PROCESSOR ORGANIZATION

Processor organization

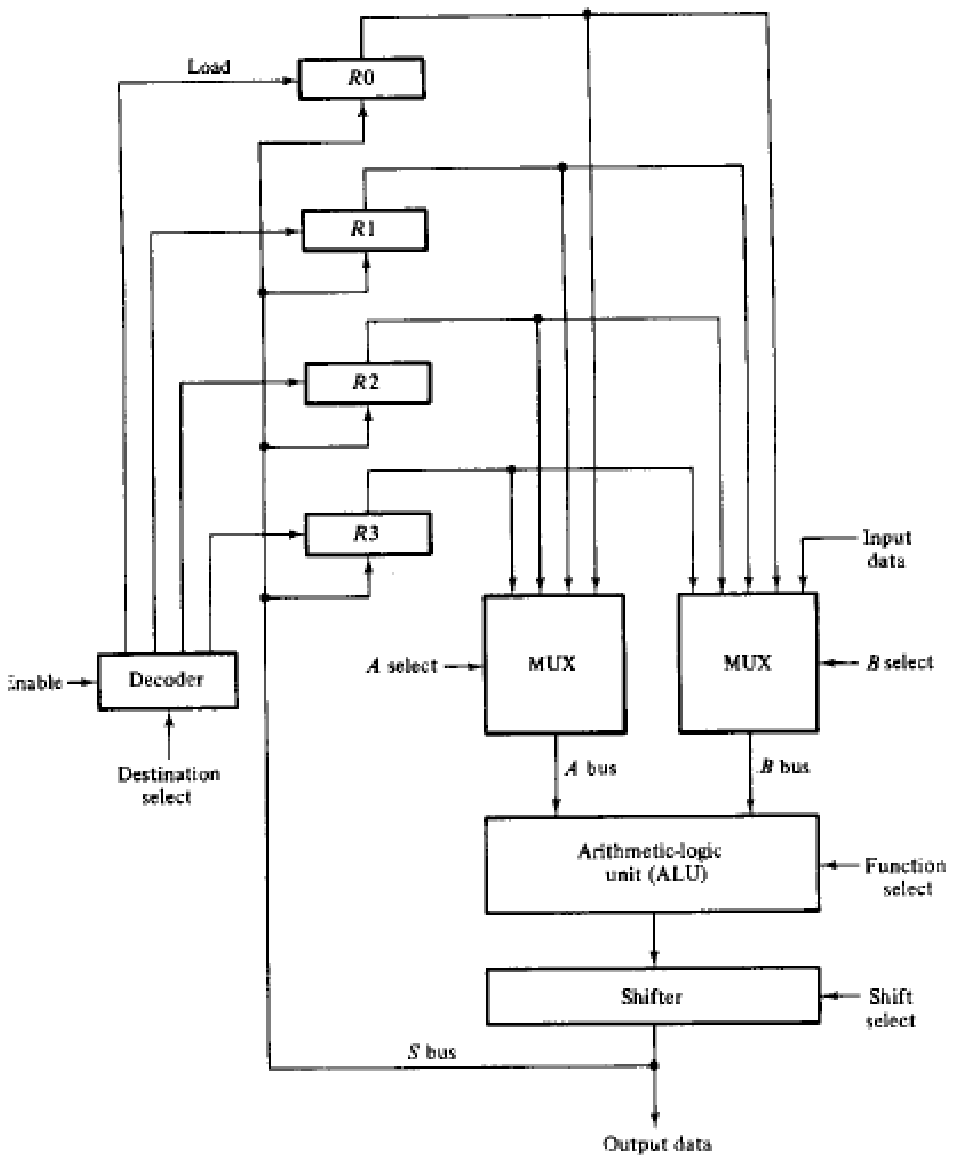


Figure 9-1 Processor registers and ALU connected through common buses

Processor- scratchpad memory

- If registers in processor enclosed in memory unit in processor-scratch pad memory
- Registers neednot be connected through bus, cheaper
- The operation

$$R1 \leftarrow R2 + R3$$

Performed as:

T_1:	$A \leftarrow M[010]$	read $R2$ into register A
T_2:	$B \leftarrow M[011]$	read $R3$ into register B
T_3:	$M[001] \leftarrow A + B$	perform operation in ALU and transfer result to $R1$

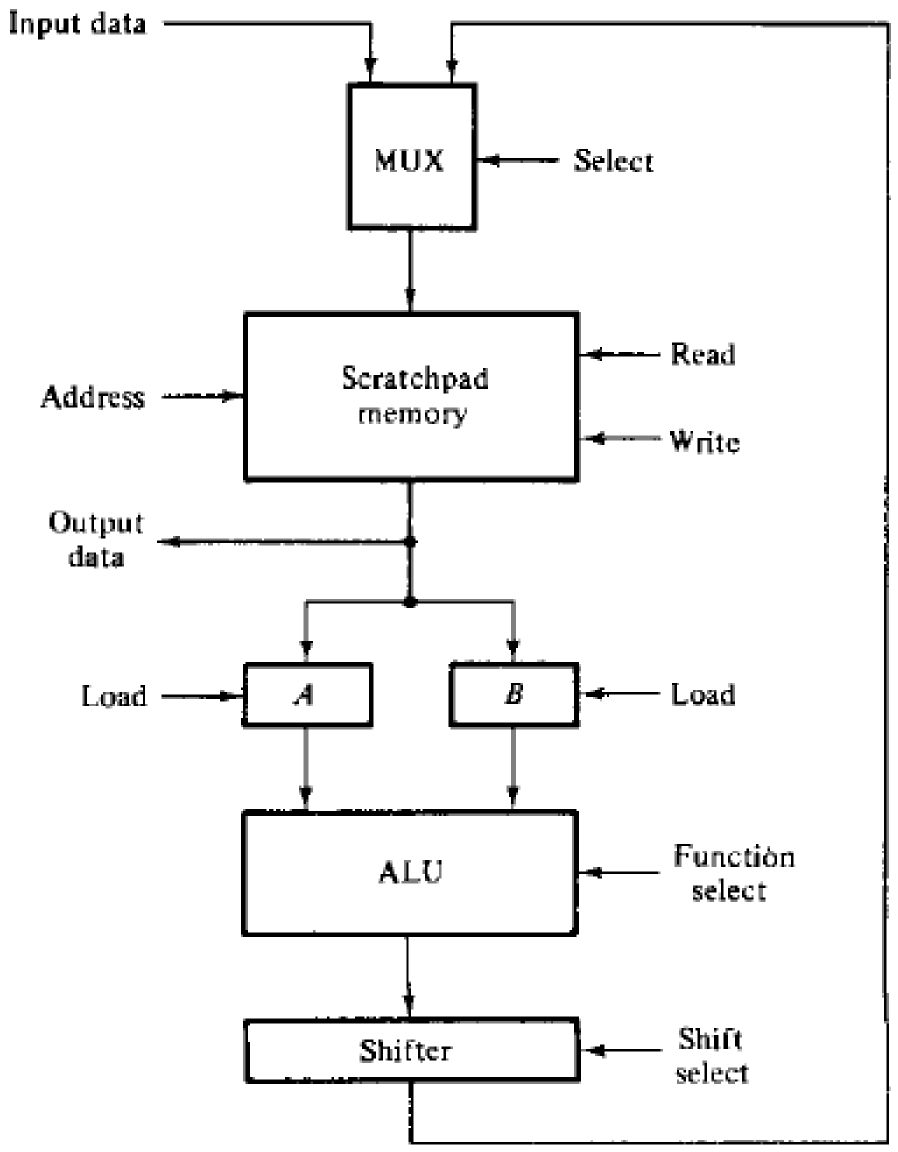


Figure 9-2 Processor unit employing a scratchpad memory

Processor with 2 port memory

- To overcome the delay caused when reading two source registers
- Has 2 address lines to select 2 words

