UNIT-III

(Lecture-17)

Techniques for Generating and Signals

- Hardwired Control
- Microprogrammed Control

Hardwired Control

- A hardwired control unit contain complex combinational logic circuits for sequencing through many micro-operations of the instruction cycle.

Control Sequence

- A counter is used to keep track of control steps
- Required control signals for each step are determined by
  - Contents of the control step counter
  - Contents of the instruction register
  - Contents of the condition code flags
  - External input signals, such as MFC and interrupt requests

Step	Action
1	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}
2	Z_{out}, PC_{in}, Y_{in}, WMFC
3	MDR_{out}, IR_{in}
4	$R3_{out}$, MAR_{in}, Read
5	$R1_{out}$, Y_{in}, WMFC
6	MDR_{out}, SelectY, Add, Z_{in}
7	Z_{out}, $R1_{in}$, End

Figure 7.6 Control sequence for execution of the instruction Add (R3),R1.

Control Unit Organization- Hardwired Control

- Encoder and Decoder blocks are combinational circuits that generates the required control outputs, depending on states of all its inputs.
- The step decoder provides a separate signal line for each step, or time slot, in the control sequence .
- the output of the instruction decoder consists of a separate line for each machine instruction.
- The input signals to the encoder block in Figure 7.11 are combined to generate the individual control signals Y_{in}, PC_{out}, Add, End, and so on.
- The End signal starts a new instruction fetch cycle by resetting the control step counter to its starting value .
- When set to 1, RUN causes the counter to be incremented by one at the end of every clock cycle. When RUN is equal to 0, the counter stops counting. This is needed whenever the WMFC signal is issued.

