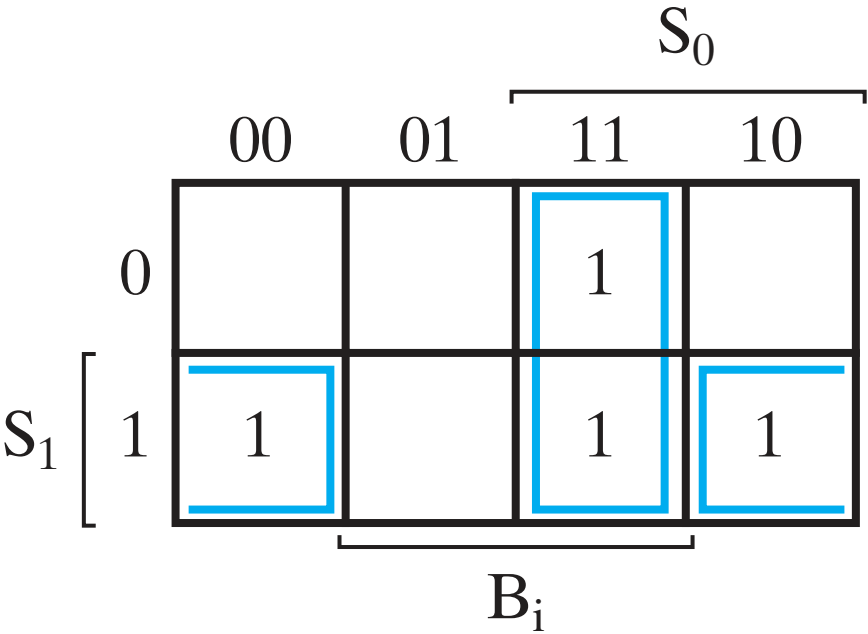


TABLE 9-1
Function Table for Arithmetic Circuit

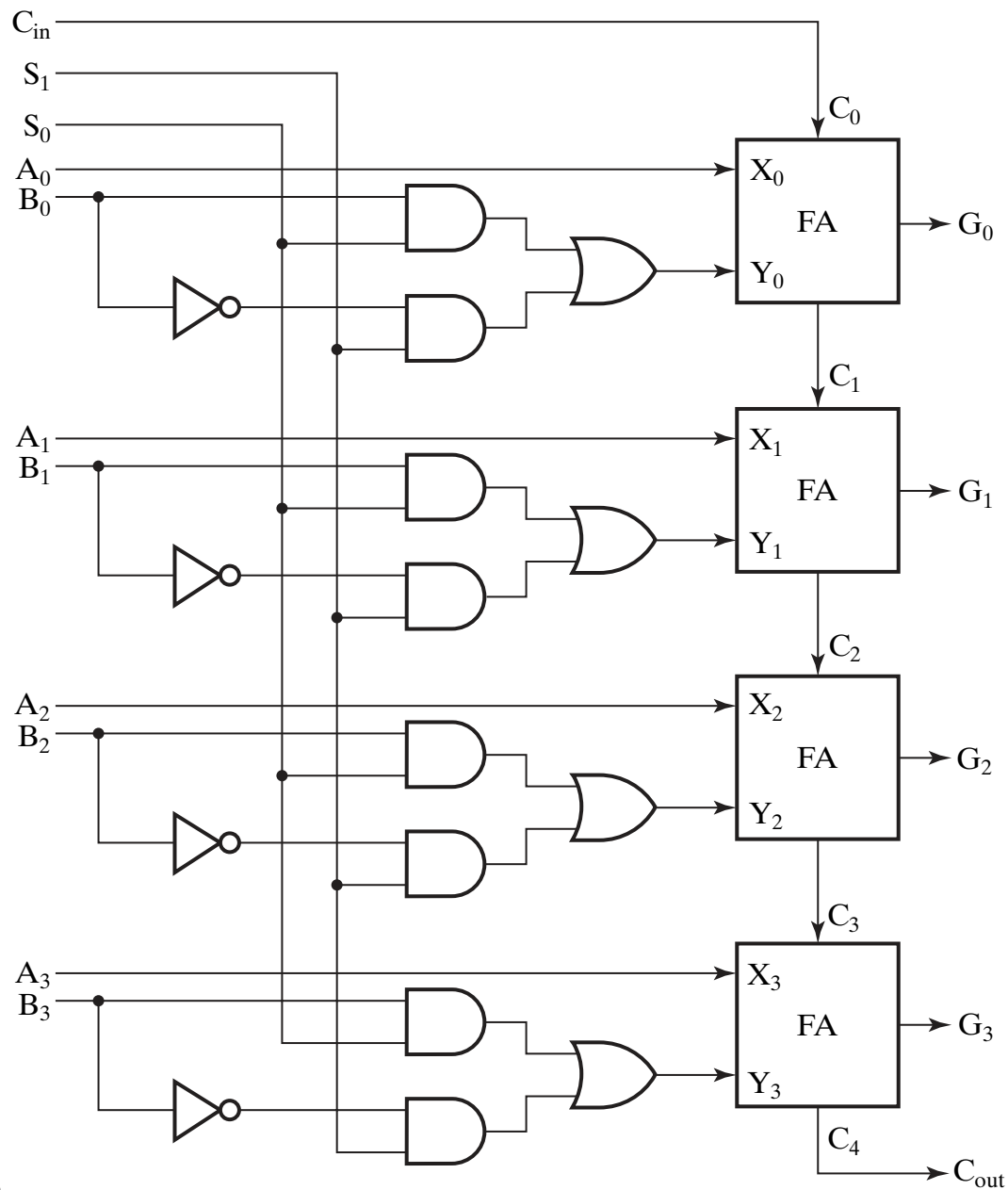
Select		Input	$G = (A \vee Y \vee C_{in})$	
S_1	S_0	Y	$C_{in} = 0$	$C_{in} = 1$
0	0	all 0s	$G = A$ (transfer)	$G = A + 1$ (increment)
0	1	B	$G = A + B$ (add)	$G = A + B + 1$
1	0	\overline{B}	$G = A + \overline{B}$	$G = A + \overline{B} + 1$ (subtract)
1	1	all 1s	$G = A - 1$ (decrement)	$G = A$ (transfer)

