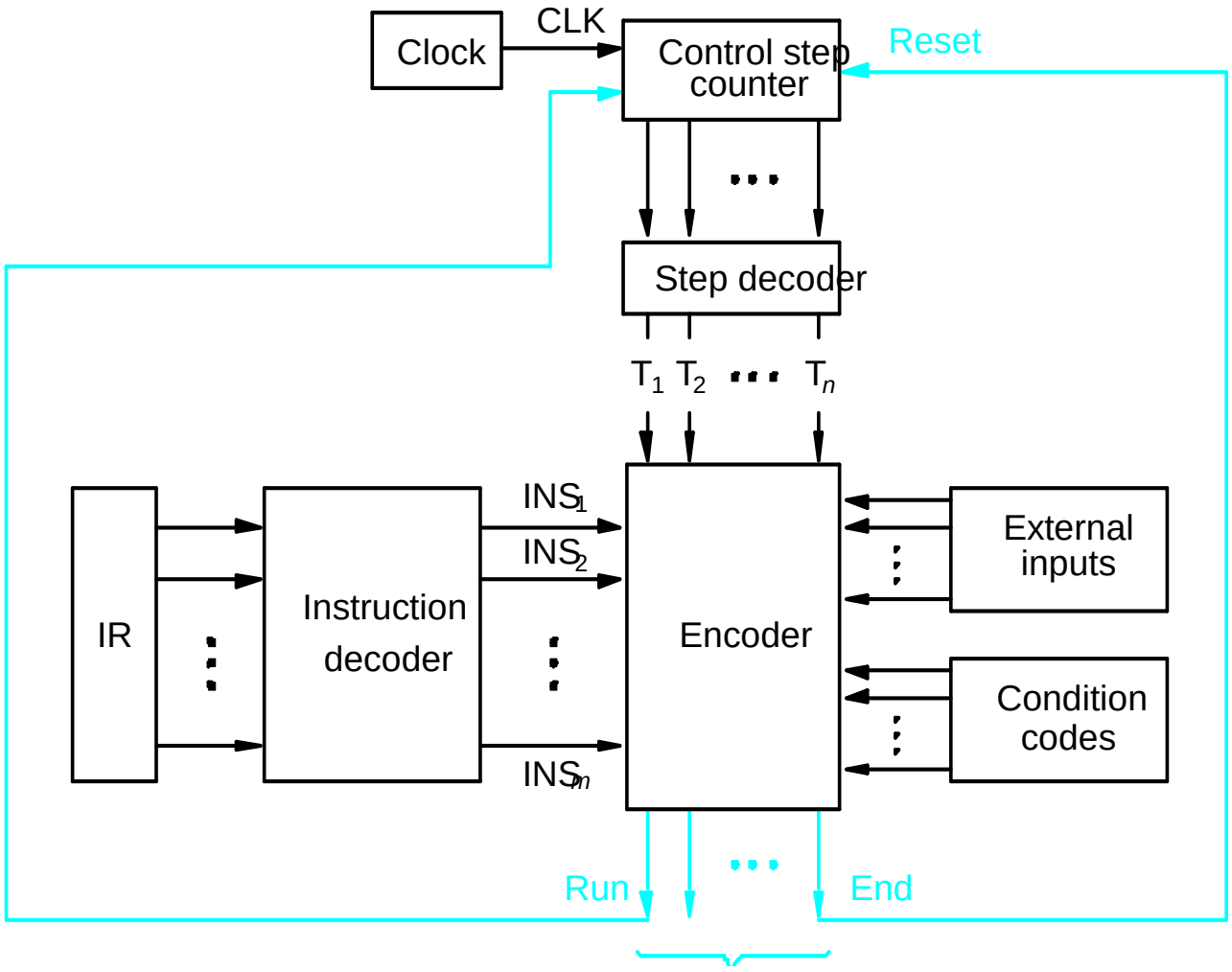
- To execute instructions, the processor must have some means of generating the control signals needed in the proper sequence.
- Two categories: hardwired control and microprogrammed control
- Hardwired system can operate at high speed; but with little flexibility.