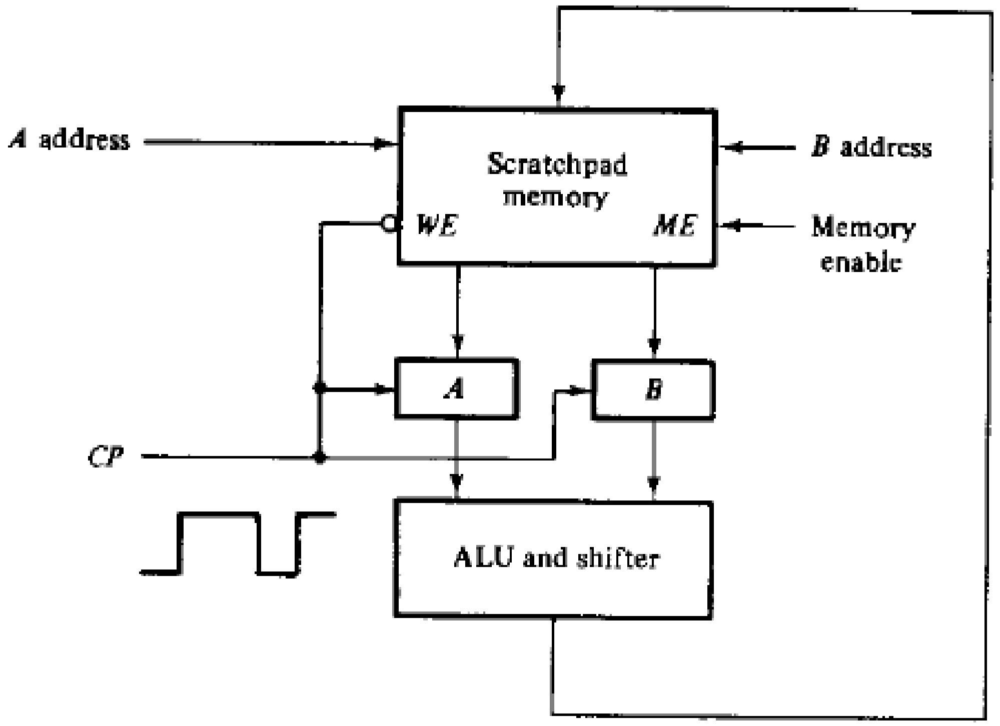


Figure 9-3 Processor unit with a 2-port memory

Design of arithmetic and logic circuits

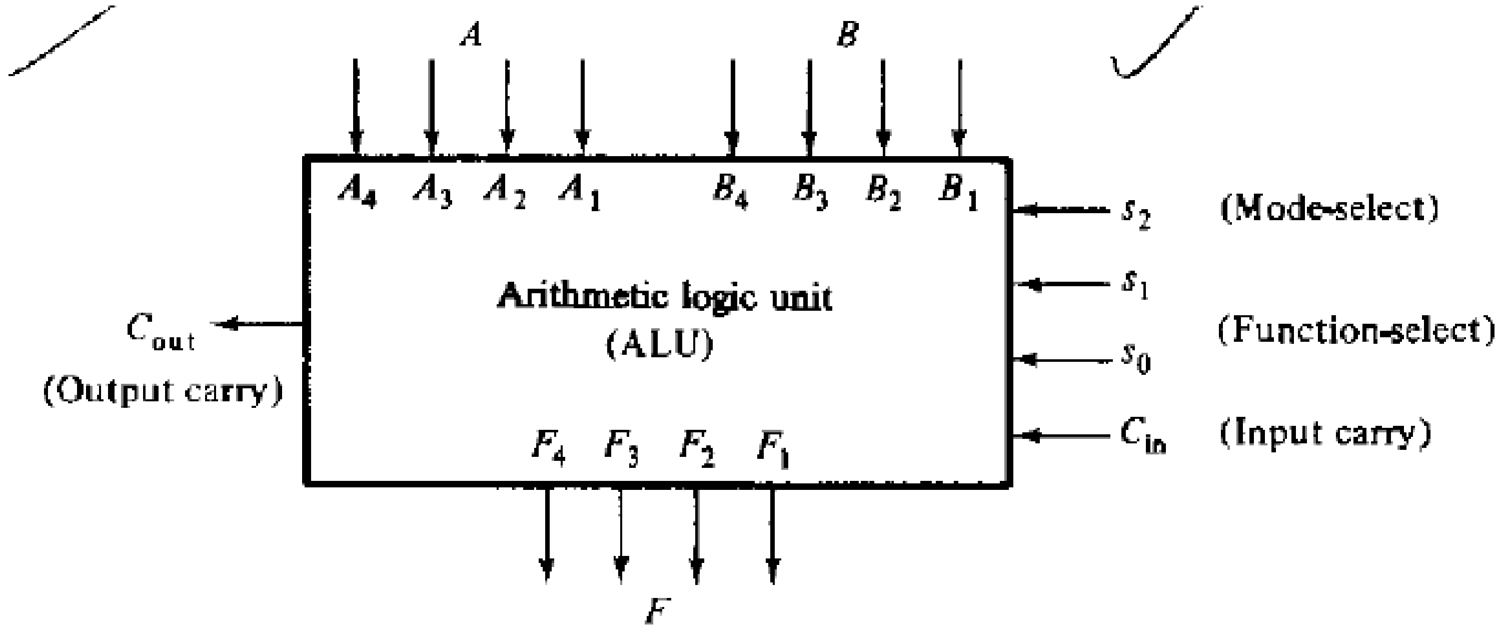
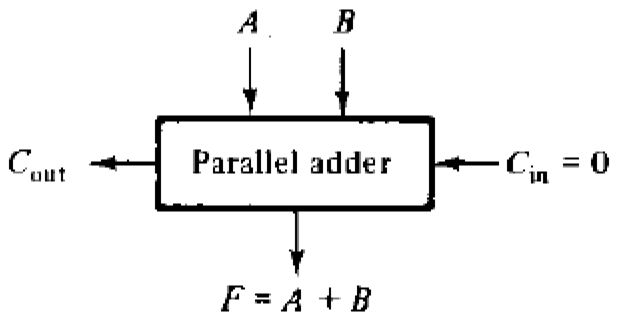


Figure 9-5 Block diagram of a 4-bit ALU

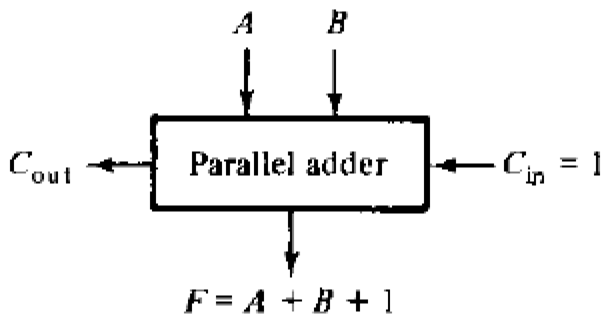
- Implemented in 3 steps:
  1. Design of arithmetic section
  2. Design of logic section
  3. Modification of arithmetic section to include logic too

