Control Unit

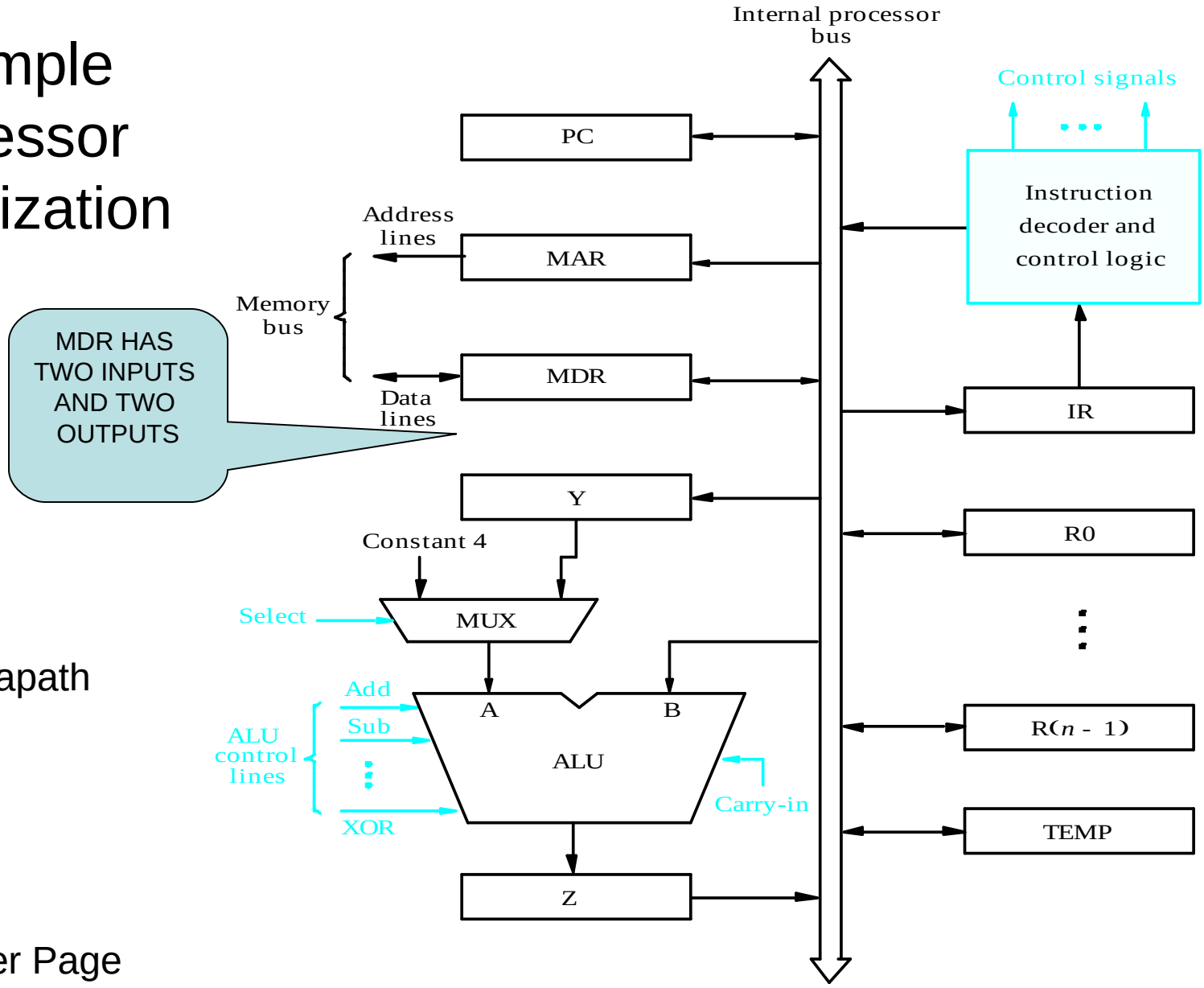
Hamacher:

Chapter 7.1,7.2,7.4,7.5

Basic Components of Processor

- ALU
- Control Unit
- Registers
- Internal bus
- External bus
  - The registers, ALU, and interconnecting bus are collectively referred to as **datapath**

A Simple Processor Organization



Datapath

Hamacher Page 413

Figure 7.1. Single-bus organization of the datapath inside a processor.

Register Transfers

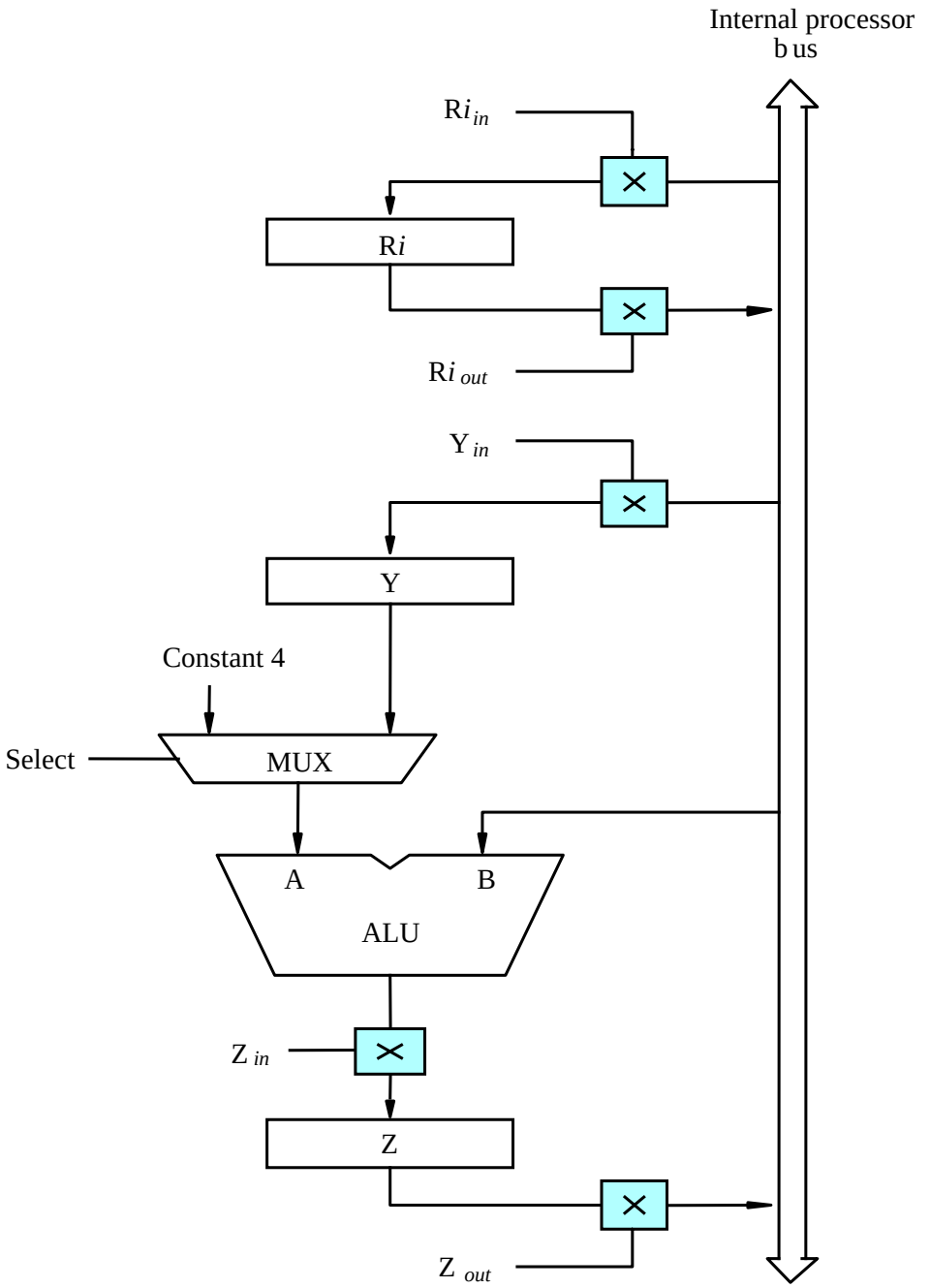
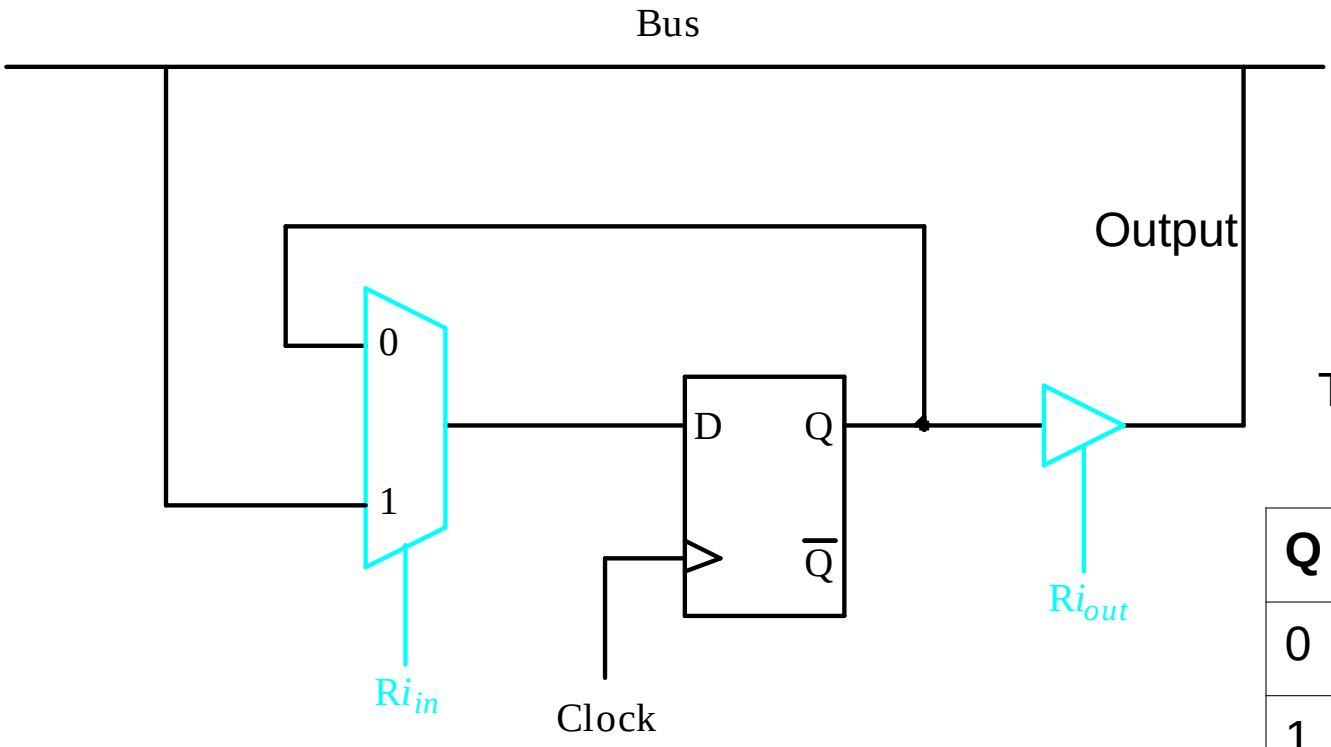


Figure 7.2. Input and output gating for the registers in Figure 7.1.