Control Unit Organization



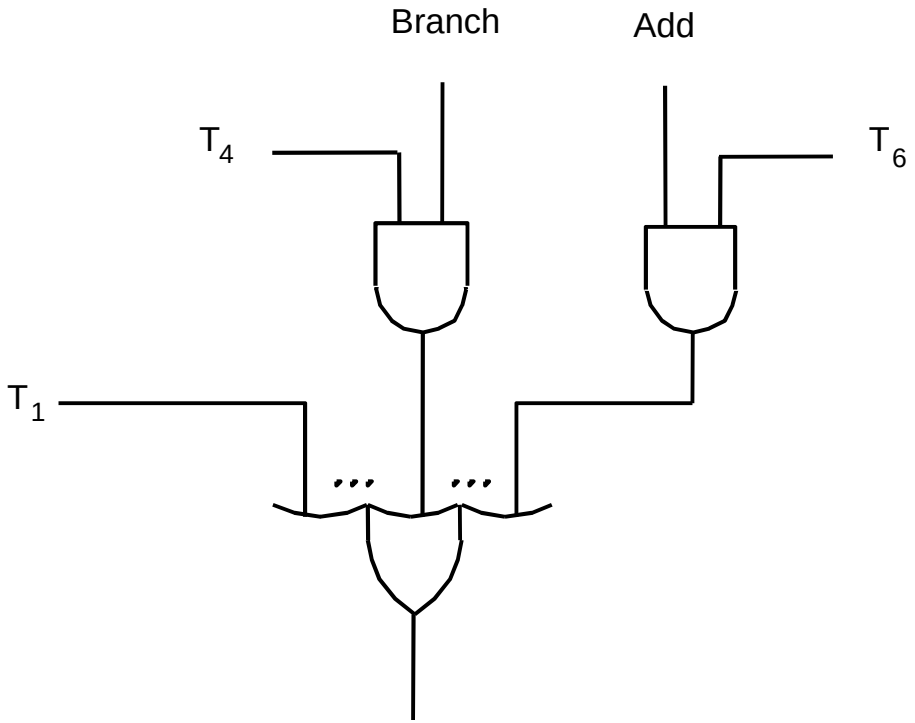
Control unit organization.

Detailed Block Description



Generating Z_{in}

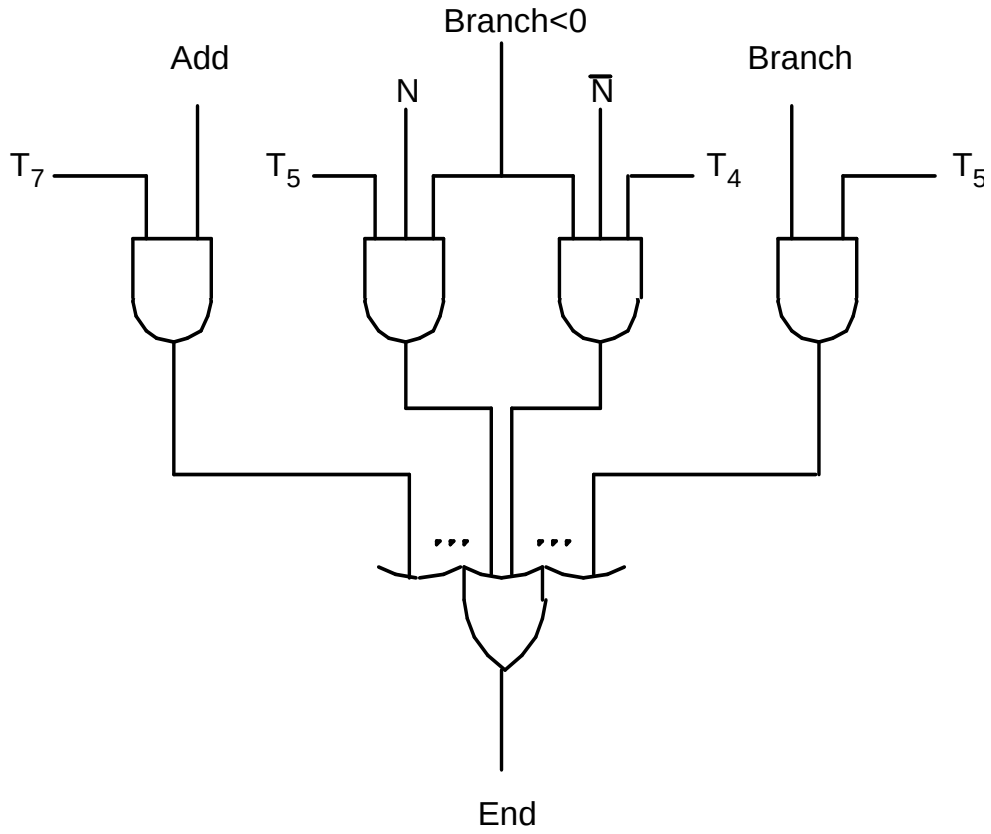
- $Z_{in} = T_1 + T_6 \bullet ADD + T_4 \bullet BR + \ldots$