Design of arithmetic circuit

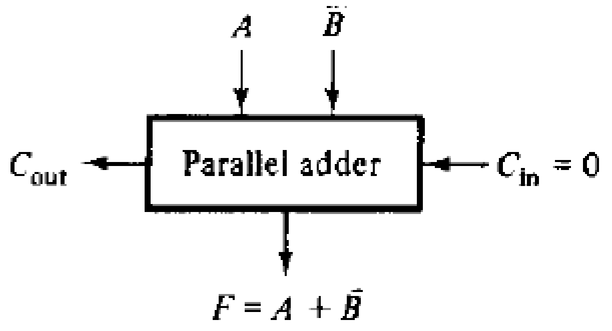
- Basic component – parallel adder



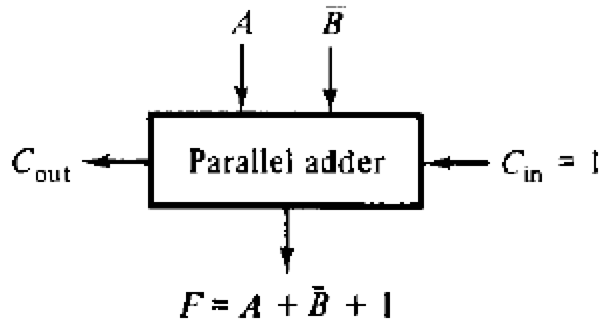
(a) Addition



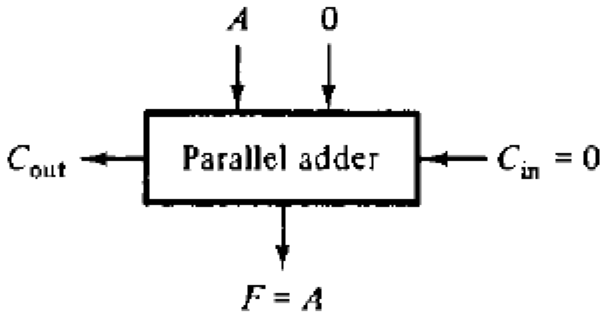
(b) Addition with carry



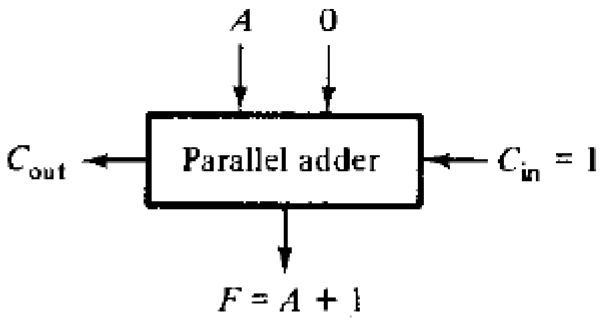
(c) A plus 1's complement of B



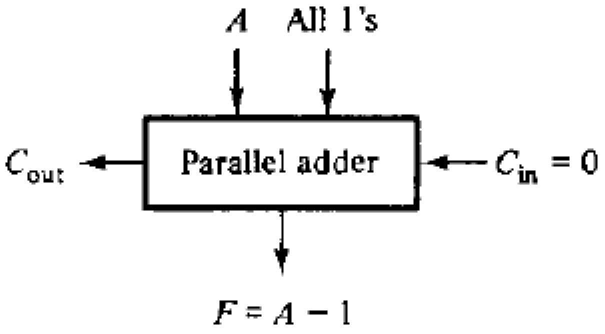
(d) Subtraction



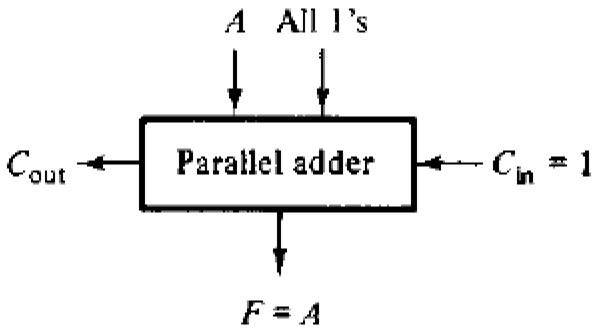
(e) Transfer A



(f) Increment A

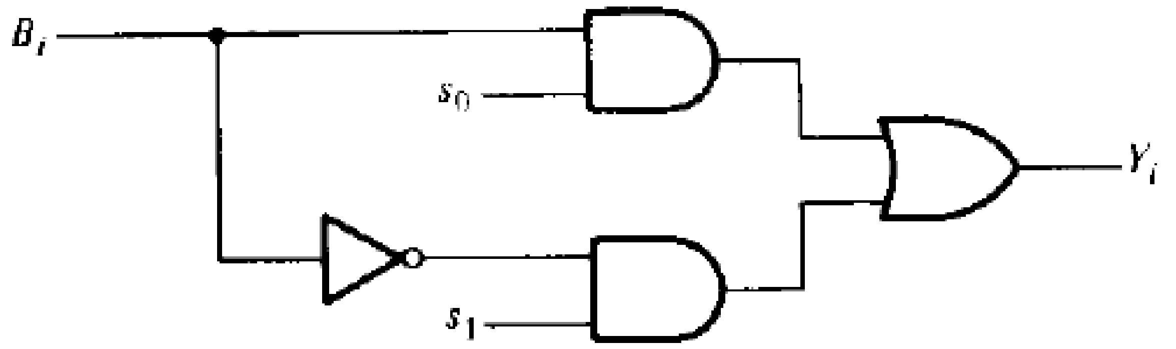


(g) Decrement A



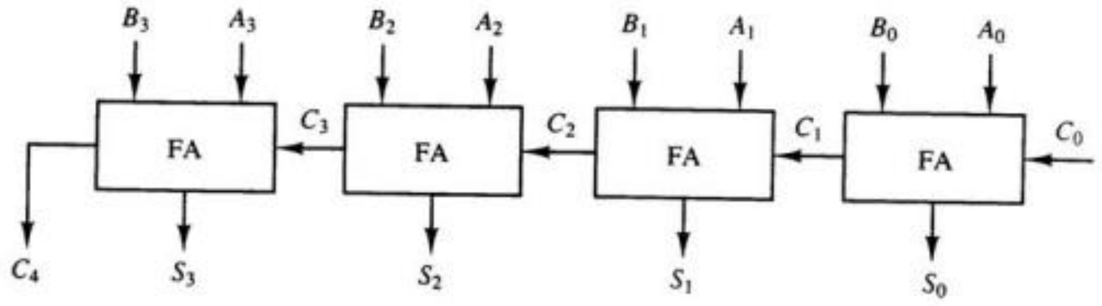
(h) Transfer A

$$X_i = A_i$$
$$Y_i = B_i s_0 + B_i' s_1 \qquad i = 1, 2, \ldots, n$$



s_1	s_0	Y_i
0	0	0
0	1	B_i
1	0	B_i'
1	1	1

Figure 9-7 True/complement, one/zero circuit



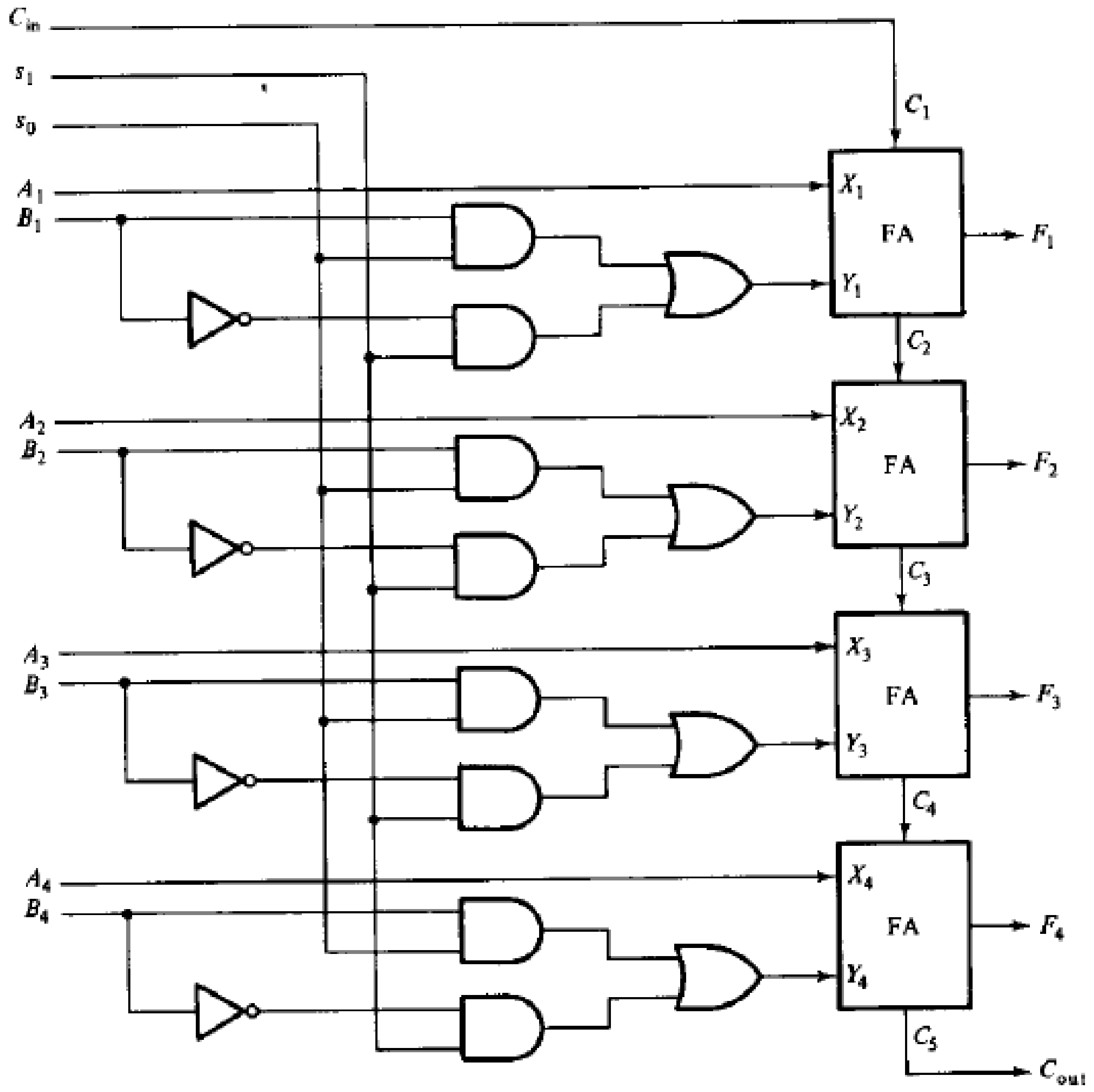
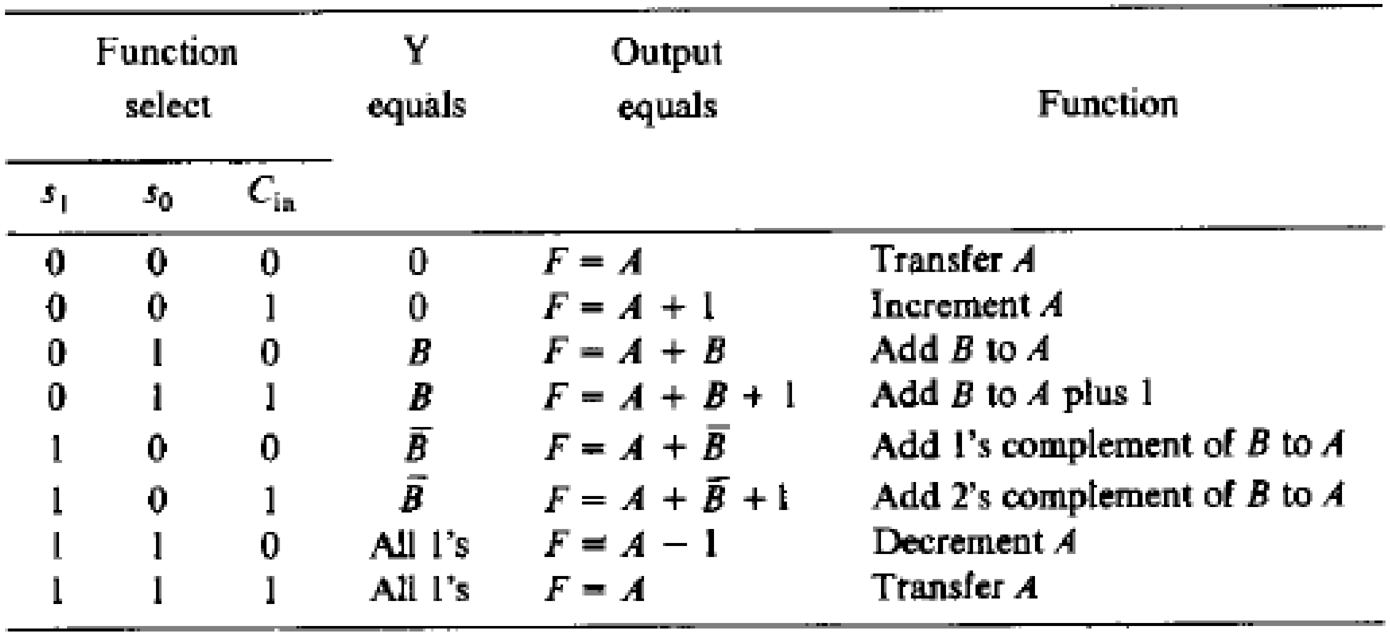


