

Figure 10-7 Flowchart for sign-magnitude addition and subtraction

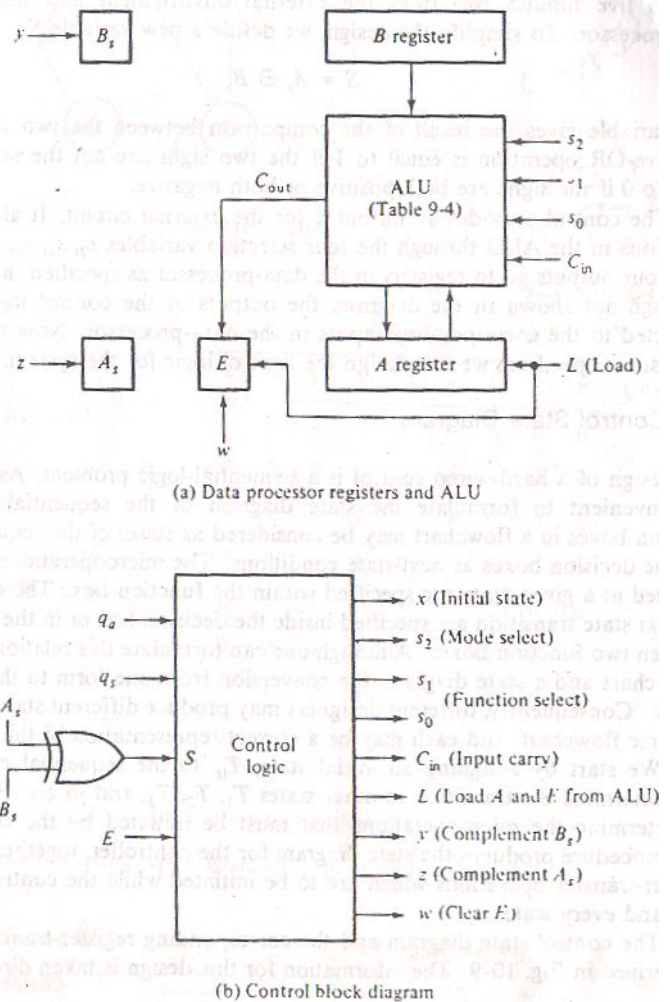
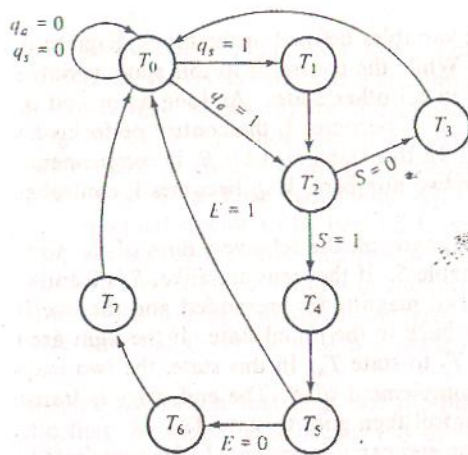


Figure 10-8 System block diagram



(a) State diagram

Control outputs

	x	s_2	s_1	s_0	C_{in}	L	y	z	w
T_0 : Initial state $x = 1$	1	0	0	0	0	0	0	0	0
T_1 : $B_5 \leftarrow \bar{B}_5$	0	0	0	0	0	0	1	0	0
T_2 : nothing	0	0	0	0	0	0	0	0	0
T_3 : $A \leftarrow A + B$, $E \leftarrow C_{out}$	0	0	0	1	0	1	0	0	0
T_4 : $A \leftarrow A + \bar{B} + 1$, $E \leftarrow C_{out}$	0	0	1	0	1	1	0	0	0
T_5 : $E \leftarrow 0$	0	0	0	0	0	0	0	0	1
T_6 : $A \leftarrow \bar{A}$	0	1	1	1	0	1	0	0	0
T_7 : $A \leftarrow A + 1$, $A_5 \leftarrow \bar{A}_5$	0	0	0	0	1	1	0	1	0

(b) Sequence of register transfers

Figure 10-9 Control state diagram and sequence of microoperations

$B_5 \rightarrow \text{sign of } B$

$A_5 \rightarrow \text{sign of } A$

$E \rightarrow \text{overflow}$

$L \rightarrow \text{load } A$

$= 1 \rightarrow \text{not } A$

\Rightarrow

q_0 Add
 q_1 Subtract
 $S = 0$ Signs alike
 $S = 1$ Signs unlike
 E Output carry

$S = A \oplus B$

$S_2 \rightarrow \text{mode select}$

$= 0 \rightarrow \text{Arithmetic}$

$= 1 \rightarrow \text{Logic}$

$y \rightarrow \text{invert } B_5$

$z \rightarrow \text{not } A_5$

$w \rightarrow \text{clear } E$

TABLE 10-3 Binary microprogram for control memory

ROM address	ROM outputs													
	x	s_2	s_1	s_0	C_{in}	L	y	z	w	Address			Select	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0 0 0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
0 0 1	0	0	0	0	0	0	0	1	0	0	1	0	0	1
0 1 0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
0 1 1	0	0	0	1	0	1	0	0	0	0	0	0	0	1
1 0 0	0	0	1	0	1	1	0	0	0	1	0	1	0	1
1 0 1	0	0	0	0	0	0	0	0	1	0	0	0	1	1
1 1 0	0	1	1	1	0	1	0	0	0	1	1	1	0	1
1 1 1	0	0	0	0	1	1	0	1	0	0	0	0	1	1