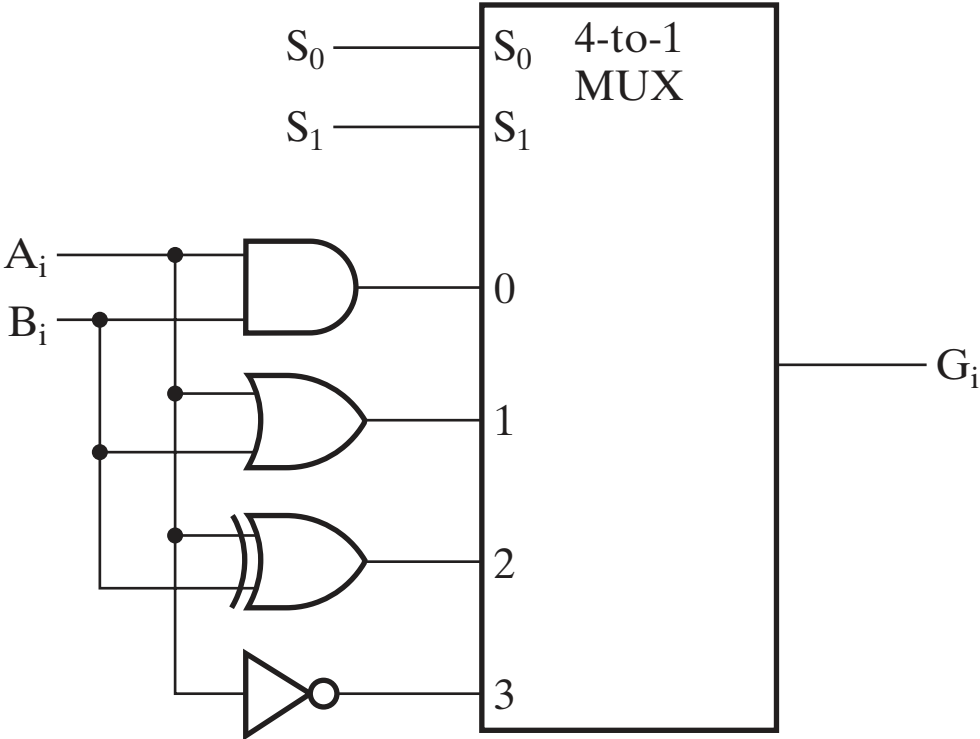
Inputs			Output
S_1	S_0	B_i	Y_i
0	0	0	0 $Y_i = 0$
0	0	1	0
0	1	0	0 $Y_i = B_i$
0	1	1	1
1	0	0	1 $Y_i = \overline{B_i}$
1	0	1	0
1	1	0	1 $Y_i = 1$
1	1	1	1