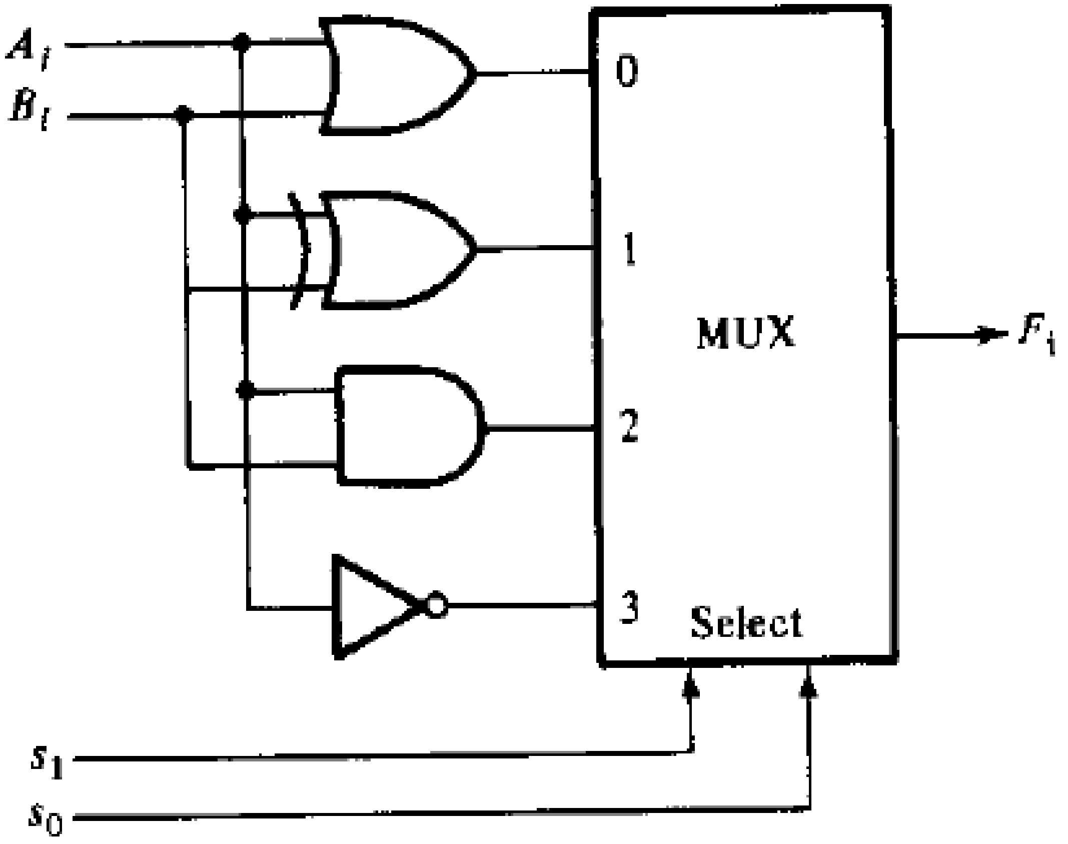
Figure 9-8 Logic diagram of arithmetic circuit



Function select			Y equals	Output equals	Function
s_1	s_0	C_{in}			
0	0	0	0	$F = A$	Transfer A
0	0	1	0	$F = A + 1$	Increment A
0	1	0	B	$F = A + B$	Add B to A
0	1	1	B	$F = A + B + 1$	Add B to A plus 1
1	0	0	$\overline{B}$	$F = A + \overline{B}$	Add 1's complement of B to A
1	0	1	$\overline{B}$	$F = A + \overline{B} + 1$	Add 2's complement of B to A
1	1	0	All 1's	$F = A - 1$	Decrement A
1	1	1	All 1's	$F = A$	Transfer A

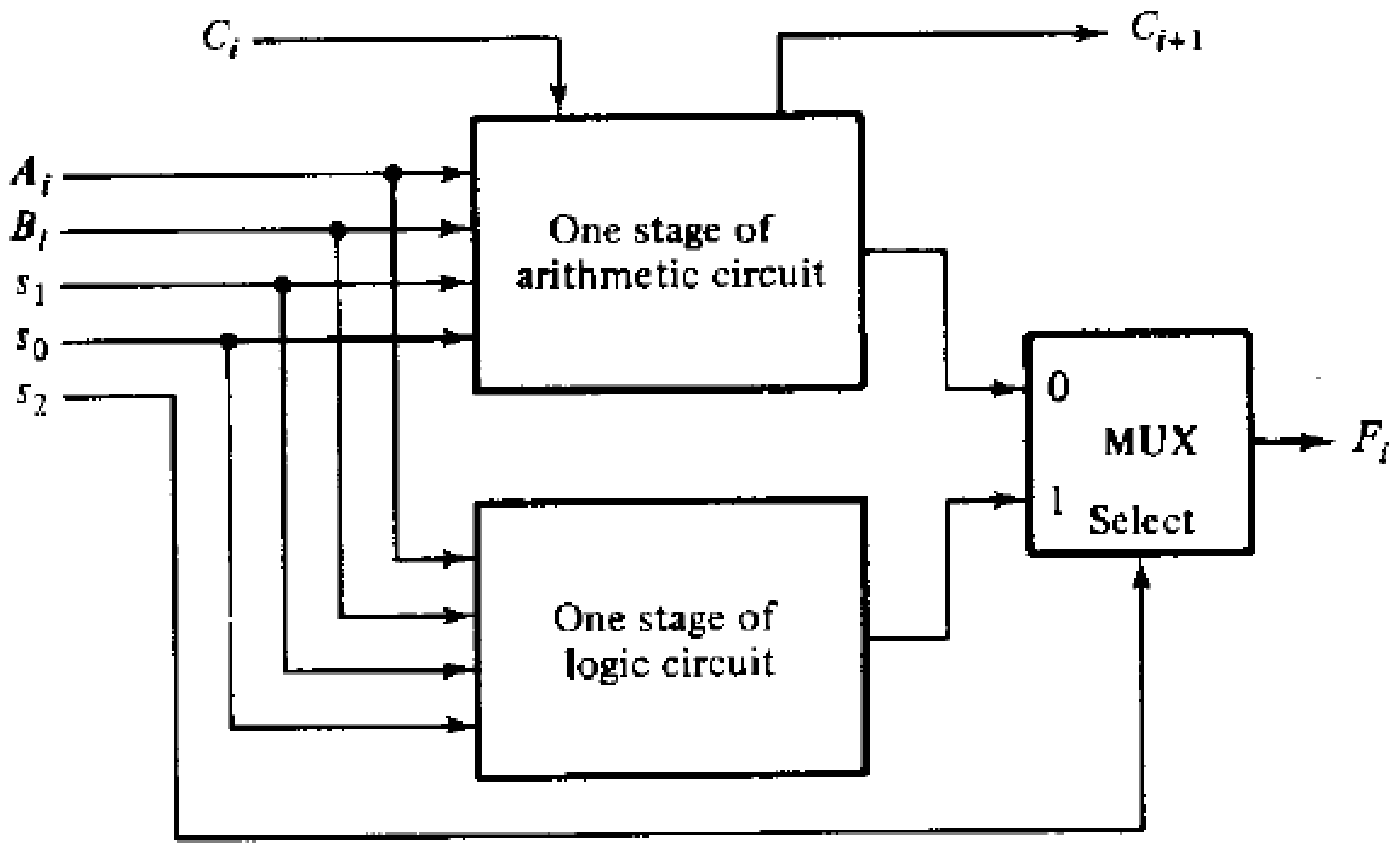
- What will be the effect on output carry in the circuit?

Design of logic circuit



s_1	s_0	Output	Operation
0	0	$F_i = A_i + B_i$	OR
0	1	$F_i = A_i \oplus B_i$	XOR
1	0	$F_i = A_i B_i$	AND
1	1	$F_i = A_i'$	NOT

Combining arithmetic and logic units

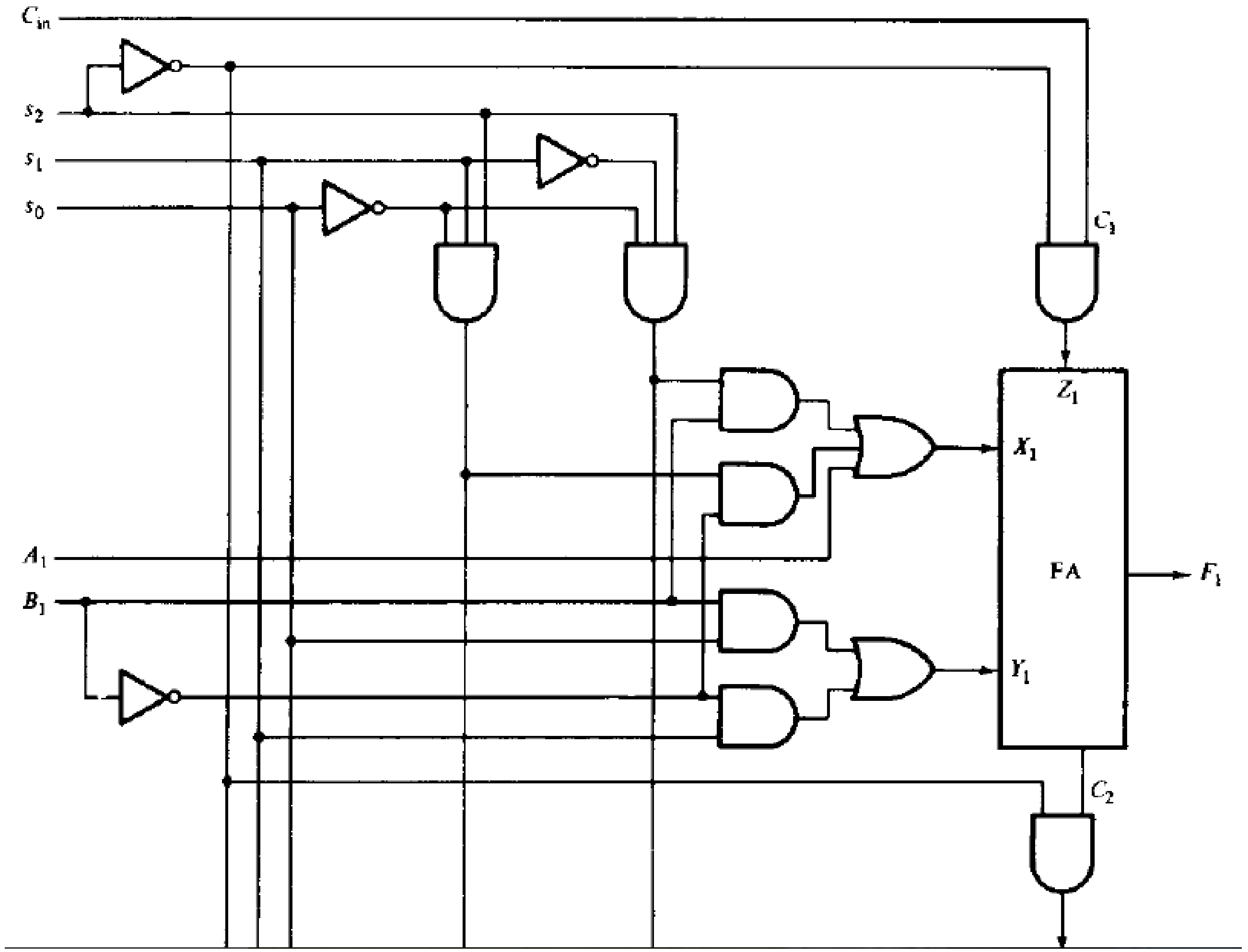


s_2	s_1	s_0	X_i	Y_i	C_i	$F_i = X_i \oplus Y_i$	Operation	Required operation
1	0	0	A_i	0	0	$F_i = A_i$	Transfer *A*	OR
1	0	1	A_i	B_i	0	$F_i = A_i \oplus B_i$	XOR	XOR
1	1	0	A_i	B_i'	0	$F_i = A_i \odot B_i$	Equivalence	AND
1	1	1	A_i	1	0	$F_i = A_i'$	NOT	NOT