Register Transfers

- All operations and data transfers are controlled by the processor clock.



Tri-state Gate

Q	Riout	Output
0	0	High Impedence
1	0	
0	1	0
1	1	1

Figure 7.3. Input and output gating for one register bit.

t.

Fetching a Word from Memory

- Address into MAR; issue Read operation; WMFC

- data into MDR.

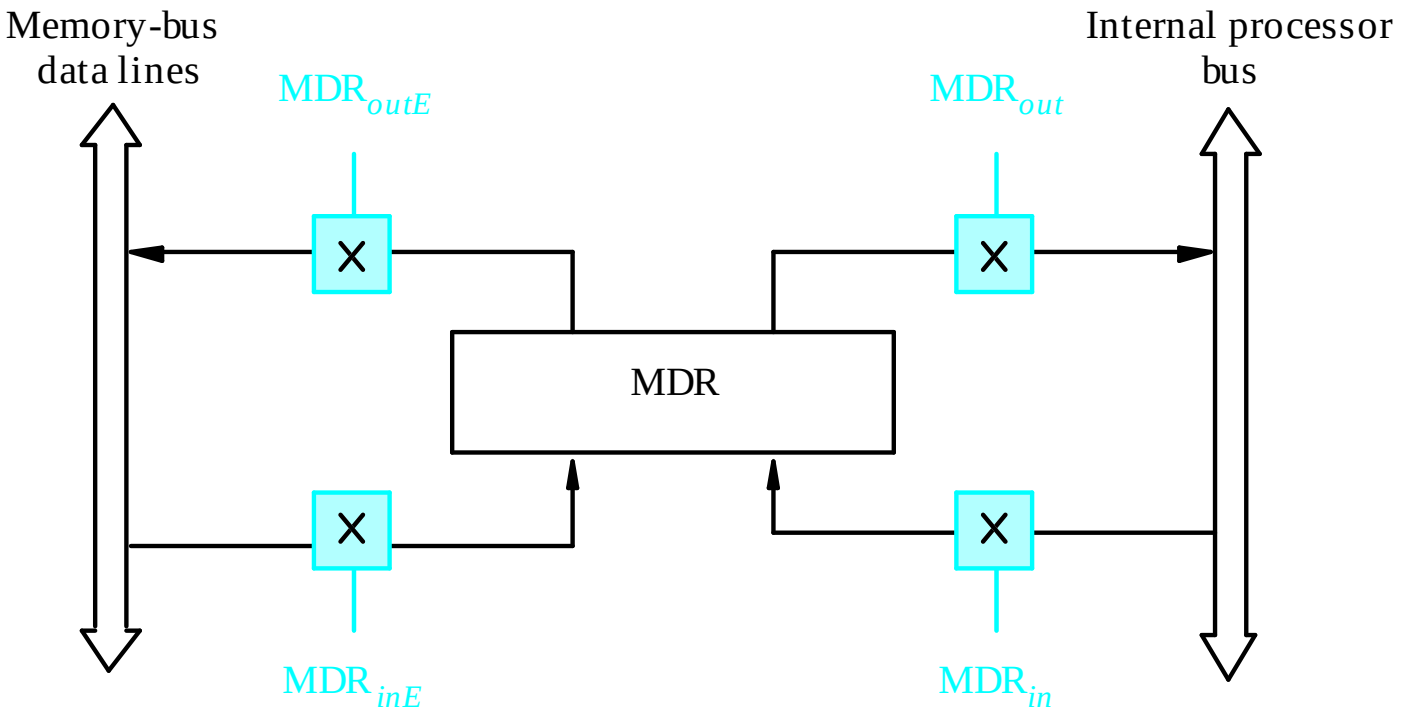


Figure 7.4. Connection and control signals for register MDR.

Functions of Components of Processor

- ALU
  - performs arithmetic and logic operations on data values stored in registers or memory
- CU (Control Unit)
  - generates all control signals to ensure the *sequence of all micro-operations (register/memory transfers/arithmetic or logic operations)* related to a machine code instruction
  - Activates the specific hardware units that are required for the set of operations that execution of a machine instruction requires

  i.e. opcode **decoding**

Register Transfers

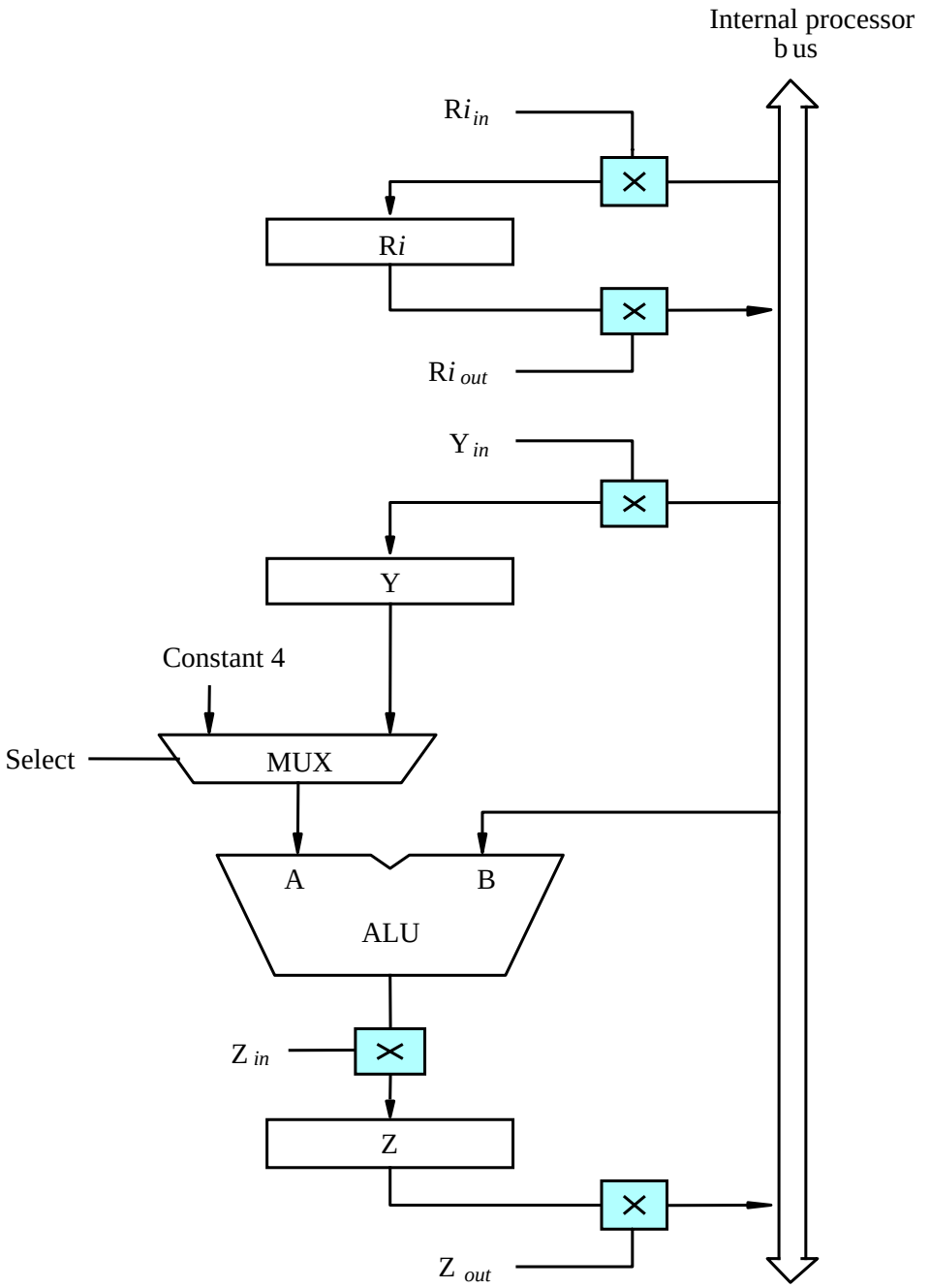


Figure 7.2. Input and output gating for the registers in Figure 7.1.

Micro-operation sequence and Control Signals for Fetching a Word from Memory

- The response time of each memory access varies (cache miss- memory-mapped I/O).
- To accommodate this, the processor waits(WMFC) until it receives from memory-a Memory-Function-Completed, MFC signal.