(a) Truth table



(b) Map simplification:
 $Y_i = B_i S_0 + \overline{B_i} S_1$





(a) Logic diagram

S_1	S_0	Output	Operation
0	0	$G = A \wedge B$	AND
0	1	$G = A \vee B$	OR
1	0	$G = A \oplus B$	XOR
1	1	$G = \overline{A}$	NOT

(b) Function table

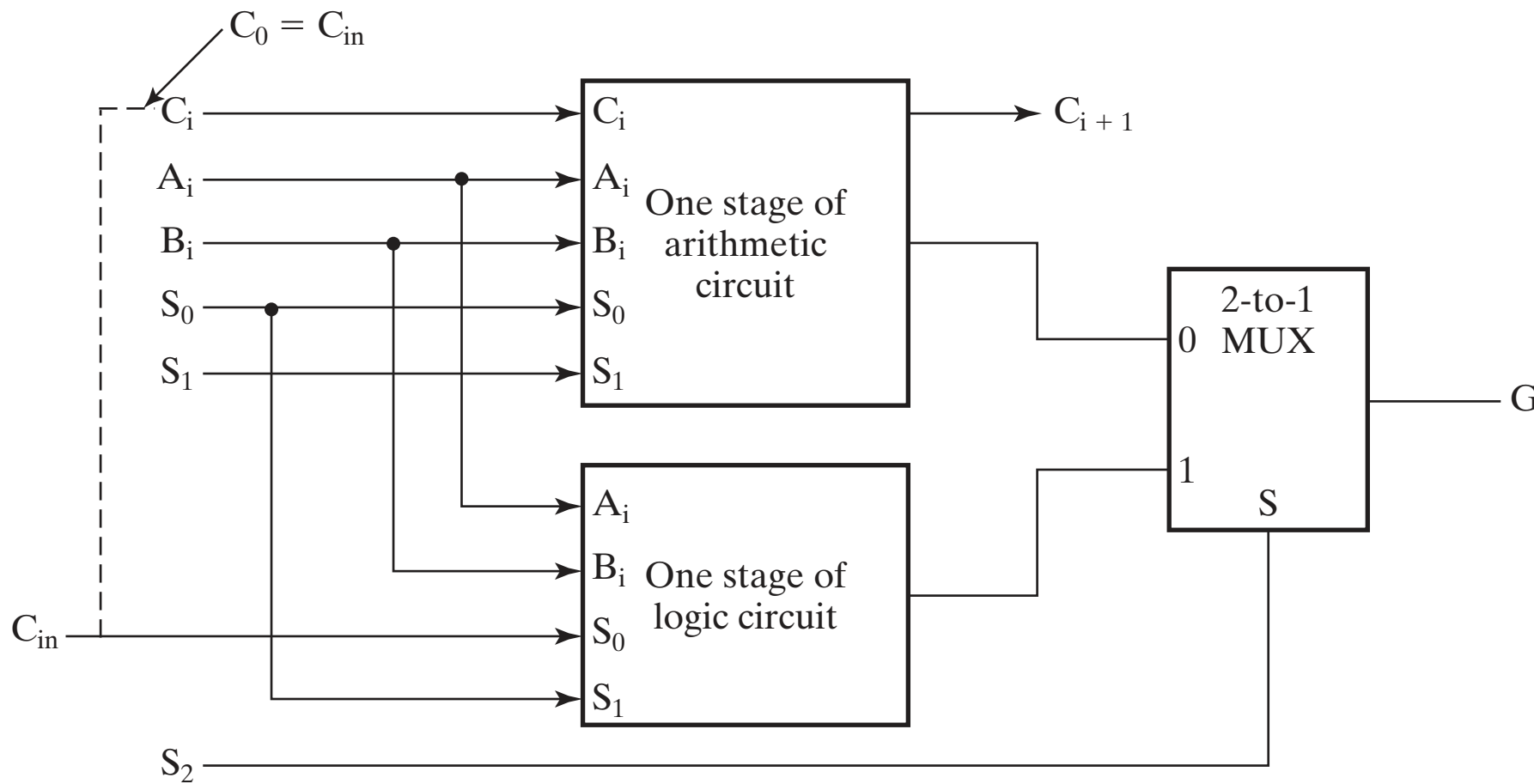


TABLE 9-2
Function Table for ALU

Operation Select				Operation	Function
S_2	S_1	S_0	C_{in}		
0	0	0	0	$G = A$	Transfer A
0	0	0	1	$G = A + 1$	Increment A
0	0	1	0	$G = A + B$	Addition
0	0	1	1	$G = A + B + 1$	Add with carry input of 1
0	1	0	0	$G = A + \overline{B}$	A plus 1s complement of B
0	1	0	1	$G = A + \overline{B} + 1$	Subtraction
0	1	1	0	$G = A - 1$	Decrement A
0	1	1	1	$G = A$	Transfer A
1	X	0	0	$G = A \wedge B$	AND
1	X	0	1	$G = A \vee B$	OR
1	X	1	0	$G = A \oplus B$	XOR
1	X	1	1	$G = \overline{A}$	NOT (1s complement)