$$X_i = A_i + s_2 s_1' s_0' B_i + s_2 s_1 s_0' B_i'$$

$$Y_i = s_0 B_i + s_1 B_i'$$

$$Z_i = s_2' C_i$$


C_{in}
s_2
s_1
s_0
C_1
Z_1
X_1
FA
F_1
Y_1
A_1
B_1
C_2

Logic diagram of ALU

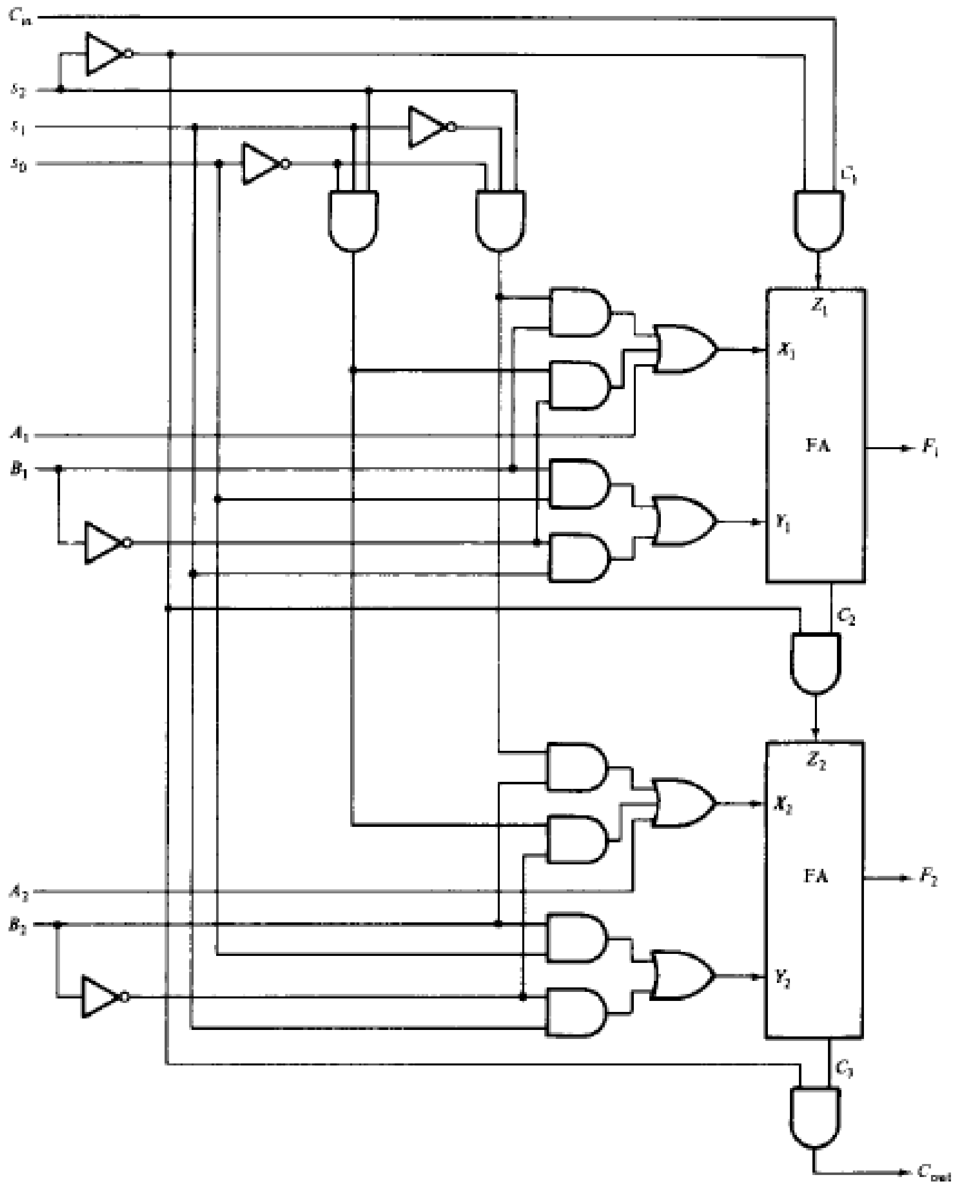


Table 9-4 Function table for the ALU of Fig. 9-13

Selection				Output	Function
s_2	s_1	s_0	C_{in}		
0	0	0	0	$F = A$	Transfer A
0	0	0	1	$F = A + 1$	Increment A
0	0	1	0	$F = A + B$	Addition
0	0	1	1	$F = A + B + 1$	Add with carry
0	1	0	0	$F = A - B - 1$	Subtract with borrow
0	1	0	1	$F = A - B$	Subtraction
0	1	1	0	$F = A - 1$	Decrement A
0	1	1	1	$F = A$	Transfer A
1	0	0	X	$F = A \vee B$	OR
1	0	1	X	$F = A \oplus B$	XOR
1	1	0	X	$F = A \wedge B$	AND
1	1	1	X	$F = \bar{A}$	Complement A

STATUS REGISTER

- Condition code/flag- C,S,Z,V
- C(carry flag): set if output carry is 1
- S(Sign flag): set if highest order bit is 1
- Z(Zero flag): set if output has all 1’s
  - If set after XOR, A,B same
- V(Overflow): set if there is overflow

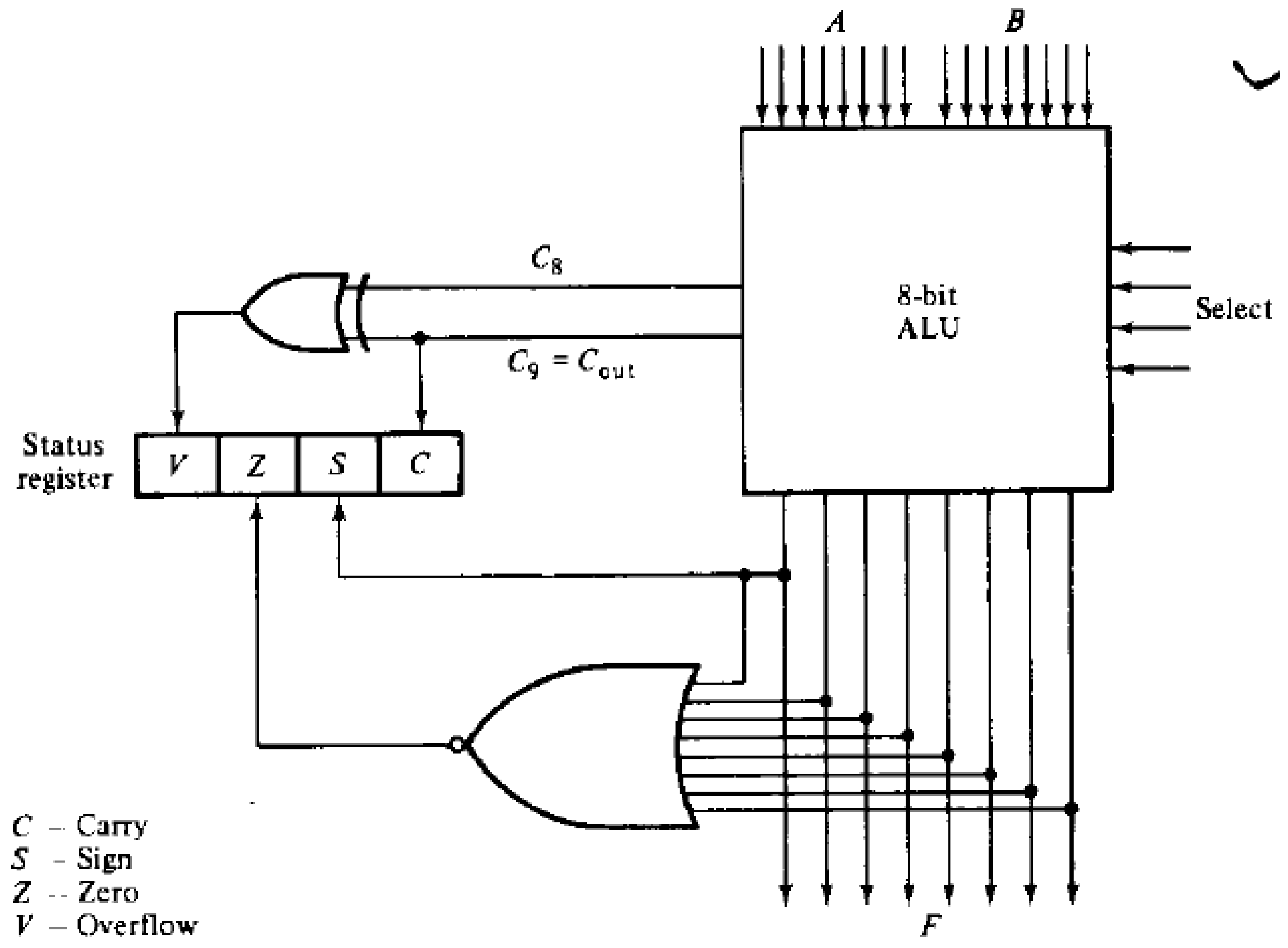


Figure 9-14 Setting bits in a status register

Shifter

H_1	H_0	Operation	Function
0	0	$S \leftarrow F$	Transfer F to S (no shift)
0	1	$S \leftarrow \text{shr } F$	Shift-right F into S
1	0	$S \leftarrow \text{shl } F$	Shift-left F into S
1	1	$S \leftarrow 0$	Transfer 0's into S