For Assembly code Instruction: MOV (R3),R1 (assuming R1 is destination

Time	**(ActionsNeeded) Micro-operations**	**Control Unit needs to generate following Control Signals**
T1:	MAR<- [R3]	R3out, MARin, Read (starts a read signal on memory bus)
T2:	Load MDR from memory bus when MFC received	MDRinE, WMFC
T2+n :	R1 <- [MDR]	MDRout, R1in

Register Transfers

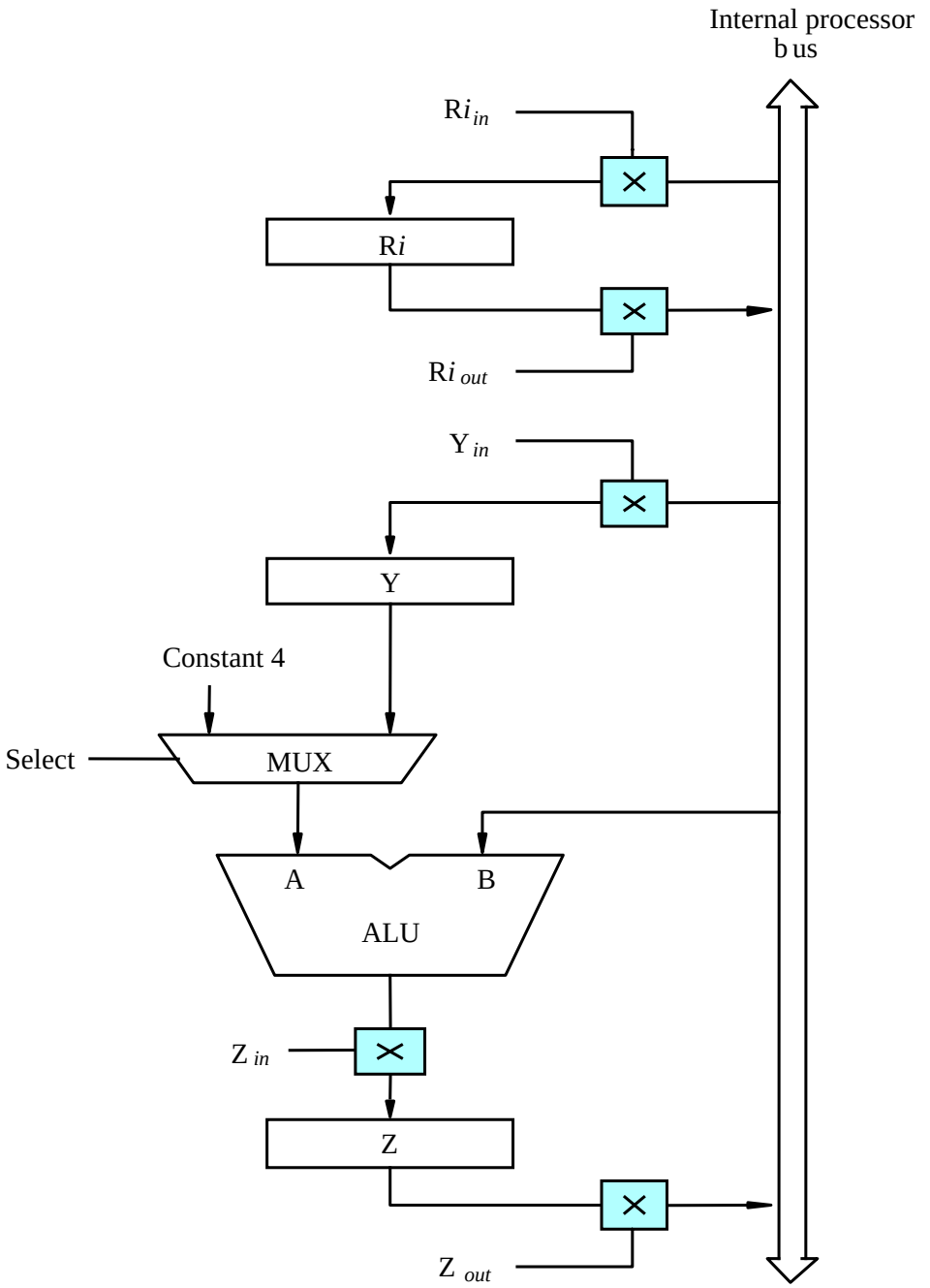


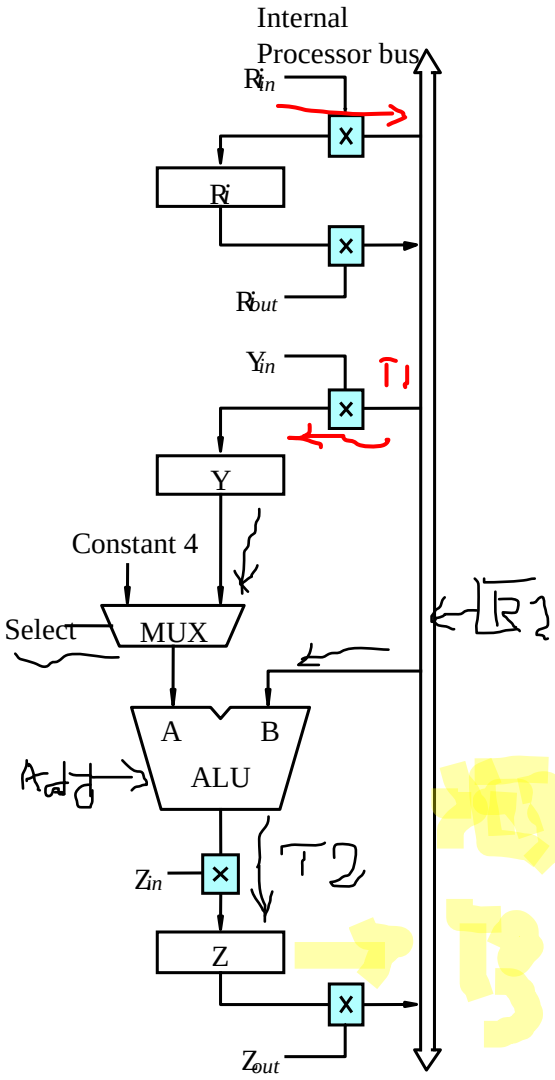
Figure 7.2. Input and output gating for the registers in Figure 7.1.

Performing an Arithmetic or Logic Operation

- The ALU is a combinational circuit that has no internal storage.
- ALU gets the two operands from MUX and bus. The result is temporarily stored in register Z.

For Assembly code ADD R1,R2,R3 //R3 destination

Time	(ActionsNeeded) Micro-operations	Control Signals Needed
T1:	Y<- R1	R1out, Yin
T2:	Z<- R2+ Y	R2out, SelectY, Add, Zin
T3:	R3 <- Z	Zout, R3in



Input and output gating for the registers in Figure 7.1.

Execution of another Assembly/M/C 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	$R3_{out}$, MAR_{in}, Read
5	$R1_{out}$, Y_{in}, WMFC
6	MDR_{out}, SelectY, Add, Z_{in}
7	Z_{out}, $R1_{in}$, End

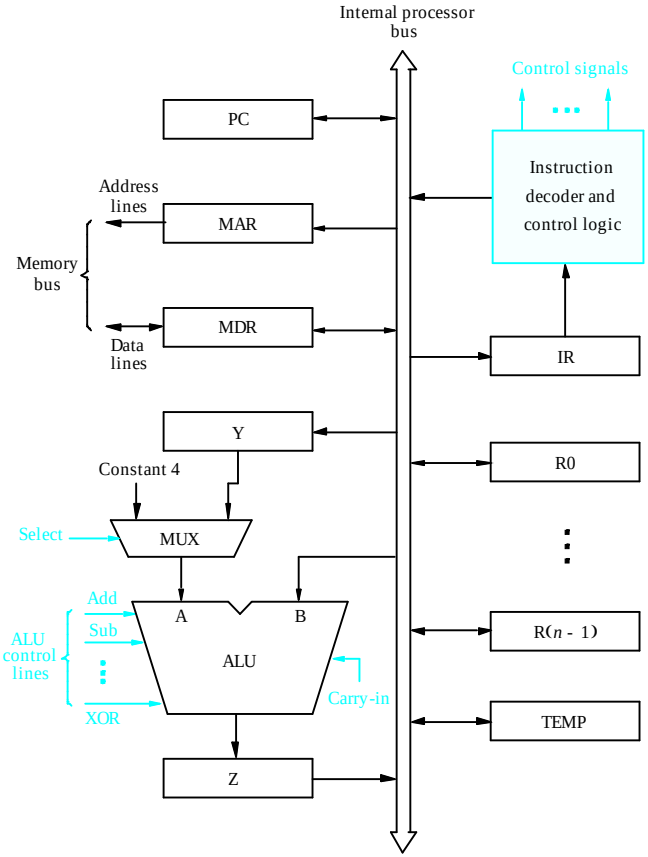


Figure 7.1. Single-bus organization of the datapath inside a processor.

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

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

Execution of Branch Instructions

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.

Quiz

- What is the control sequence for execution of the instruction

  Add R1, R2

  including the instruction fetch phase? (Assume single bus architecture)

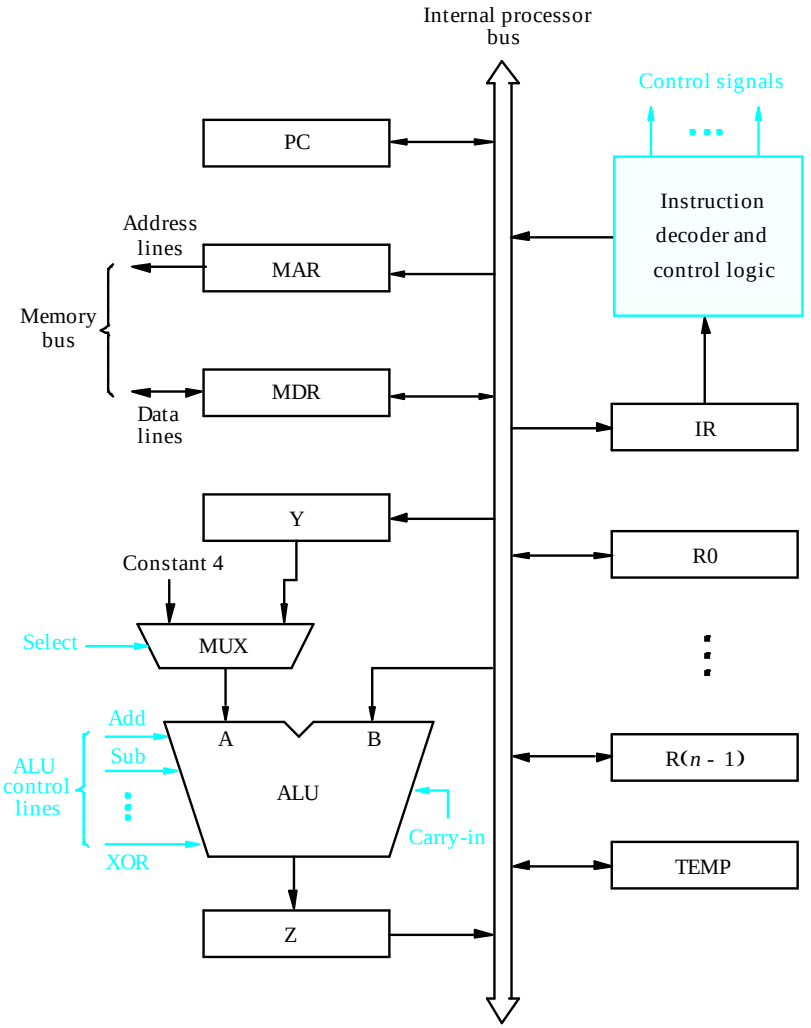


Figure 7.1. Single-bus organization of the datapath inside a processor.