$B_3 \rightarrow$ sign of B

$A_5 \rightarrow$... A_5

$E \rightarrow$ overflow

$L \rightarrow$ load A

$= 1 \rightarrow$... A

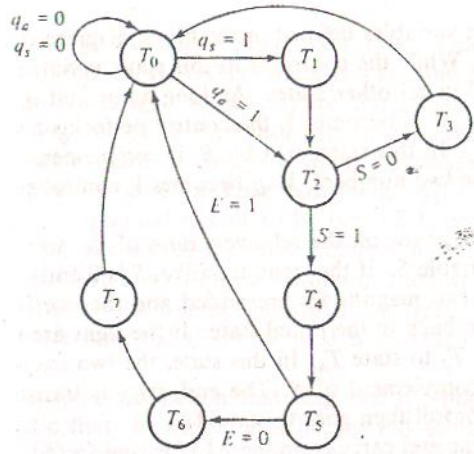
q_0 Add
 q_1 Subtract
 $S = 0$ Signs alike
 $S = 1$ Signs unlike
 E Output carry

$$S = A \oplus B$$

$S_2 \rightarrow$ mode select
 $= 0 \rightarrow$ Arithmetic
 $= 1 \rightarrow$ Logic

$y \rightarrow$ invert B_3
 $z \rightarrow$... A_5

$w \rightarrow$ clear E



(a) State diagram

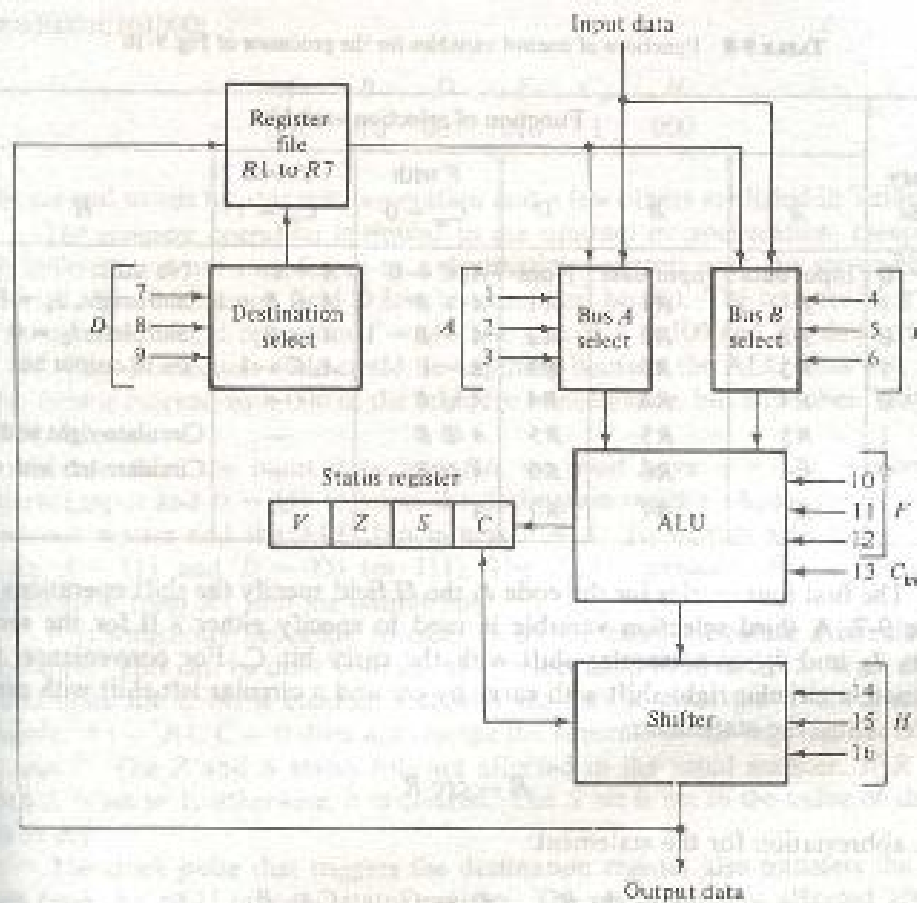
	Control outputs									
	x	s_2	s_1	s_0	C_{in}	L	y	z	w	
T_0 : Initial state $x = 1$	1	0	0	0	0	0	0	0	0	
T_1 : $B_3 \leftarrow \bar{B}_3$	0	0	0	0	0	0	1	0	0	
T_2 : nothing	0	0	0	0	0	0	0	0	0	
T_3 : $A \leftarrow A + B$, $E \leftarrow C_{out}$	0	0	0	1	0	1	0	0	0	
T_4 : $A \leftarrow A + \bar{B} + 1$, $E \leftarrow C_{out}$	0	0	1	0	1	1	0	0	0	
T_5 : $E \leftarrow 0$	0	0	0	0	0	0	0	0	1	
T_6 : $A \leftarrow \bar{A}$	0	1	1	1	0	1	0	0	0	
T_7 : $A \leftarrow A + 1$, $A_5 \leftarrow \bar{A}_5$	0	0	0	0	1	1	0	1	0	