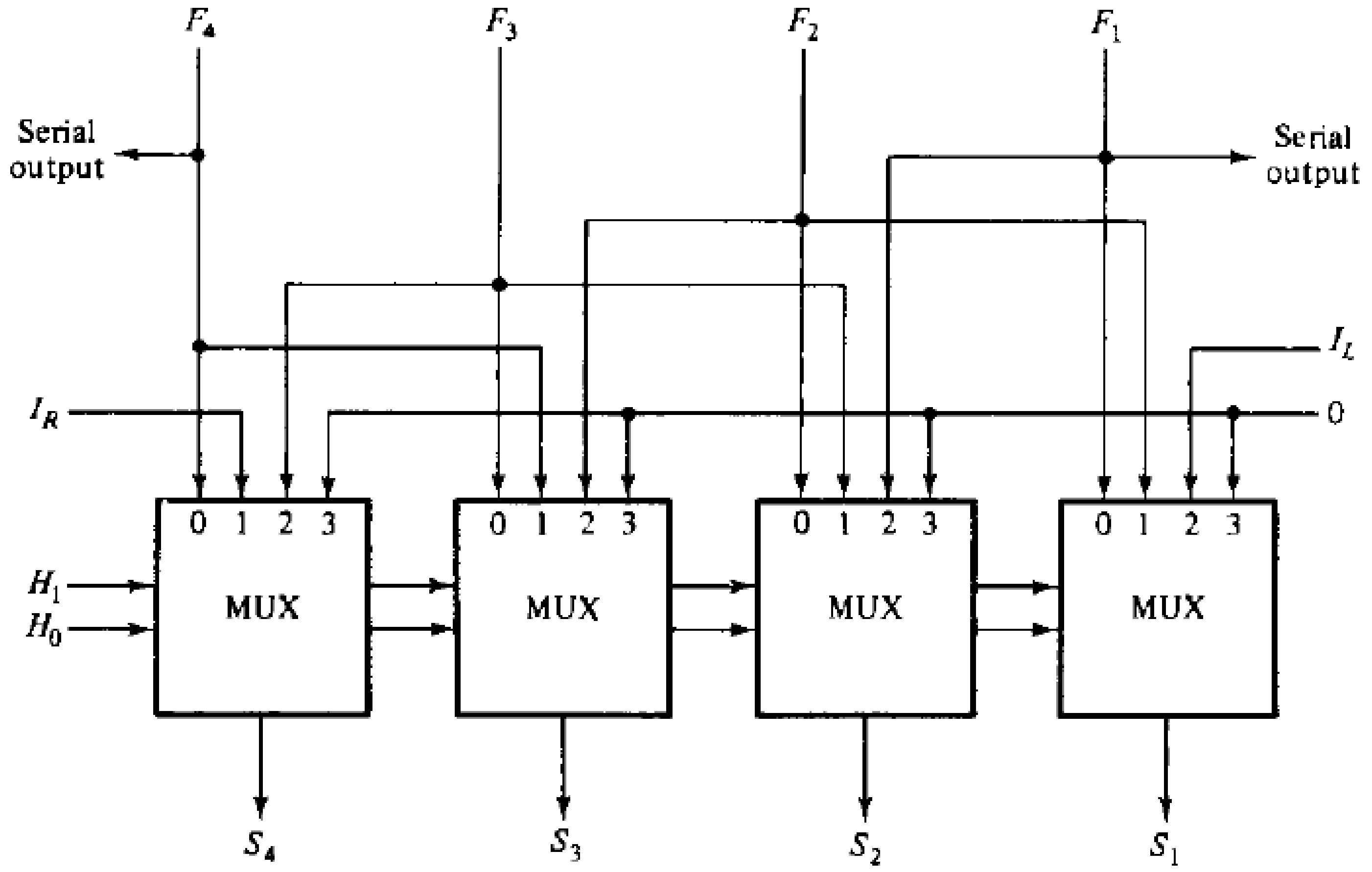
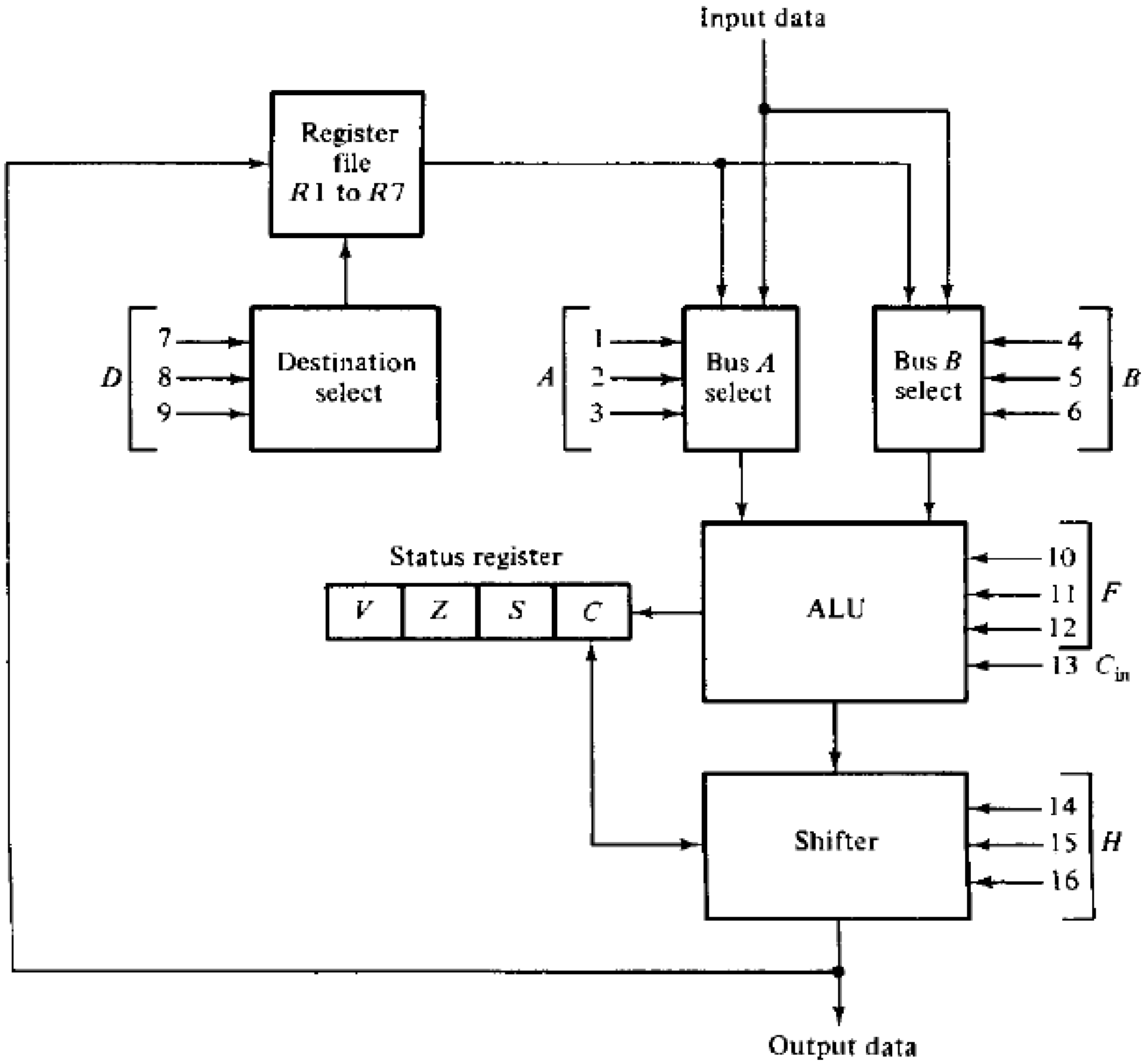
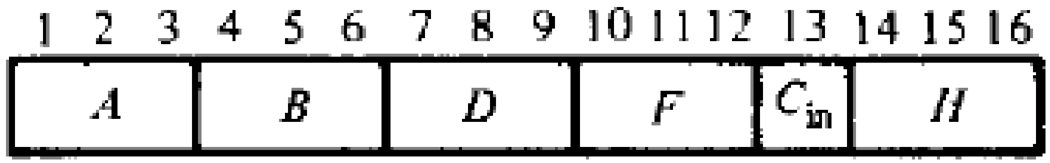