States of a Simple Accumulator based Processor from Hayes Page 328

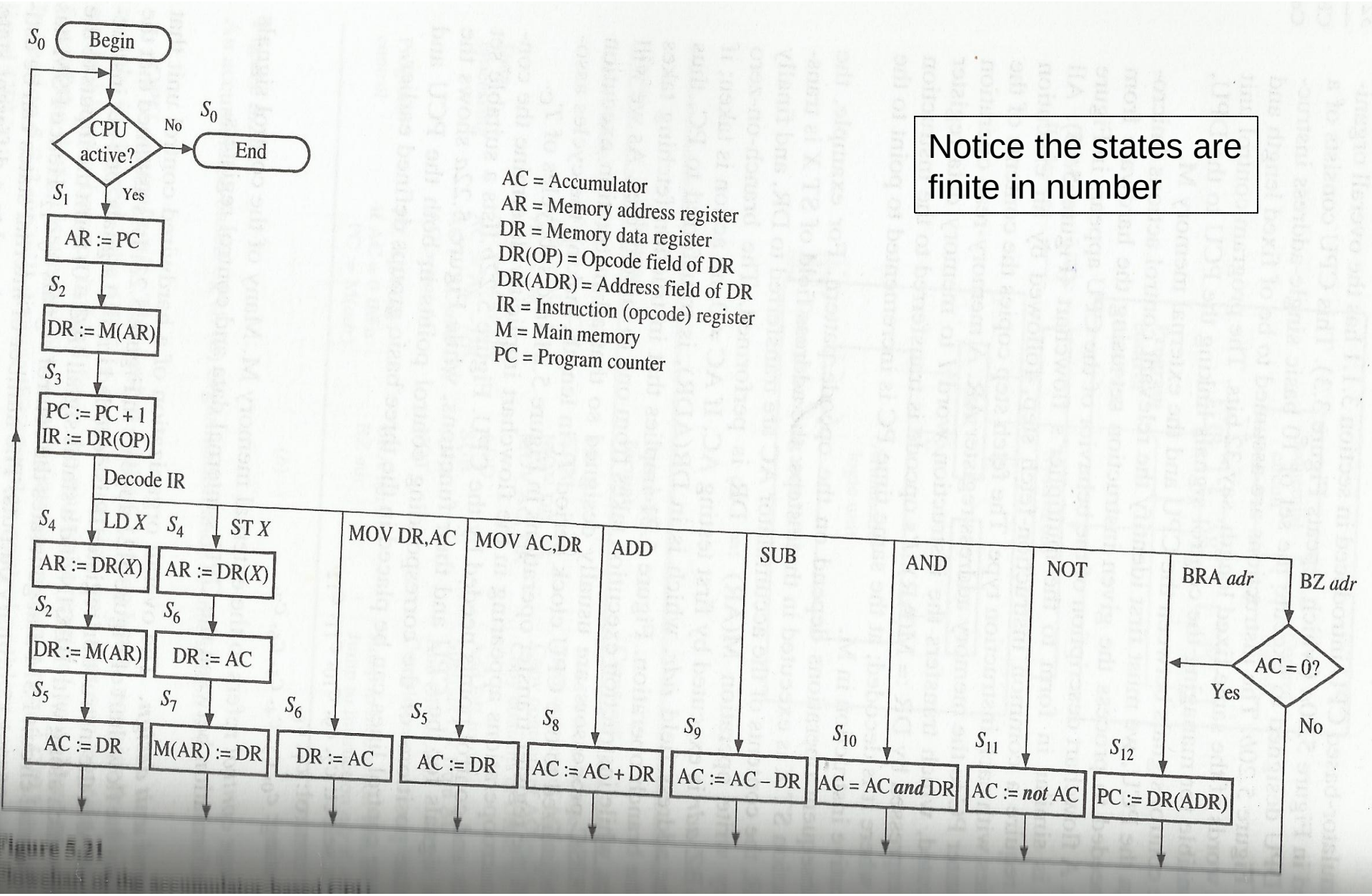


Figure 5.21
Flow chart of the accumulator based CPU

Hardwired Control

Processor (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

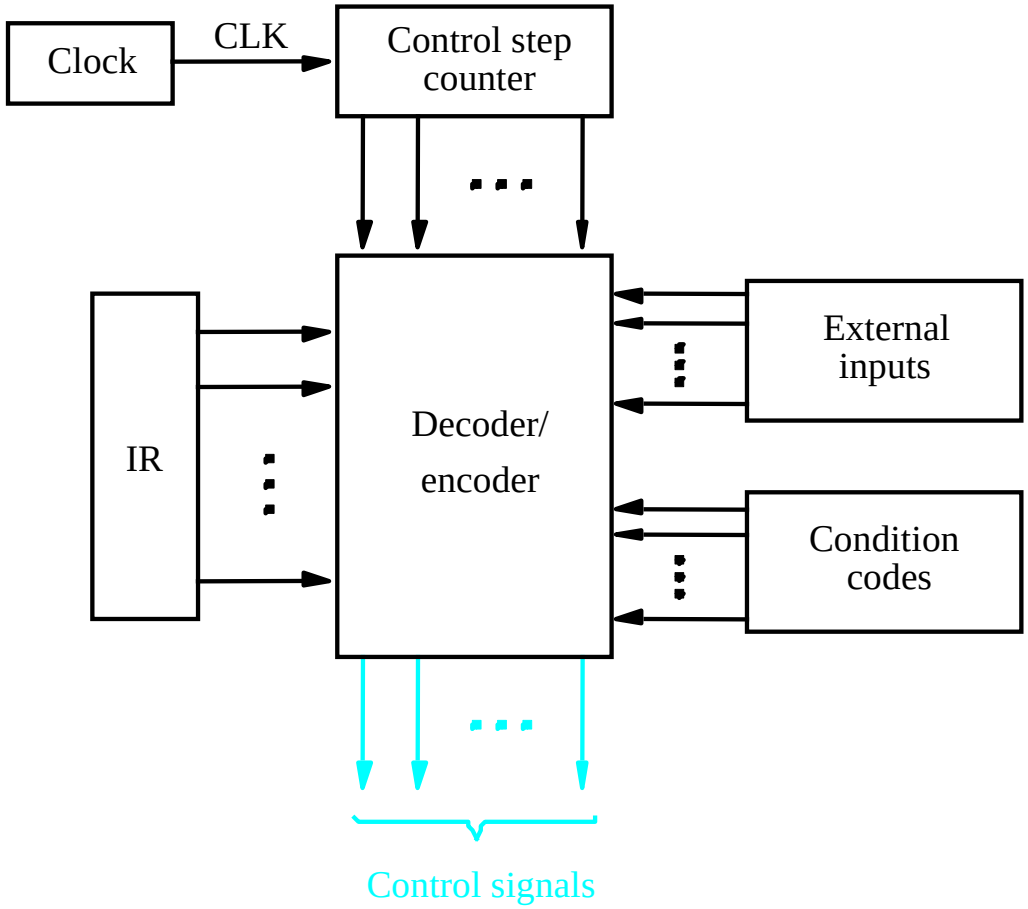


Figure 7.10. Control unit organization.