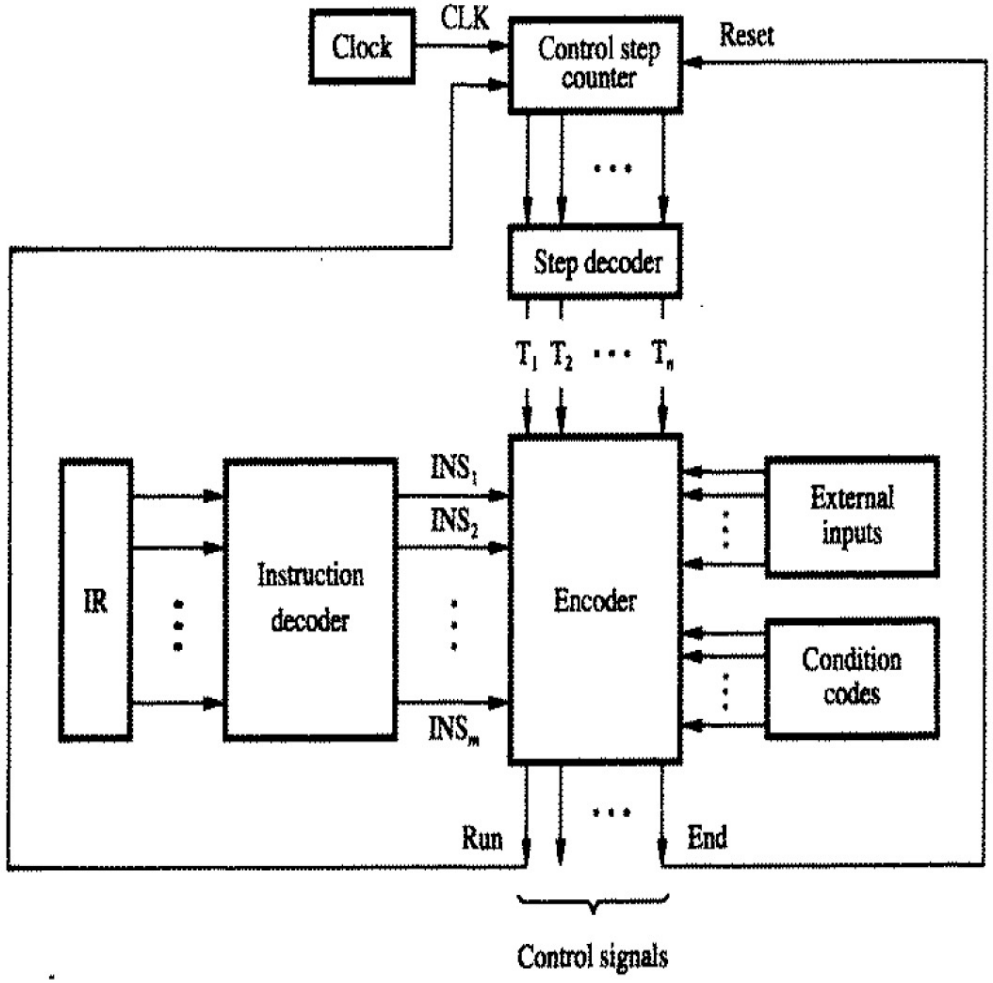


Figure 7.11 Separation of the decoding and encoding functions.

Example: How the encoder generates control signals Z_{in} and End

Step	Action
1	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}
2	Z_{out}, PC_{in}, Y_{in}, WMFC
3	MDR_{out}, IR_{in}
4	$R3_{out}$, MAR_{in}, Read
5	$R1_{out}$, Y_{in}, WMFC
6	MDR_{out}, SelectY, Add, Z_{in}
7	Z_{out}, $R1_{in}$, End

Figure 7.6 Control sequence for execution of the instruction Add (R3),R1.

Step	Action
1	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}
2	Z_{out}, PC_{in}, Y_{in}, WMFC
3	MDR_{out}, IR_{in}
4	Offset-field-of-IR_{out}, Add, Z_{in}
5	Z_{out}, PC_{in}, End

Figure 7.7 Control sequence for an unconditional Branch instruction.

Zin=T1+T6.ADD + T4.BR+---

End = T7.ADD + T5. BR + (T5 . N + T4 . $\overline{N}$) . BRN +---

Combinational Circuit for Z_{in} and End

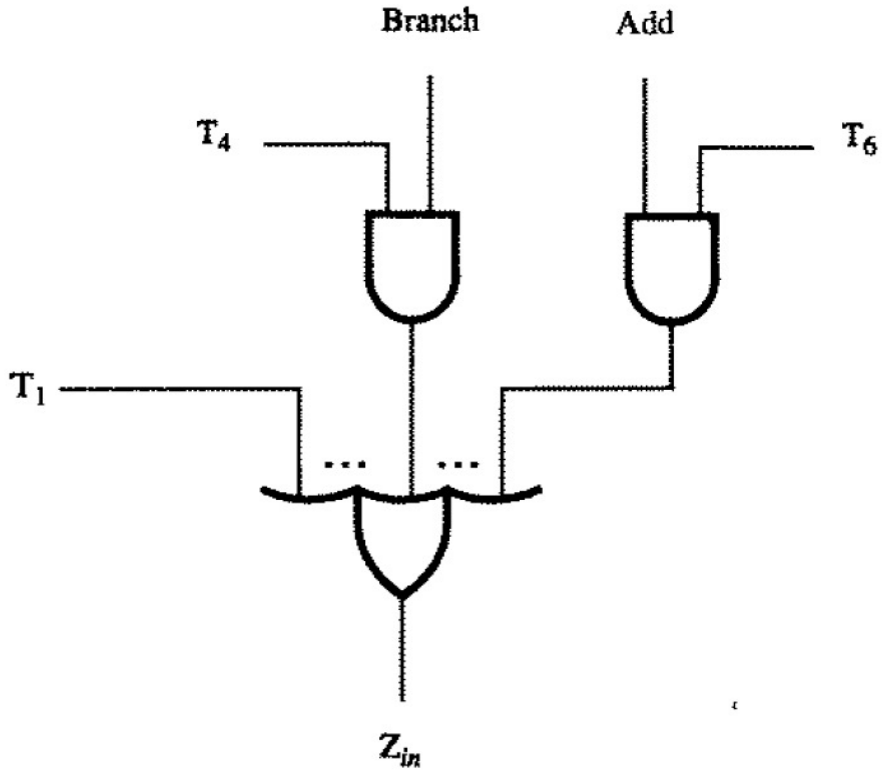


Figure 7.12 Generation of the Z_{in} control signal for the processor in Figure 7.1.