Figure 9-15 4-bit combinational-logic shifter

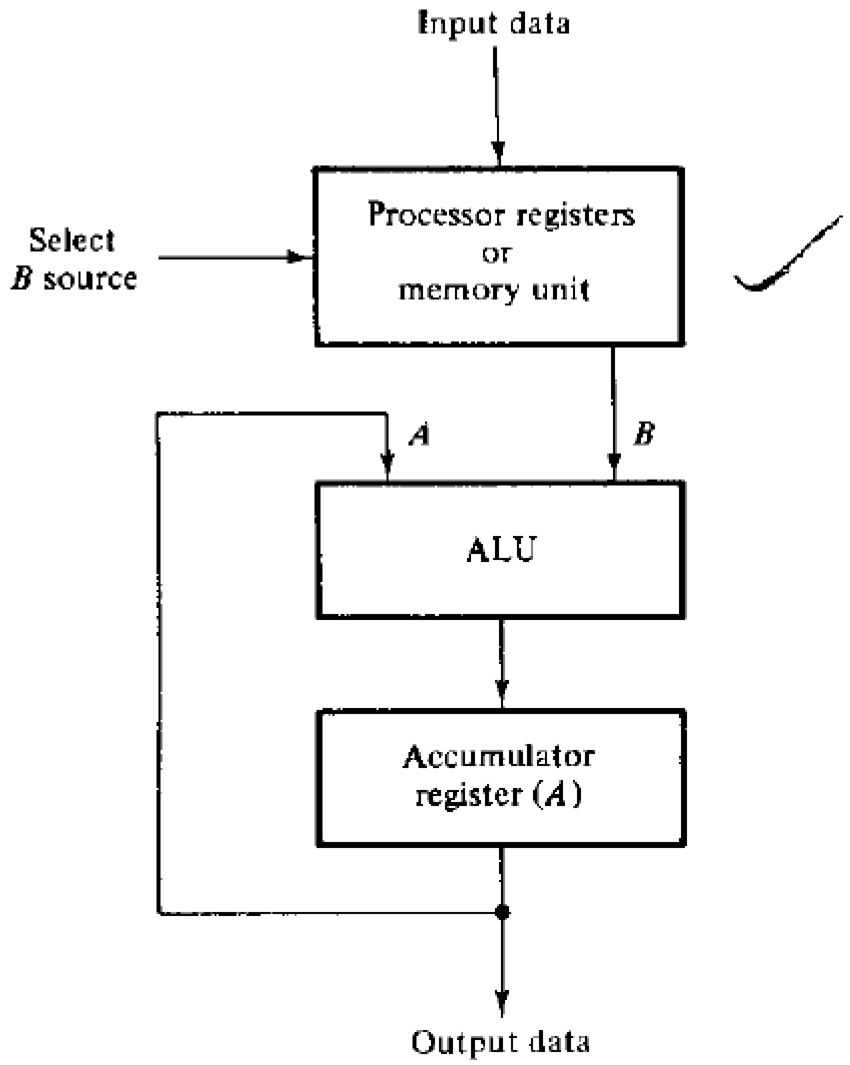
Processor unit





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

Accumulator register



T_1:	$A \leftarrow 0$	clear A
T_2:	$A \leftarrow A + R1$	transfer $R1$ to A
T_3:	$A \leftarrow A + R2$	add $R2$ to A

Accumulator design

- N bit accumulator needs n stages connected in cascade
- Each stage have a JK flipflop with associated circuitries

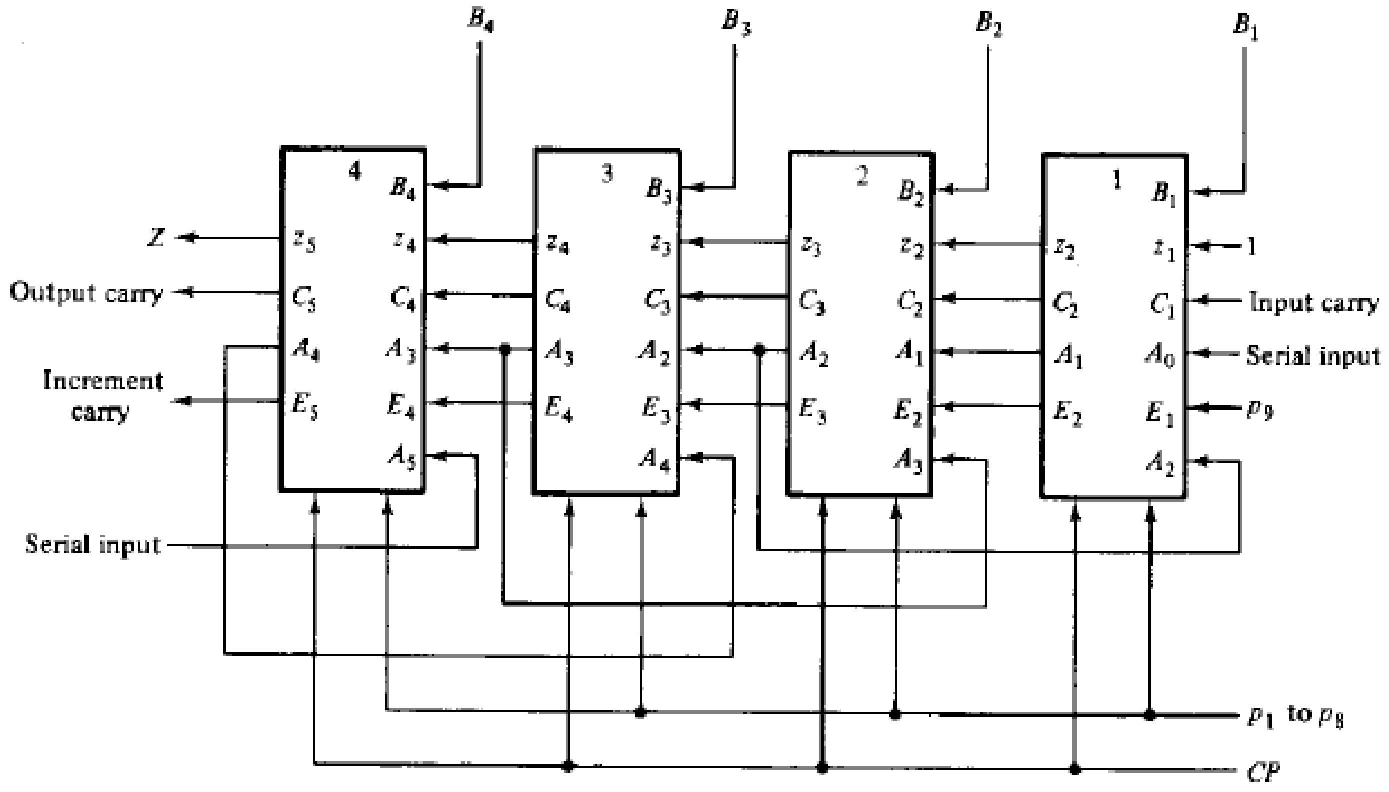


Figure 9-23 4-bit accumulator constructed with four stages

Accumulator design

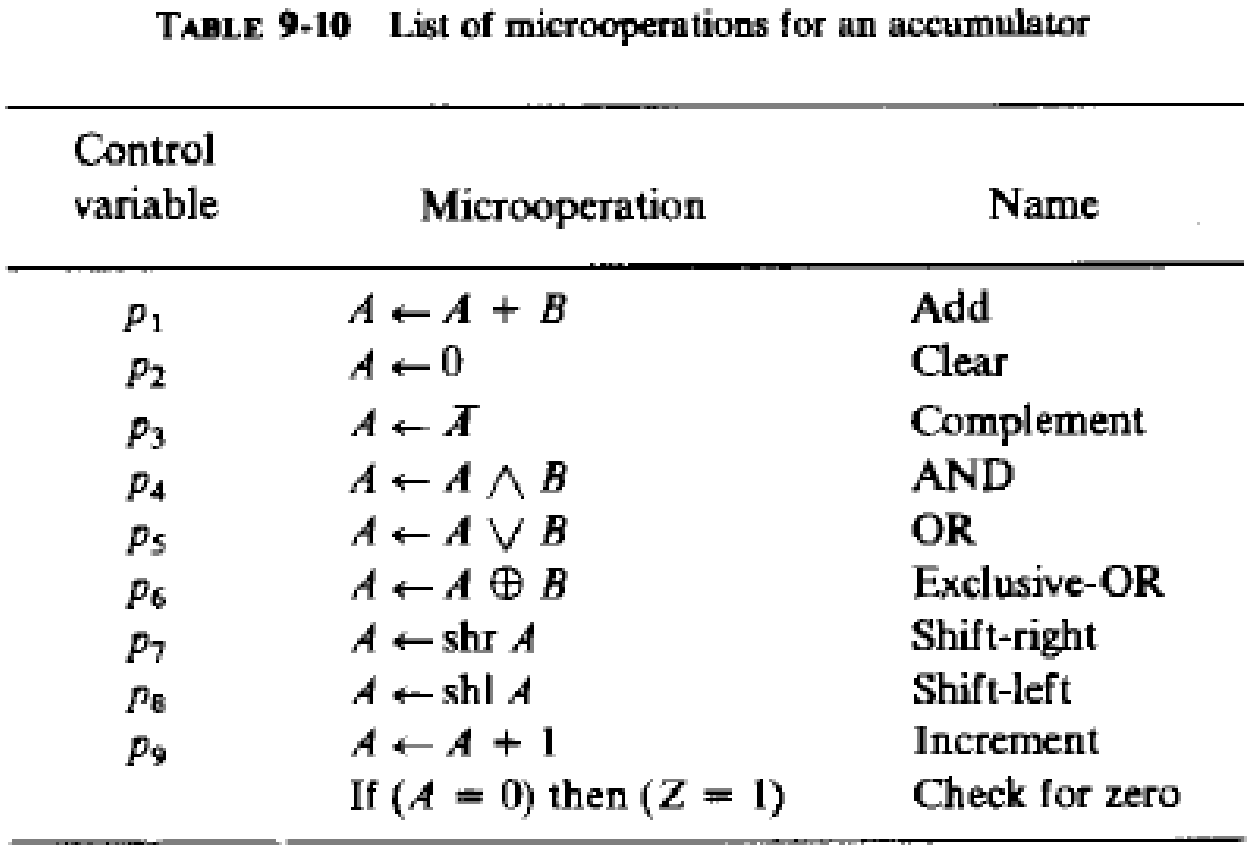


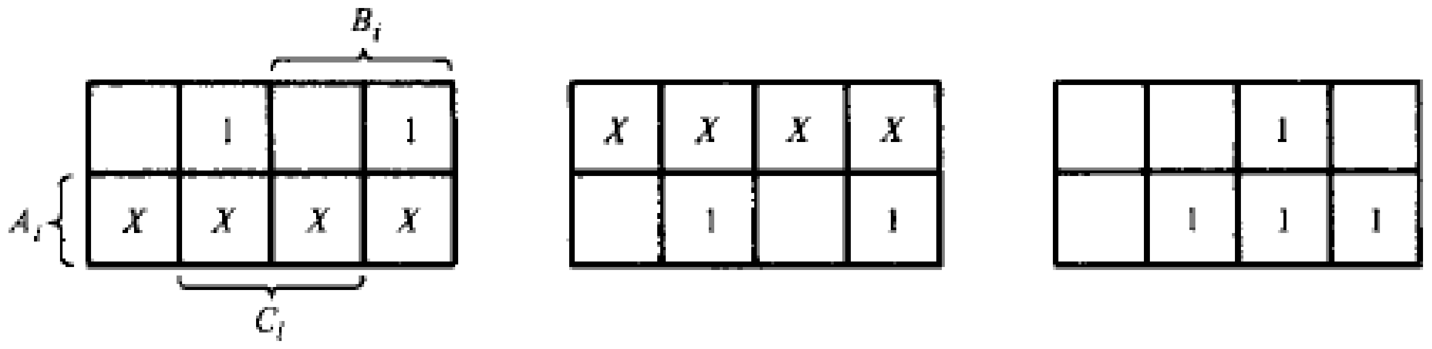
TABLE 9-10 List of microoperations for an accumulator