Detailed Block Description

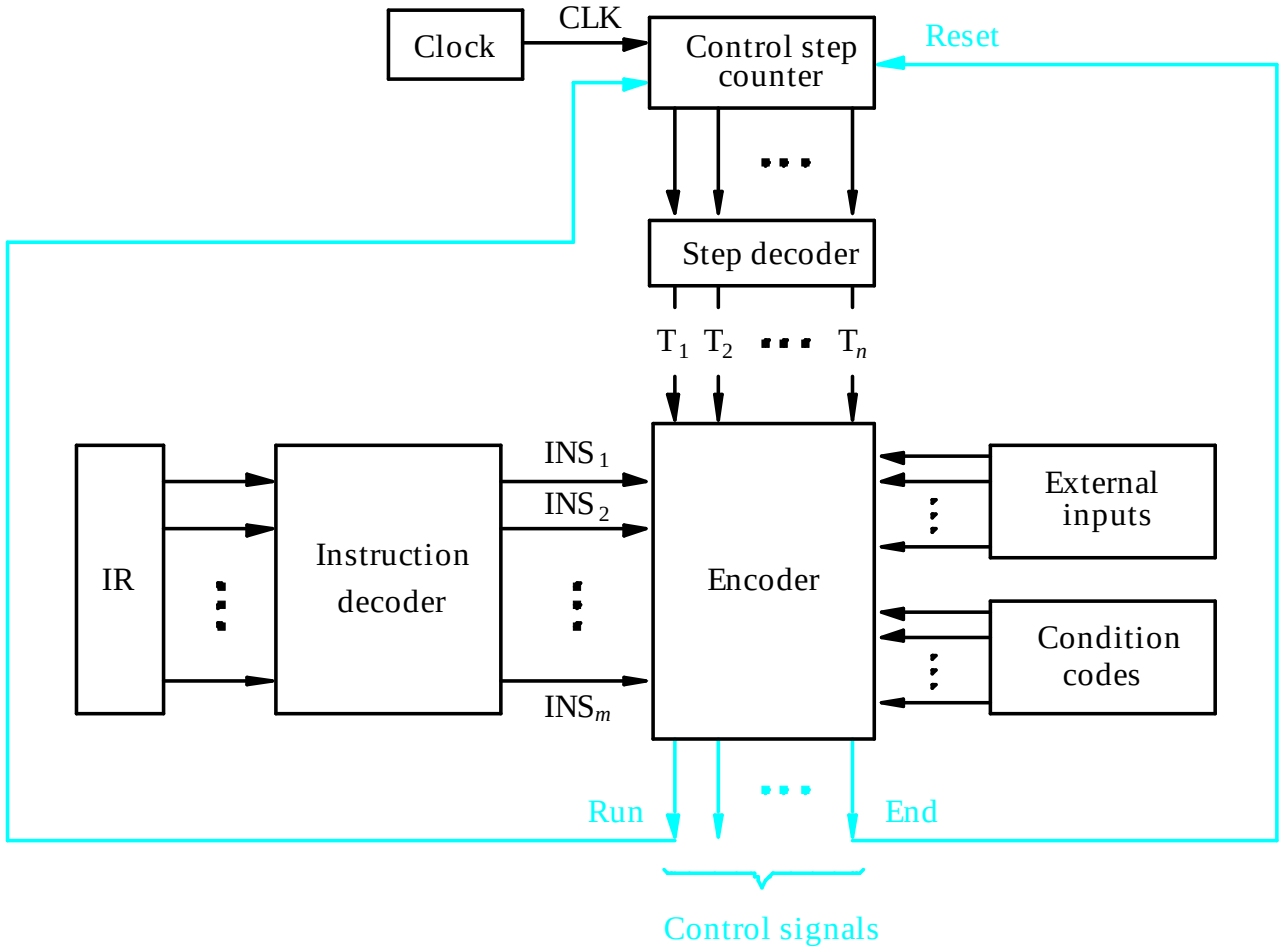


Figure 7.11. Separation of the decoding and encoding functions.

Hardwired Control Unit (for our simple example)

From Malvino

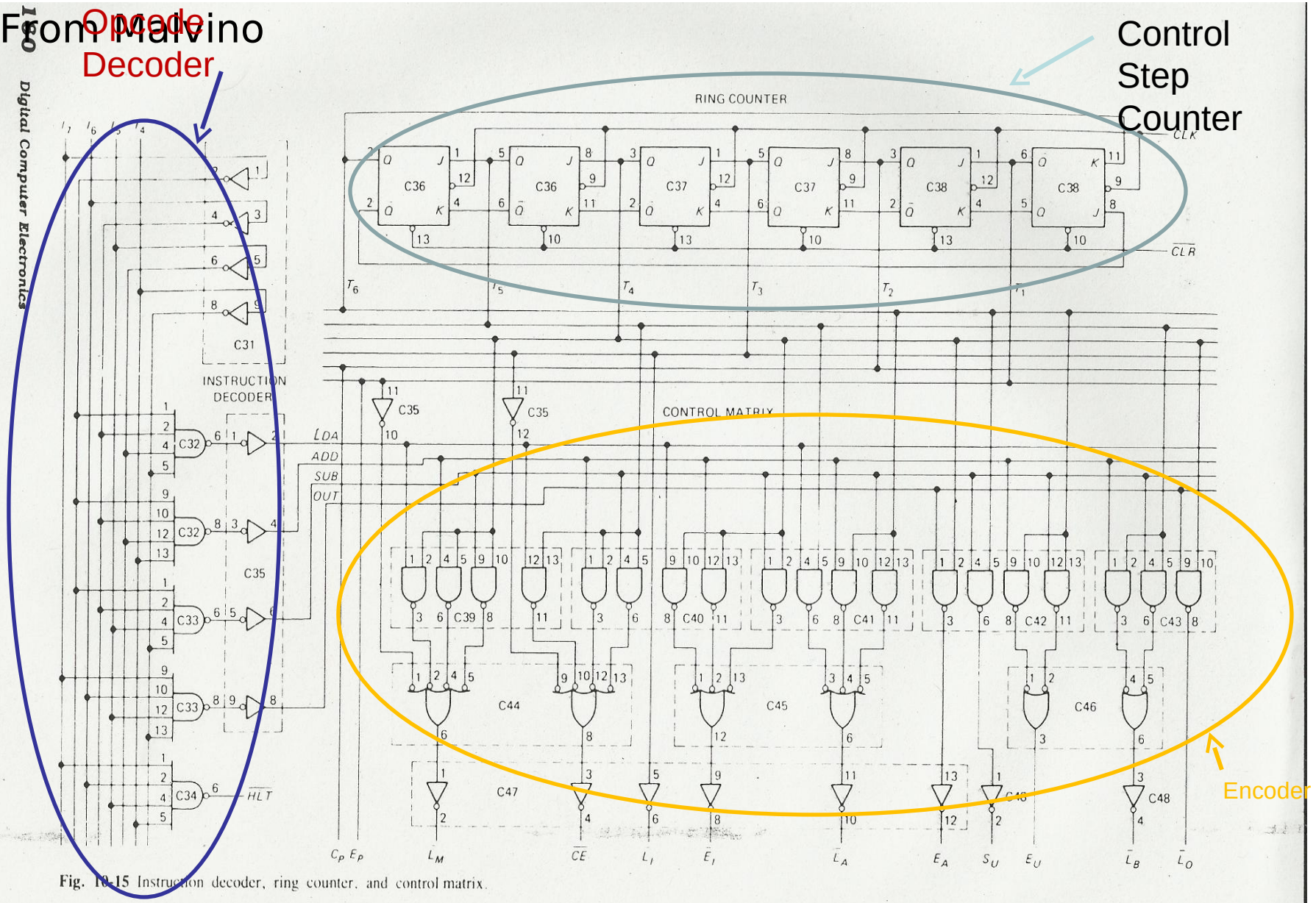
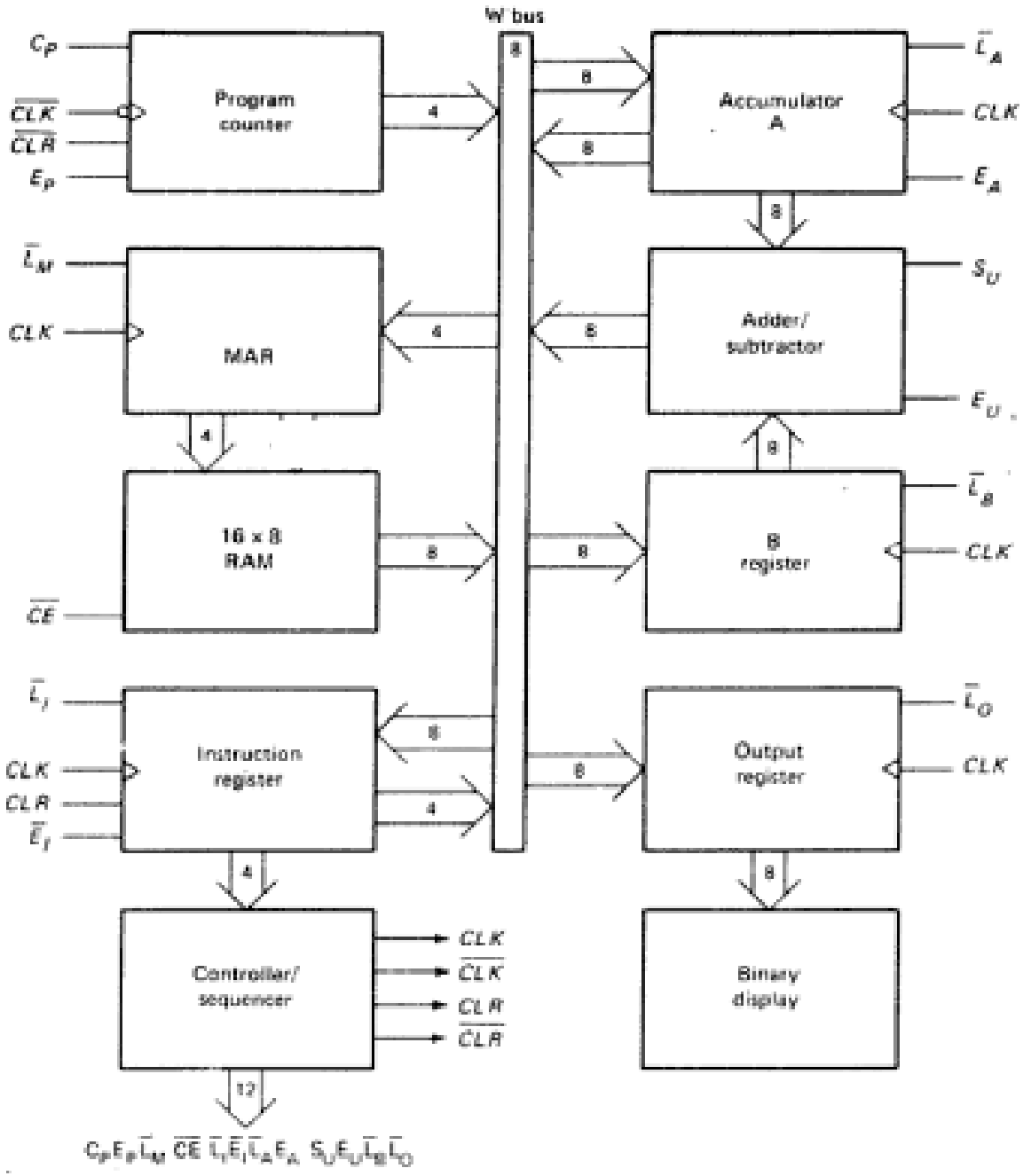


Fig. 10-15 Instruction decoder, ring counter, and control matrix.

From Malvino

No input from outside: It is assumed all instruction and data already stored in RAM.