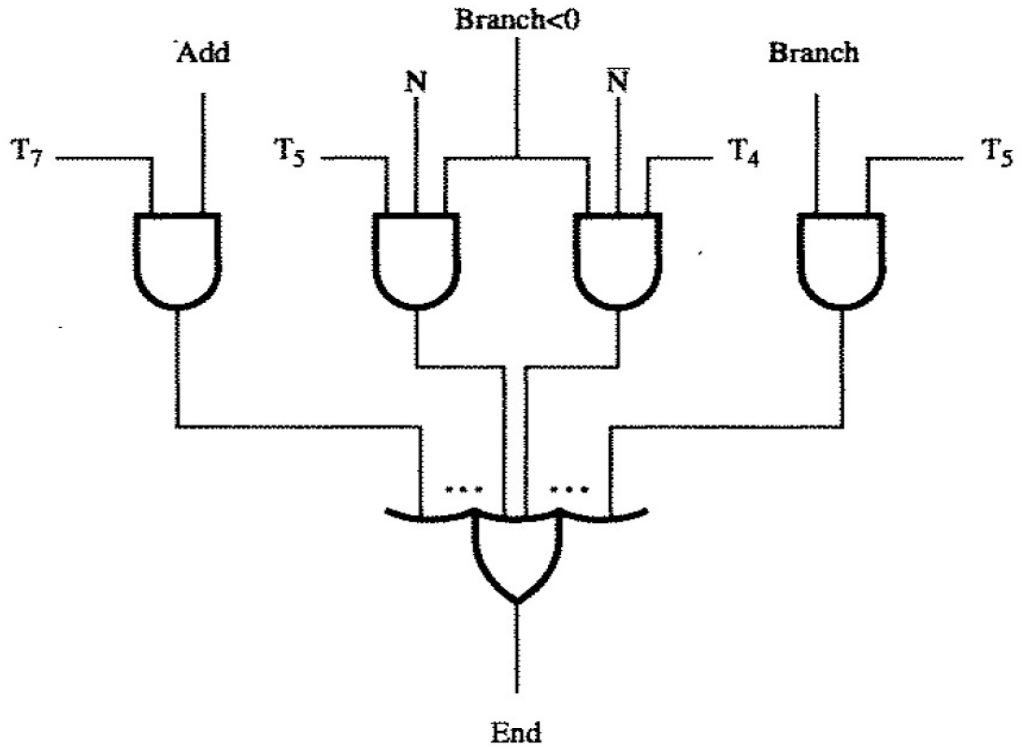


Figure 7.13 Generation of the End control signal.

Block Diagram of a Complete Processor

- It has separate processing units to deal with integer and floating-point data
- A processor may contain several units of each type to increase the potential for concurrency
- Instruction and data cache may be separate or combined