Generation of the Z_{in} control signal for the processor in Figure 7.1.

Generating End

- End = $T_7 \bullet ADD + T_5 \bullet BR + (T_5 \bullet N + T_4 \bullet N) \bullet \overline{BR}N + \ldots$



Microprogrammed Control (1)

- To execute instructions, the CPU must have some means of generating the control signals in proper sequence.
- A ***control word*** (CW) is a word whose individual bits represent the various control signals.

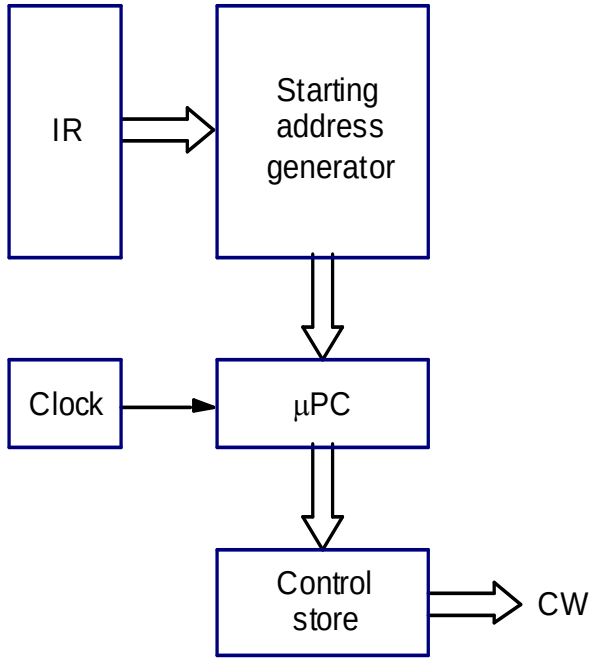
Micro - instruction	..	PC_{in}	PC_{out}	MAR_{in}	Read	MDR_{out}	IR_{in}	Y_{in}	Select	Add	Z_{in}	Z_{out}	$R1_{out}$	$R1_{in}$	$R3_{out}$	WMFC	End	..
1		0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	
2		1	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	
3		0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	
4		0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	
5		0	0	0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

Microprogrammed Control (2)

- A sequence of CWs corresponding to the control sequence of a machine instruction constitutes the ***microroutine***.
- The individual CW in the microroutine is referred to as ***microinstruction***.
- A ***control store*** stores the microroutines for all instructions.

Microprogrammed Control (3)

- Every time a new instruction is loaded into the IR, a “starting address generator” will output to a ***microprogram counter*** (μPC).
- The μPC is automatically incremented by the clock, causing successive microinstructions to be read from the control store.
- The control signals are delivered to various parts of the CPU in the correct sequence.



Basic Organization of a microprogammed control unit

Microinstructions (1)

- One way is to assign one bit position to each control signal (as in the previous example).
- Drawbacks
  - long microinstructions
  - poor use of bit space
- Solution
  - signals can be grouped so that all mutually exclusive signals are placed in the same group, e.g., read and write
  - thus, at most one ***microoperation*** per group is specified

Microinstructions (2)

Microinstruction

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

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

An example of a partial format for field-encoded microinstructions

Microinstructions (3)

- ***Vertical organization***: highly encoded schemes that use compact codes to specify only a small number of control functions in each microinstruction. Each meaningful combination of active control signals can be assigned a distinct code that represents the microinstruction.
- ***Horizontal organization***: minimally encoded scheme in which many resources can be controlled with a single instruction.