Control variable	Microoperation	Name
p_1	$A \leftarrow A + B$	Add
p_2	$A \leftarrow 0$	Clear
p_3	$A \leftarrow \bar{A}$	Complement
p_4	$A \leftarrow A \wedge B$	AND
p_5	$A \leftarrow A \vee B$	OR
p_6	$A \leftarrow A \oplus B$	Exclusive-OR
p_7	$A \leftarrow \text{shr } A$	Shift-right
p_8	$A \leftarrow \text{shl } A$	Shift-left
p_9	$A \leftarrow A + 1$	Increment
	If $(A = 0)$ then $(Z = 1)$	Check for zero

Accumulator design

Present state	Inputs		Next state	Flip-flop inputs		Output
A_i	B_i	C_i	A_i	JA_i	KA_i	C_{i+1}
0	0	0	0	0	X	0
0	0	1	1	1	X	0
0	1	0	1	1	X	0
0	1	1	0	0	X	1
1	0	0	1	X	0	0
1	0	1	0	X	1	1
1	1	0	0	X	1	1
1	1	1	1	X	0	1

$$JA_i = B_iC_i'p_1 + B_i'C_ip_1$$
$$KA_i = B_iC_i'p_1 + B_i'C_ip_1$$
$$C_{i+1} = A_iB_i + A_iC_i + B_iC_i$$



$JA_i = B_iC_i' + B_i'C_i$ $KA_i = B_iC_i' + B_i'C_i$ $C_{i+1} = A_iB_i + A_iC_i + B_iC_i$

Figure 9-18 Excitation table for add microoperation

Accumulator design

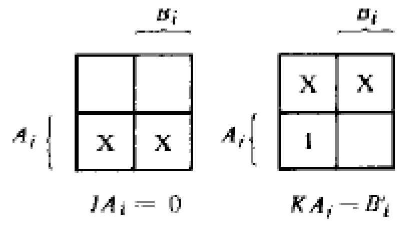
- Accumulator: clear

$$JA_i = 0$$
$$KA_i = p_2$$

- Complement

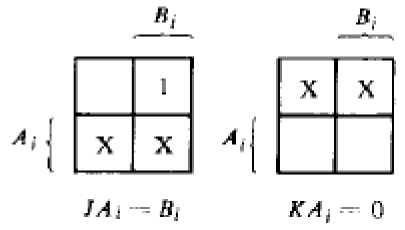
$$JA_i = p_3$$
$$KA_i = p_3$$

Present state	Input	Next state	Flip-flop inputs	
A_i	B_i	A_i	JA_i	KA_i
0	0	0	0	X
0	1	0	0	X
1	0	0	X	1
1	1	1	X	0



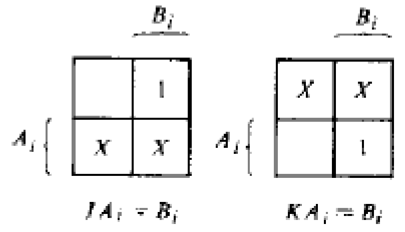
(a) AND

Present state	Input	Next state	Flip-flop inputs	
A_i	B_i	A_i	JA_i	KA_i
0	0	0	0	X
0	1	1	1	X
1	0	1	X	0
1	1	1	X	0



(b) OR

Present state	Input	Next state	Flip-flop inputs	
A_i	B_i	A_i	JA_i	KA_i
0	0	0	0	X
0	1	1	1	X
1	0	1	X	0
1	1	0	X	1



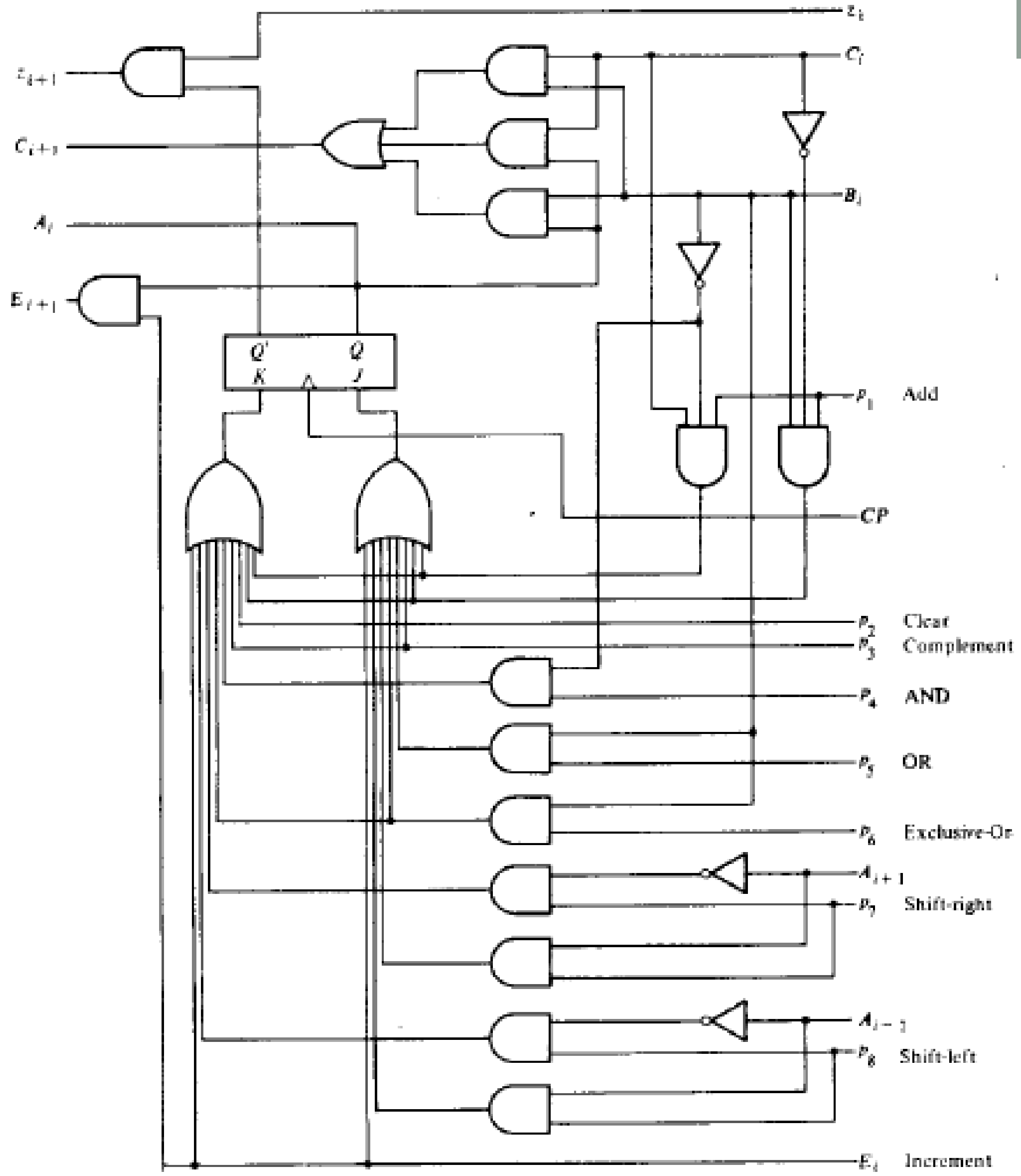
(c) Exclusive-OR

Figure 9-19 Excitation tables for logic microoperations

- One stage accumulator

$$
\begin{aligned}
JA_i &= B_iC_i'p_1 + B_i'C_ip_1 + p_3 + B_ip_5 + B_ip_6 + A_{i+1}p_7 + A_{i-1}p_8 + E_i \\
KA_i &= B_iC_i'p_1 + B_i'C_ip_1 + p_2 + p_3 + B_i'p_4 + B_ip_6 + A_{i+1}'p_7 \\
&\quad + A_{i-1}'p_8 + E_i
\end{aligned}
$$

$$
\begin{aligned}
C_{i+1} &= A_iB_i + A_iC_i + B_iC_i \\
E_{i+1} &= E_iA_i \\
z_{i+1} &= z_iA_i'
\end{aligned}
$$



z_i
z_{i+1}
C_i
C_{i+1}
B_i
A_i
E_{i+1}
Q'
Q
K
J
p_1 Add
CP
p_2 Clear
p_3 Complement
p_4 AND
p_5 OR
p_6 Exclusive-Or
A_{i+1}
p_7 Shift-right
A_{i-1}
p_8 Shift-left
E_i Increment