- The above 2 slides are just to illustrate what goes into the blocks (shown in the detailed block descrition slide) for a very simple processor with Single bus Organization of Datapath
  - NOT for your exam purposes

Generating Z_{in}

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

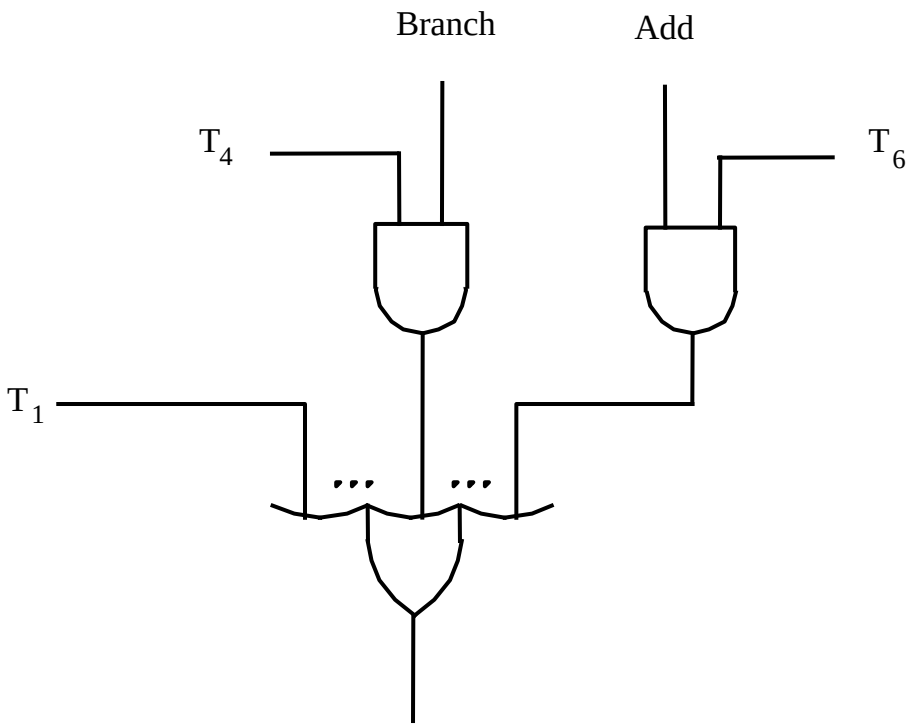


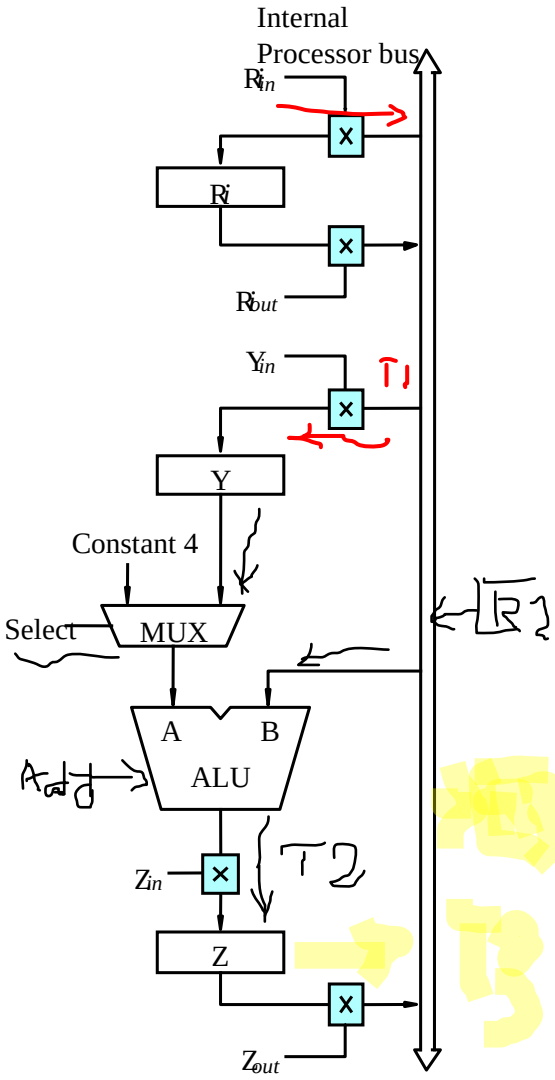
Figure 7.12. Generation of the Z_{in} control signal for the processor in Figure 7.1.

Performing an Arithmetic or Logic Operation

- The ALU is a combinational circuit that has no internal storage.
- ALU gets the two operands from MUX and bus. The result is temporarily stored in register Z.

For Assembly code ADD R1,R2,R3 //R3 destination

Time	**(ActionsNeeded) Micro-operations**	**Control Signals Needed**
Tn:	Y<- R1	R1out, Yin
Tn+1:	Z<- R2+ Y	R2out, SelectY, Add, Zin
Tn+2:	R3 <- Z	Zout, R3in



Input and output gating for the registers in Figure 7.1.

Generating End

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

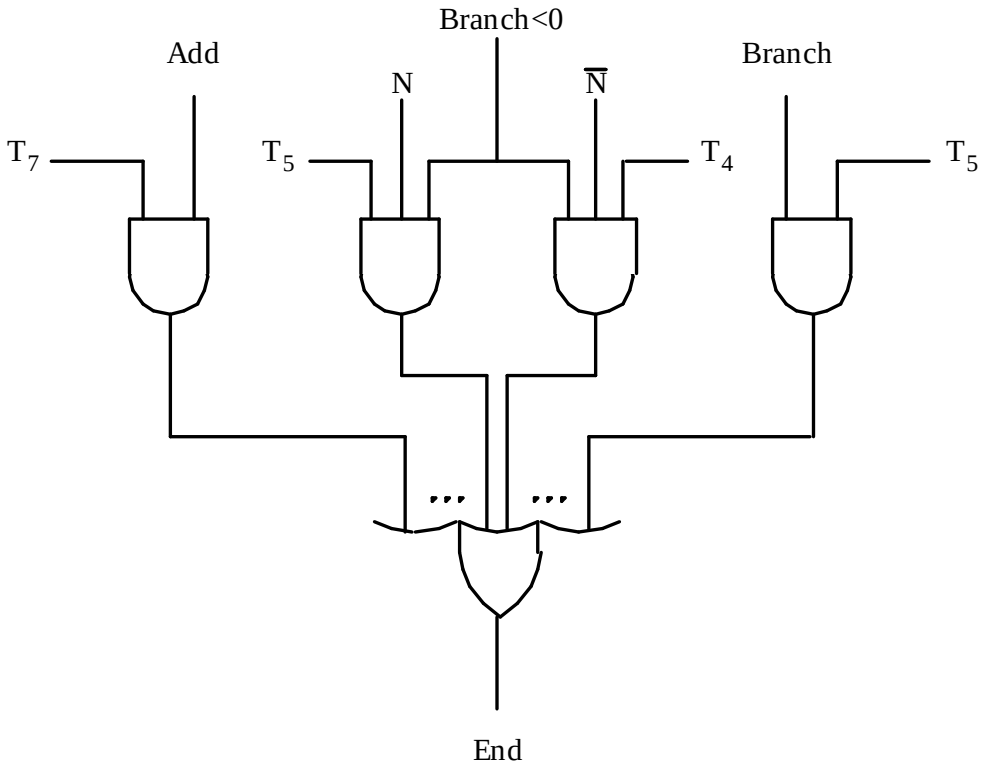


Figure 7.13. Generation of the End control signal.

States of a Simple Accumulator based Processor from Hayes Page 328

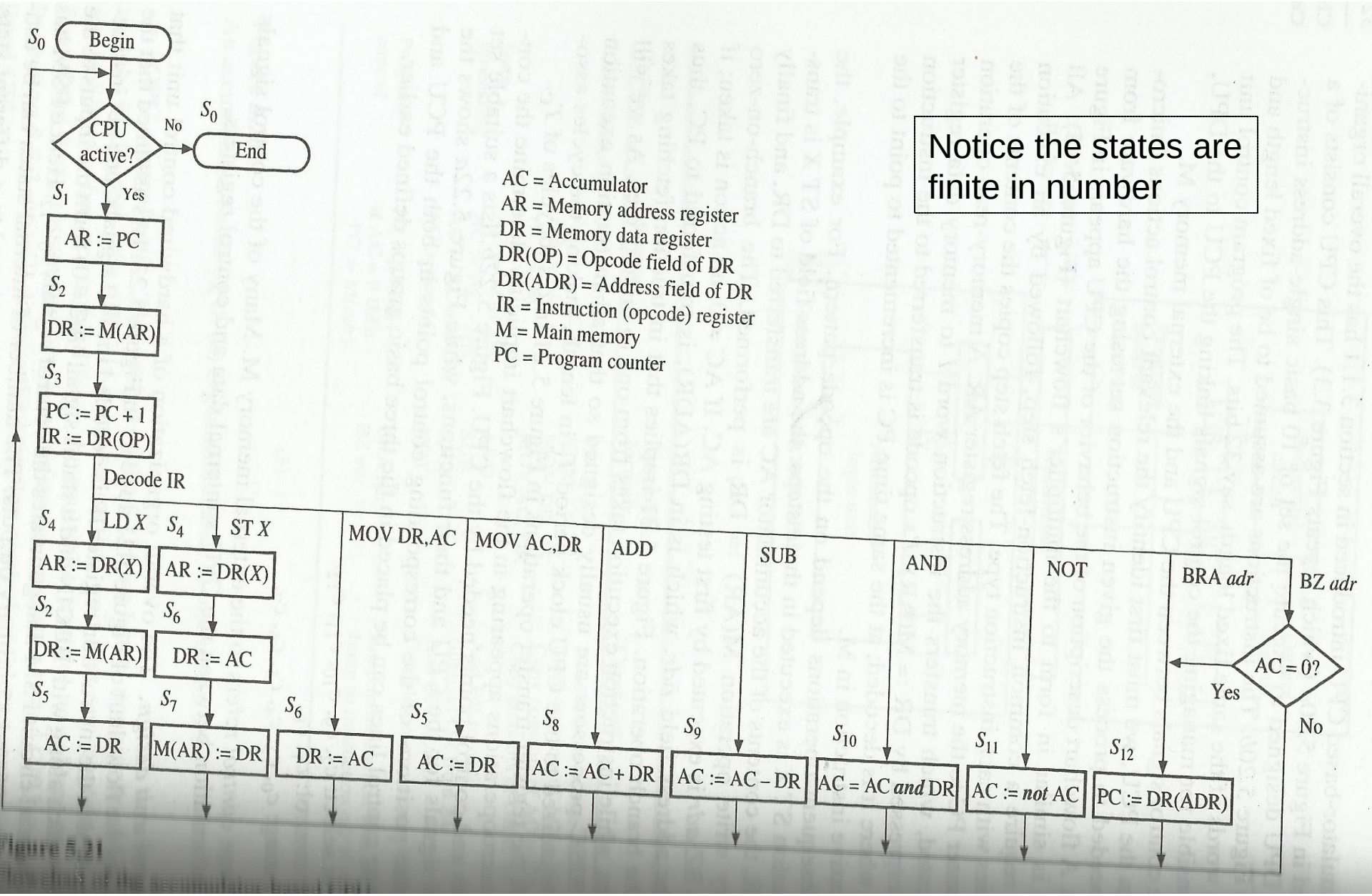


Figure 5.21
Flow chart of the accumulator based CPU

Describe how the processor generates the control signals needed in the proper sequence to execute instructions using following Detailed Block Description of Hardwire