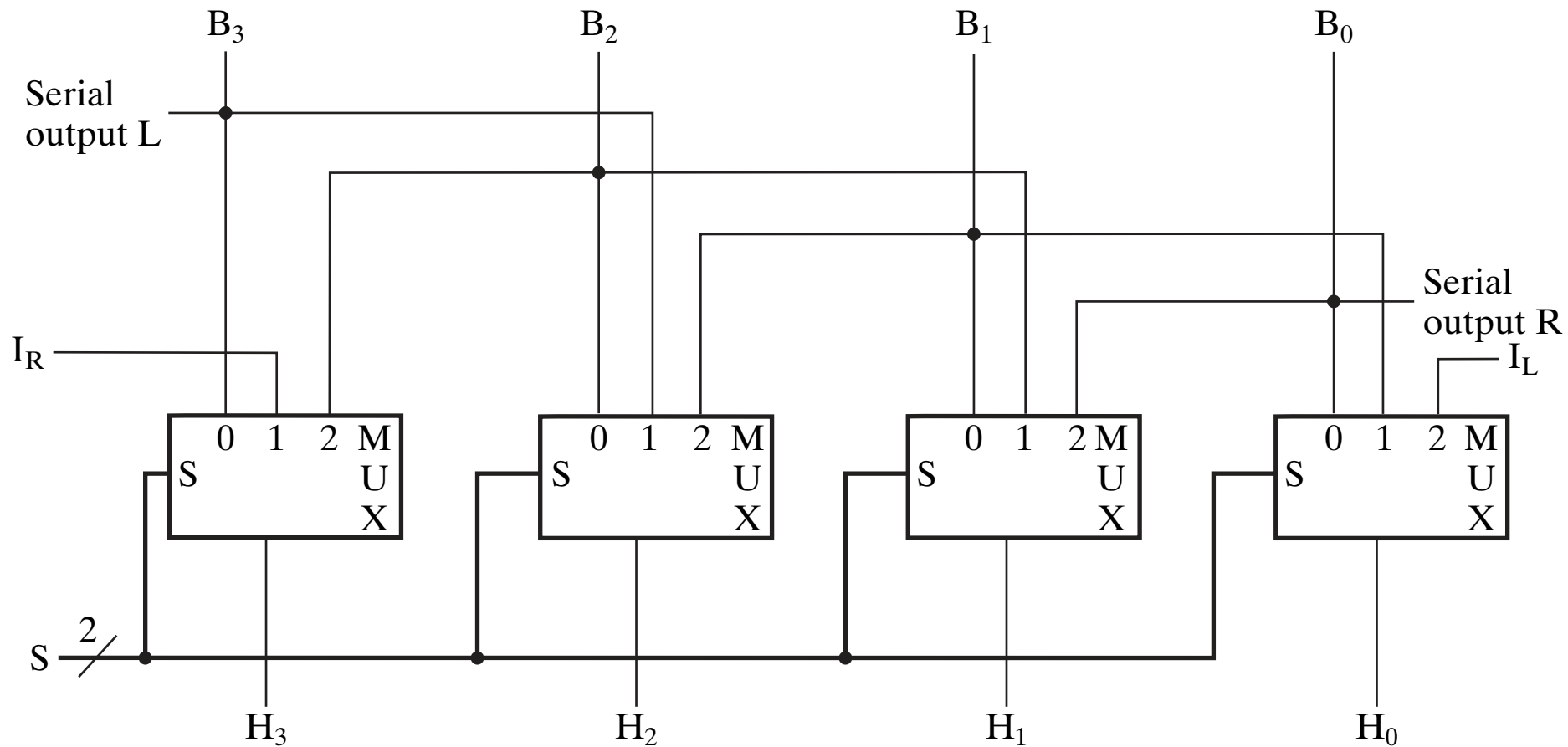