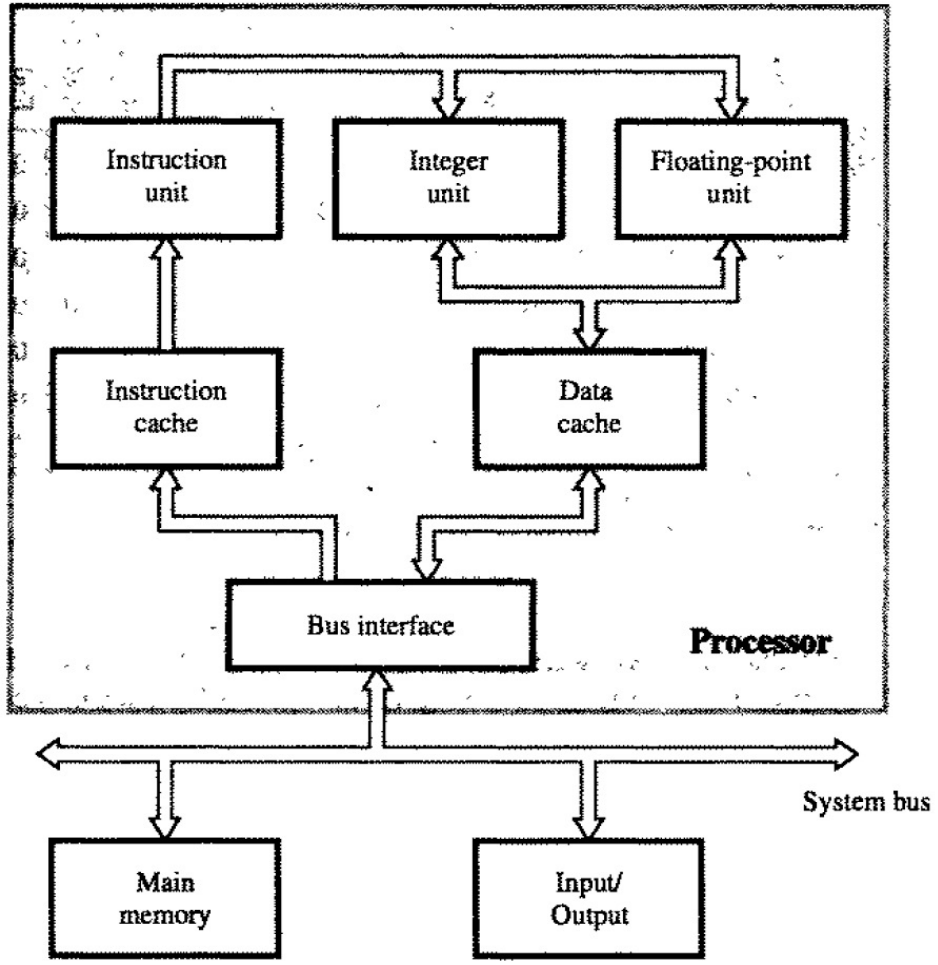
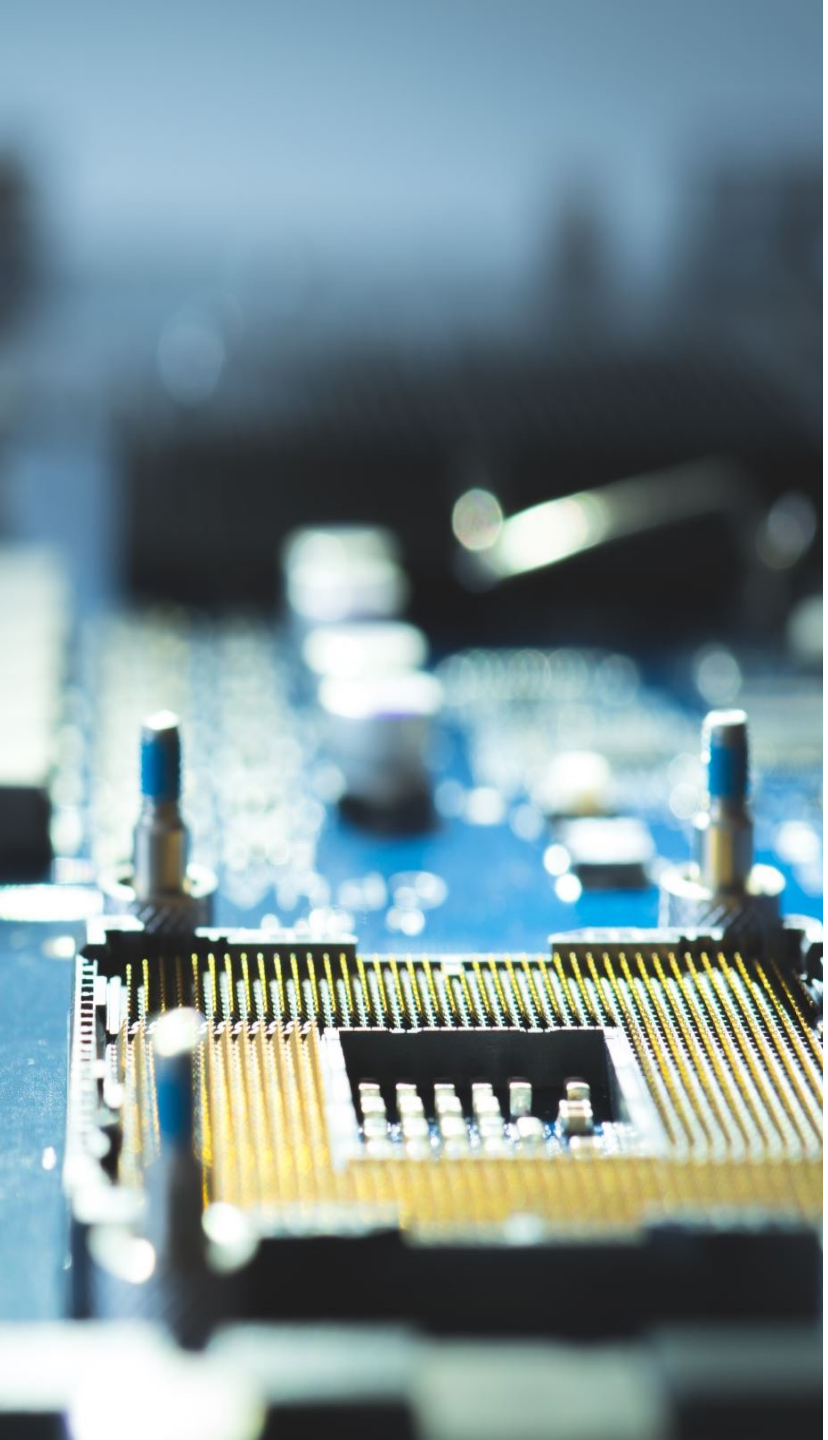