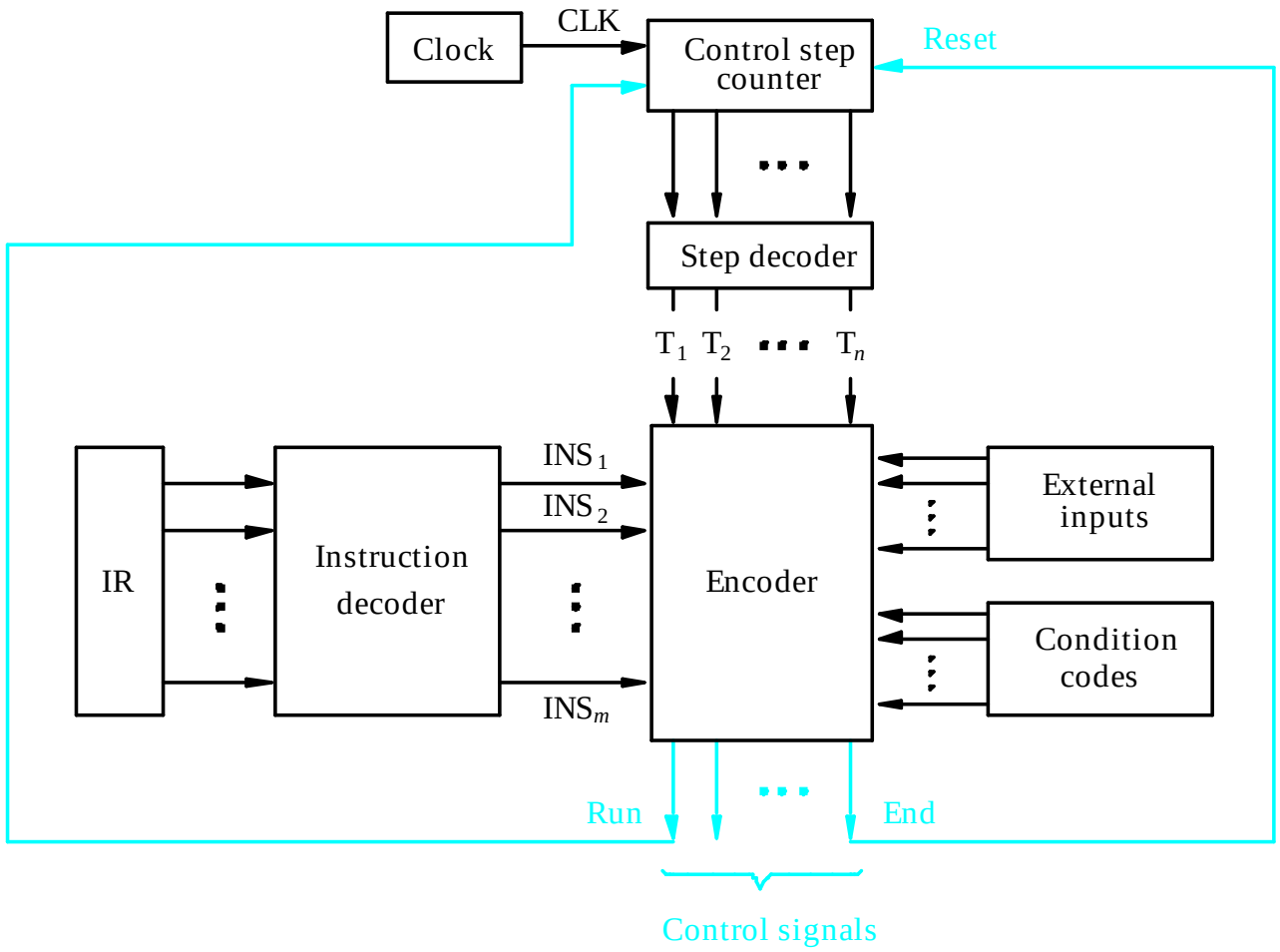


Figure 7.11. Separation of the decoding and encoding functions.

Microprogrammed Control Unit

- 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	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.

Reference-Fig 7.5 in previous slide

- 1 control word(CW) is 1 row of above table
  - Each bit responsible for Activating some micro-oprations
  - 1 CW contains 1 micro-instruction.
- 7 microinstructions in Fig 7.5 called micro-routine
- All micro-routines required to code a computer instruction set are stored on **control memory** ROM

States of a Simple Accumulator based Processor from Hayes Page 328

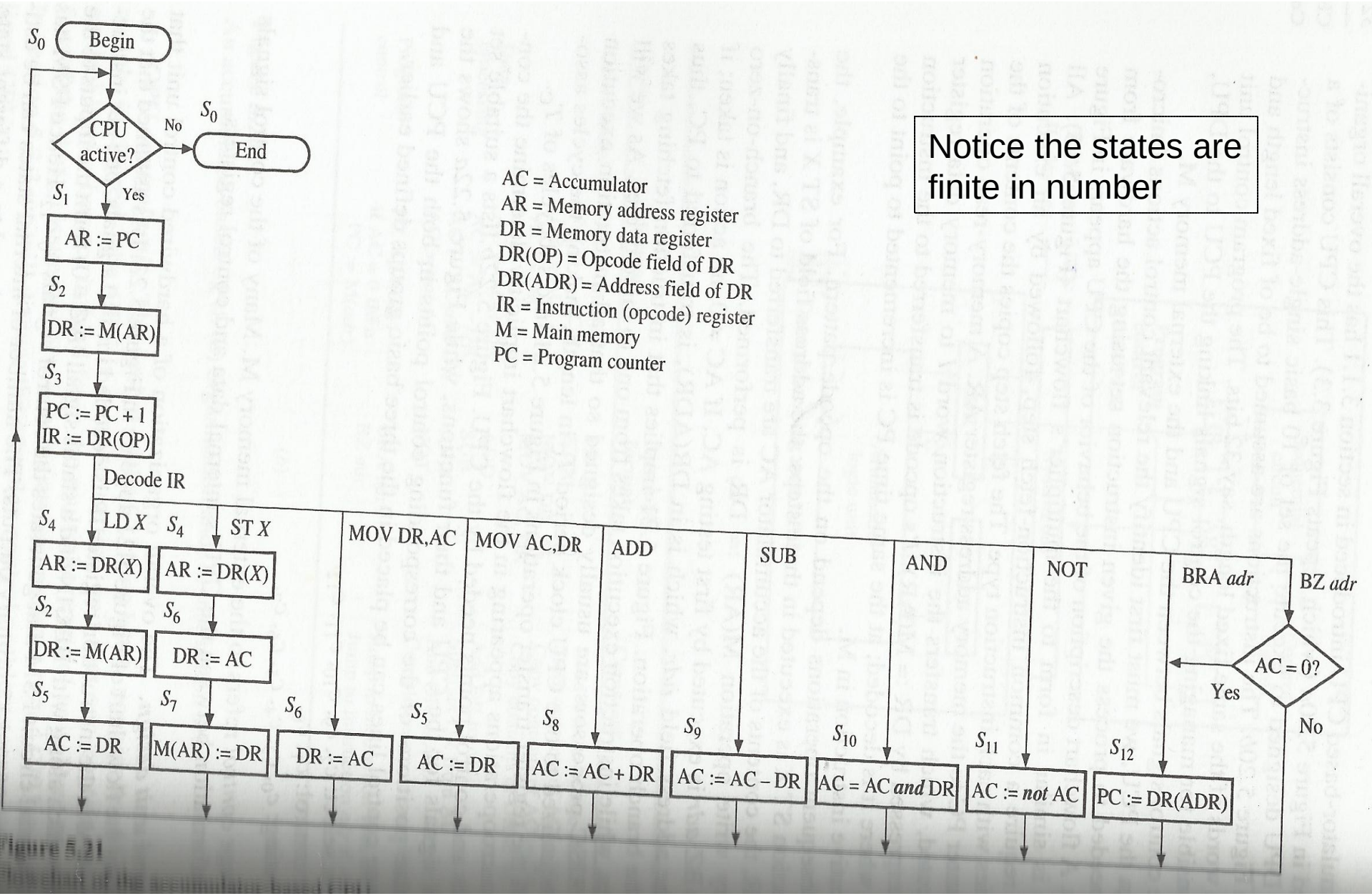


Figure 5.21
Flow chart of the accumulator based CPU

Control Sequencer

- Sends out control words (16 bit in example)
  - One during each T state or clock cycle
- Each is a microinstruction , stored in control memory(ROM)
- Each bit of these 16 bits 1 s and 0s are the set of control signals required for all the microoperations(register input/output enable,arithmetic/login operation) required at that state.

Control Memory(ROM)

Fetch

LDA

ADD

Address	Control Word
0 H	……
1 H	……
2H	……
3H	……
4H	……
5H	…… …..
6H	.
7H	.
8H	.

Address ROM

Op Code	Starting Address
0000 (LDA)	0011
0001 (ADD)	0110
0010 (SUB)	1001
	……
	…… …..
	.
	.
	.