(b) Sequence of register transfers

Figure 10-9 Control state diagram and sequence of microoperations

TABLE 9-8 Functions of control variables for the processor of Fig. 9-16

Binary code	Function of selection variables					
	<i>A</i>	<i>B</i>	<i>D</i>	<i>F</i> with $C_{in} = 0$	<i>F</i> with $C_{in} = 1$	<i>H</i>
0 0 0	Input data	Input data	None	$A, C \leftarrow 0$	$A + 1$	No shift
0 0 1	<i>R1</i>	<i>R1</i>	<i>R1</i>	$A + B$	$A + B + 1$	Shift-right, $I_R = 0$
0 1 0	<i>R2</i>	<i>R2</i>	<i>R2</i>	$A - B - 1$	$A - B$	Shift-left, $I_L = 0$
0 1 1	<i>R3</i>	<i>R3</i>	<i>R3</i>	$A - 1$	$A, C \leftarrow 1$	0's to output bus
1 0 0	<i>R4</i>	<i>R4</i>	<i>R4</i>	$A \vee B$	—	—
1 0 1	<i>R5</i>	<i>R5</i>	<i>R5</i>	$A \oplus B$	—	Circulate-right with <i>C</i>
1 1 0	<i>R6</i>	<i>R6</i>	<i>R6</i>	$A \wedge B$	—	Circulate-left with <i>C</i>
1 1 1	<i>R7</i>	<i>R7</i>	<i>R7</i>	\bar{A}	—	—



(a) Block diagram



(b) Control word

Figure 9-16 Processor unit with control variables

Start (address 8)

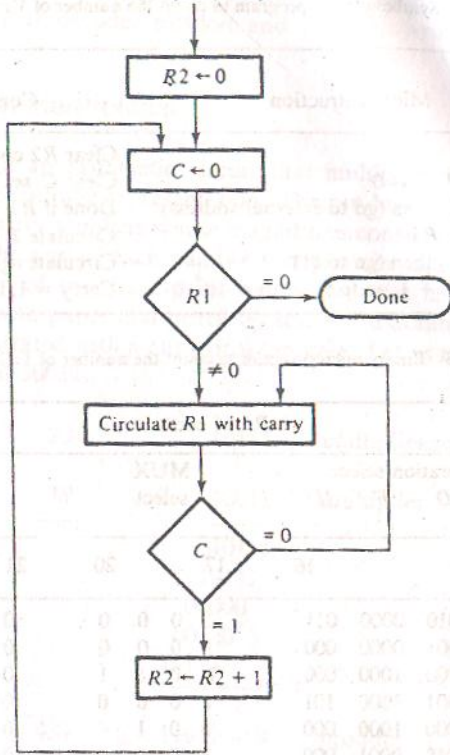


Figure 10-12 Flowchart for counting the number of 1's in register R1

TABLE 10-4 Symbolic microprogram to count the number of 1's in R1

ROM address	Microinstruction	Comments
8	$R2 \leftarrow 0$	Clear R2 counter
9	$R1 \leftarrow R1, C \leftarrow 0$	Clear C, set status bits
10	If (Z = 1) then (go to external address)	Done if R1 = 0
11	$R1 \leftarrow \text{crc } R1$	Circulate R1 right with carry
12	If (C = 0) then (go to 11)	Circulate again if C = 0
13	$R2 \leftarrow R2 + 1$, go to 9	Carry = 1, increment R2

TABLE 10-5 Binary microprogram to count the number of 1's in R1

ROM address	ROM content									
	Microoperation select					MUX select		Address field		
	A	B	D	F	H	17	20	21	26	
001000	000	000	010	0000	011	1	0	0	0	0 0 1 0 0 1
001001	001	000	001	0000	000	1	0	0	0	0 0 1 0 1 0
001010	001	001	000	1000	000	0	0	1	1	0 0 0 0 0 0
001011	001	001	001	1000	101	1	0	0	0	0 0 1 1 0 0
001100	001	001	000	1000	000	1	0	1	0	0 0 1 0 1 1
001101	010	000	010	0001	000	1	0	0	0	0 0 1 0 0 1