Figure 7.14 Block diagram of a complete processor.

Lecture 18

(UNIT-III)



Summary of Hardwired Control

- The control hardware shown can be viewed as a state machine that changes from one state to another in every clock cycle, depending on the contents of
  - the instruction register,
  - the condition codes, and
  - the external inputs.
- The outputs of the state machine are the control signals.
- The sequence of operations carried out by this machine is determined by the wiring of the logic elements, hence the name "hardwired."
- A controller that uses this approach can operate at high speed.
- However, it has little flexibility, and the complexity of the instruction set it can implement is limited.

Microprogrammed Control

In this scheme control signals are generated by a program similar to machine language program.

Some Common Terms:

Control Words (micro instructions)	**Micro routine** – set of micro instructions for a given machine instruction	**Control Store** – stores micro routines for all instructions in the instruction set	**A micro program counter** (µPC) - points to the next micro instructions in the micro routine

Control Word

Step	Action
1	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}
2	Z_{out}, PC_{in}, Y_{in}, WMFC
3	MDR_{out}, IR_{in}
4	$R3_{out}$, MAR_{in}, Read
5	$R1_{out}$, Y_{in}, WMFC
6	MDR_{out}, SelectY, Add, Z_{in}
7	Z_{out}, $R1_{in}$, End

Figure 7.6 Control sequence for execution of the instruction Add (R3),R1.

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	
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	

Figure 7.15 An example of microinstructions for Figure 7.6.

Basic Organization of Microprogrammed CU

- To read the control words sequentially from the control store, a micro program counter (µPC) is used.
- Every time a new instruction is loaded into the IR, **starting address generator** is loaded into the µPC.
- The µPC is then automatically incremented by the clock, causing successive microinstructions to be read from the **control store**.
- Hence, the control signals are delivered to various parts of the processor in the **correct sequence**.

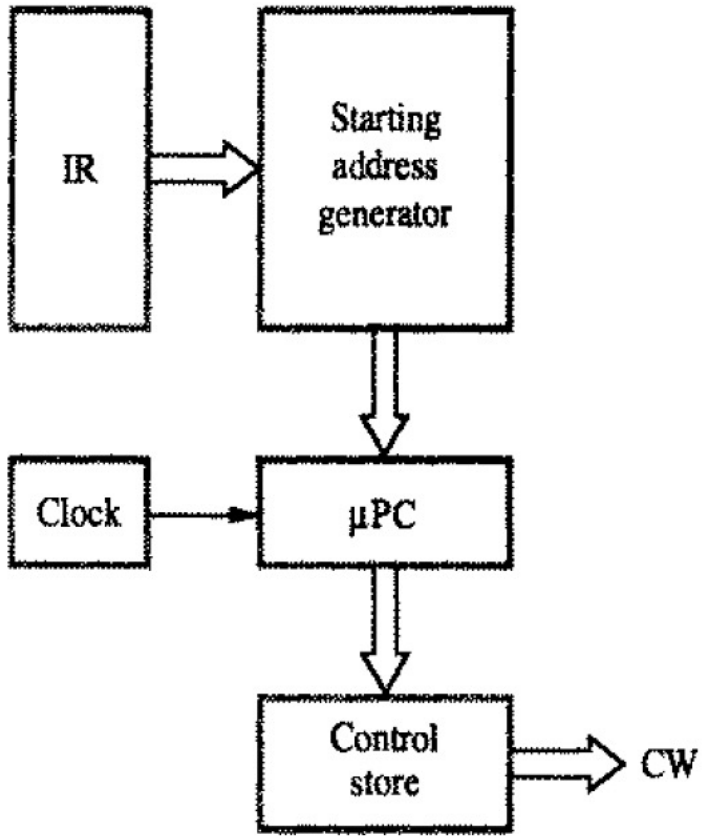


Figure 7.16 Basic organization of a microprogrammed control unit.