Next Address Decision

- Depending on ALU flags and control buffer register:
  - Get next instruction
    - Add 1 to control address register
  - Jump to new routine based on jump microinstruction
    - Load address field of control buffer register into control address register
  - Jump to machine instruction routine
    - Load control address register based on opcode in IR

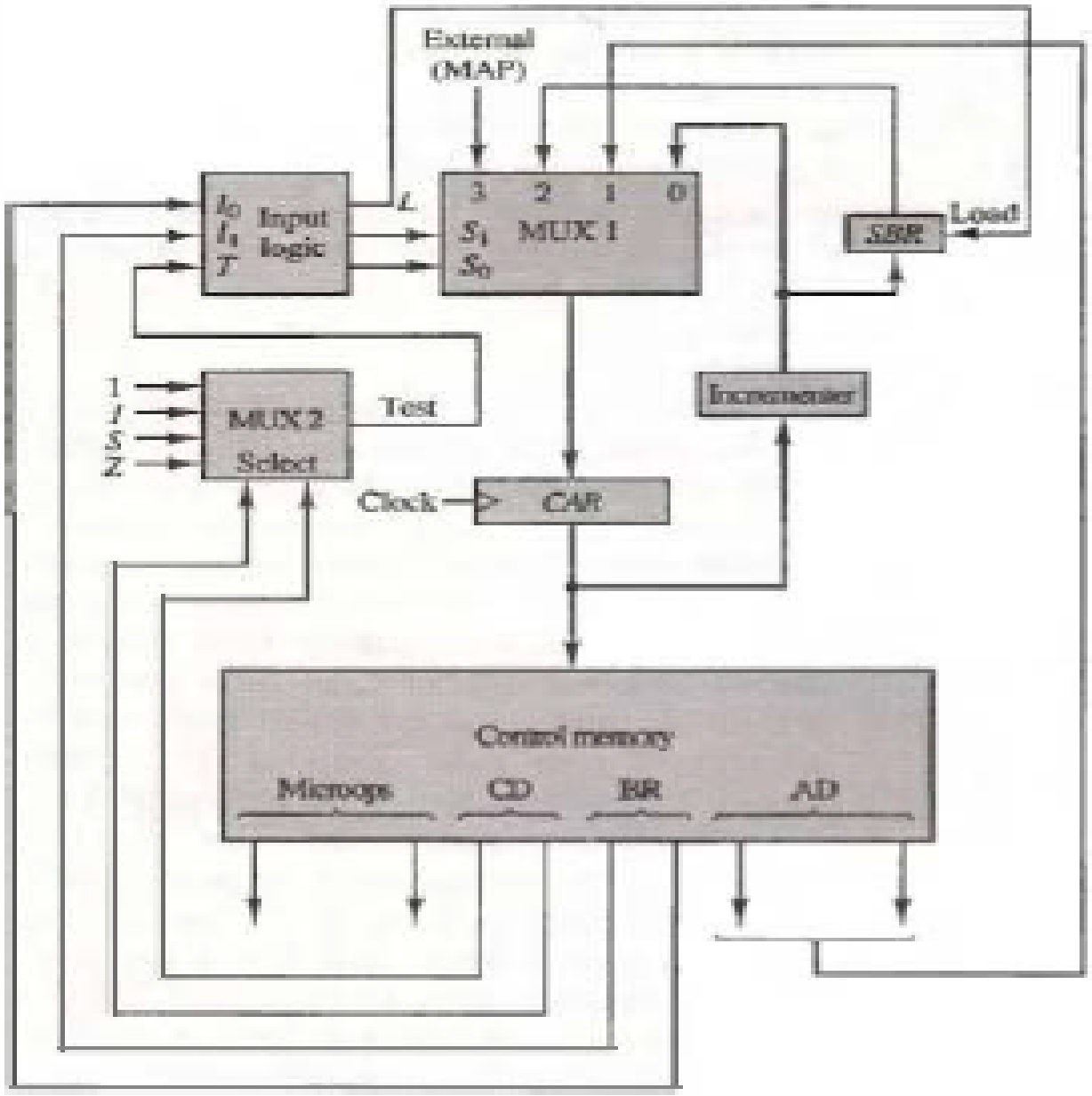


Figure 7-8 Microprogram sequencer for a general memory.

MUX1	
S1 S0	L
0 0	
0 1	
1 0	
1 1	1

Control Memory(ROM)

Fetch

LDA

ADD

Address	Control Word
0 H	……
1 H	……
2H	……
3H	……
4H	……
5H	…… …..
6H	.
7H	.
8H	.

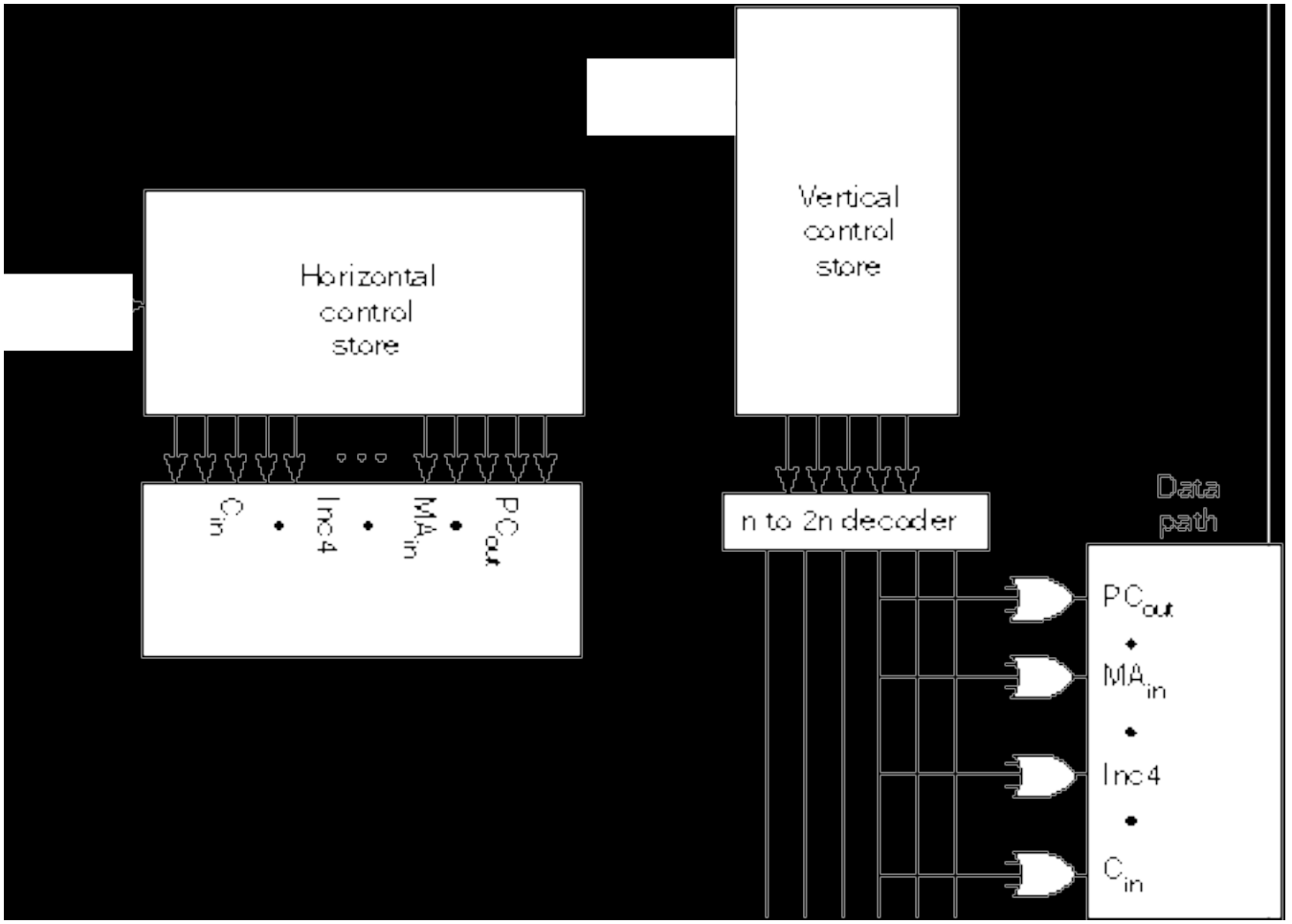
Address ROM

Op Code	Starting Address
0000 (LDA)	0011
0001 (ADD)	0110
0010 (SUB)	1001
	……
	…… …..
	.
	.
	.

Horizontal Versus Vertical Microcode Schemes

- In **horizontal** microcode, each control signal is represented by a bit in the μinstruction
  - Fewer control store words of more bits per word
- In **vertical** microcode, a set of true control signals is represented by a shorter code
  - Vertical μcode only allows RTs in a step for which there is a vertical μinstruction code
  - Thus vertical μcode may take more control store words of fewer bits

Completely Horizontal and Vertical Microcoding



Horizontal Micro-programming

- Wide control memory word
- High degree of parallel operations possible
- Little encoding of control information
- Fast

Vertical Micro-programming

- Width can be much narrower
- Control signals encoded into function codes – need to be decoded
- More complex, more complicated to program, less flexibility
- More difficult to modify
- Slower

Difference between Hardwired Control Unit Processor and Microprogrammed Control Unit Processor

Hardwired Control system

- Advantage
  - can operate at high speed (RISC)
- Disadvanage:
  - Hardware error : difficult to rectify
  - Too complex instruction set – impossible to implement
  - Not easy to modify(very little flexibility)

Microprogrammed Control System

- Advantage:
  - Simplifies design of control unit
    - Cheaper
    - Less error-prone
    - Easier to modify

  (CISC)
- Disadvantage:
  - Slower

Practice Exercises: Control Unit

1. State the difference between Hardwired Control Unit Processor and Microprogrammed Control Unit Processor.
2. Explain using block diagram the functioning of a
   a. hardwired control unit
   b. microprogrammed control unit

Practice Exercises: Control Unit

3. Explain with a block diagram how the next address for the control word to be fetched is found using a MUX.
4. Given a control memory of size 2048 and word size of 12 bits, how many bits each will be the 4 addresses that go into the MUX ?