With Conditional Branching

- In this control unit, the μPC is incremented every time a new microinstruction is fetched from the micro program memory, except in the following situations:

  1. When a new instruction is loaded into the IR, the μPC is loaded with the starting address of the micro routine for that instruction.
  2. When a Branch microinstruction is encountered and the branch condition is satisfied, the μPC is loaded with the branch address.
  3. When an End microinstruction is encountered, the μPC is loaded with the address of the first CW in the micro routine for the instruction fetch cycle .

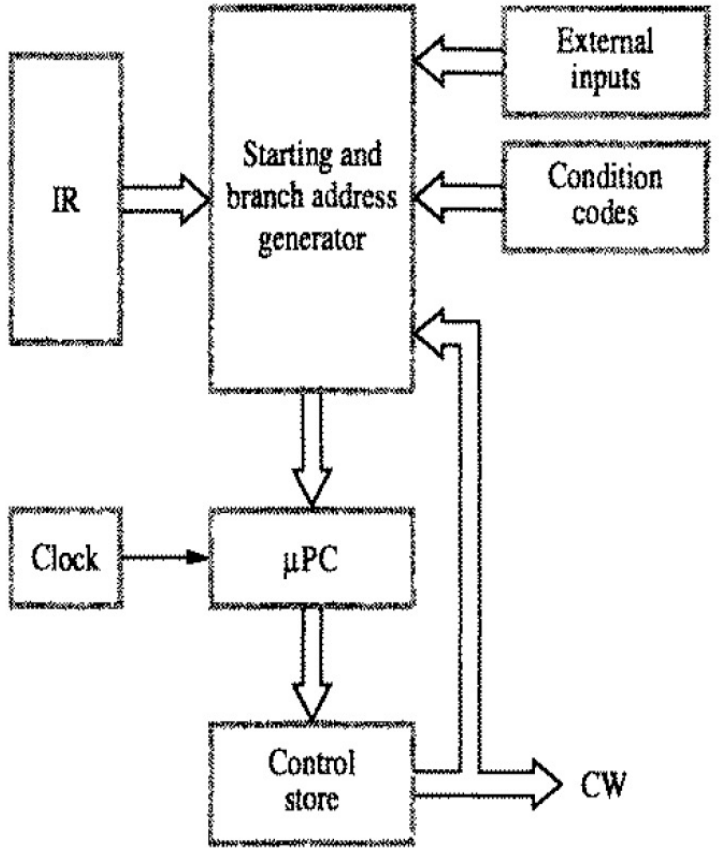


Figure 7.18 Organization of the control unit to allow conditional branching in the microprogram.

Micro routine for conditional branching

Address	Microinstruction
0	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}
1	Z_{out}, PC_{in}, Y_{in}, WMFC
2	MDR_{out}, IR_{in}
3	Branch to starting address of appropriate microroutine
...	...
25	If N=0, then branch to microinstruction 0
26	Offset-field-of-IR_{out}, SelectY, Add, Z_{in}
27	Z_{out}, PC_{in}, End

Microriutine for conditional branch instruction

Figure 7.17 Microroutine for the instruction Branch < 0.

Exercise: List the number of control signals for the given processor

With

- Single bus architecture as shown in Fig. 7.1
- 4 general purpose registers (R0 to R3) – gating signals required
- Special registers PC,MAR,MDR and IR - gating signals required
- Y and Z registers
- 16 ALU Functions (including ADD,SUB,XOR,AND,etc.)
- Other Control Signals like Read, Write, Select, WMFC and End
- Assume that some of the connections in this processor are permanently enabled, such as the output of the IR to the decoding circuits and both inputs to the ALU.