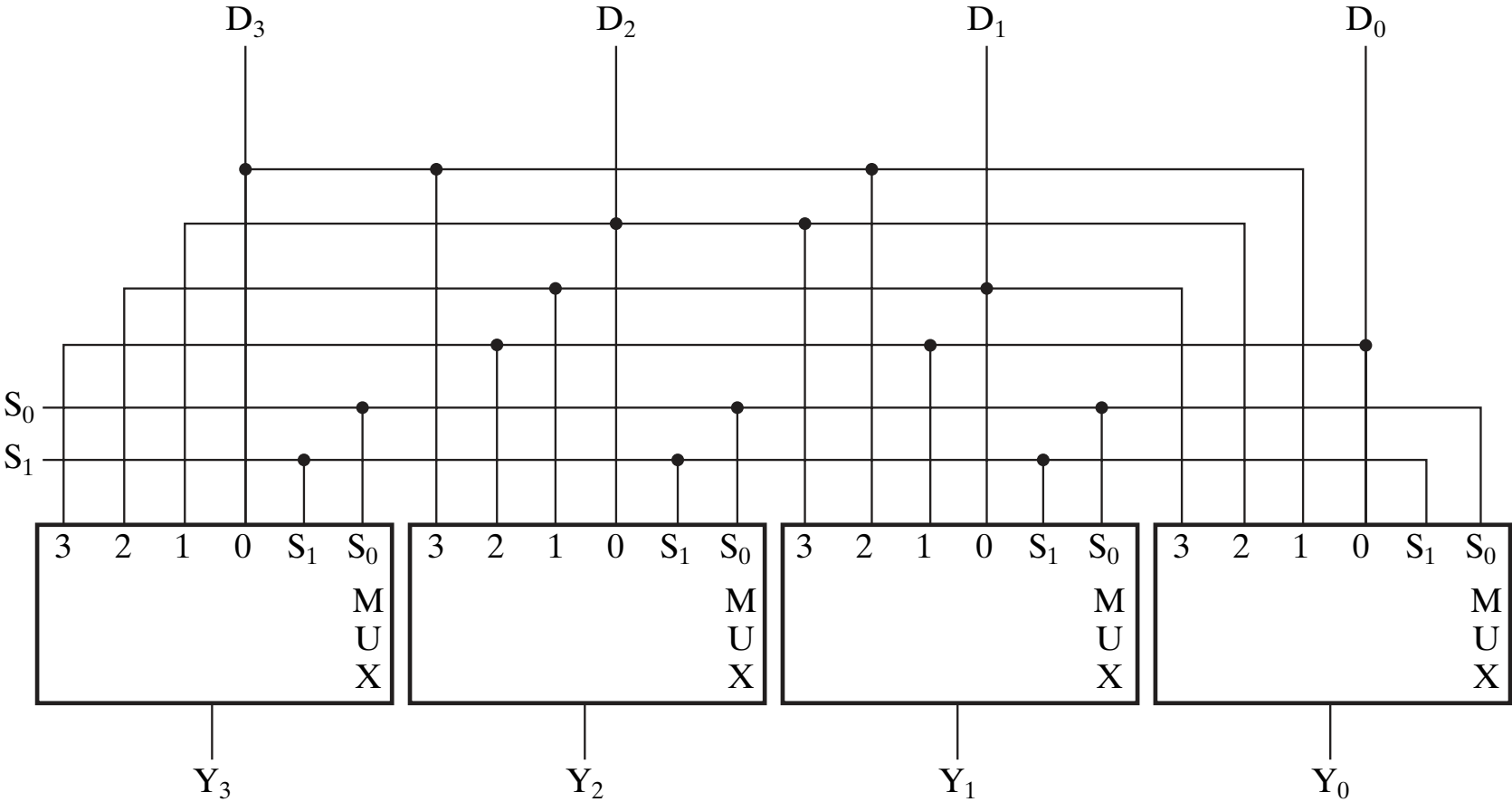
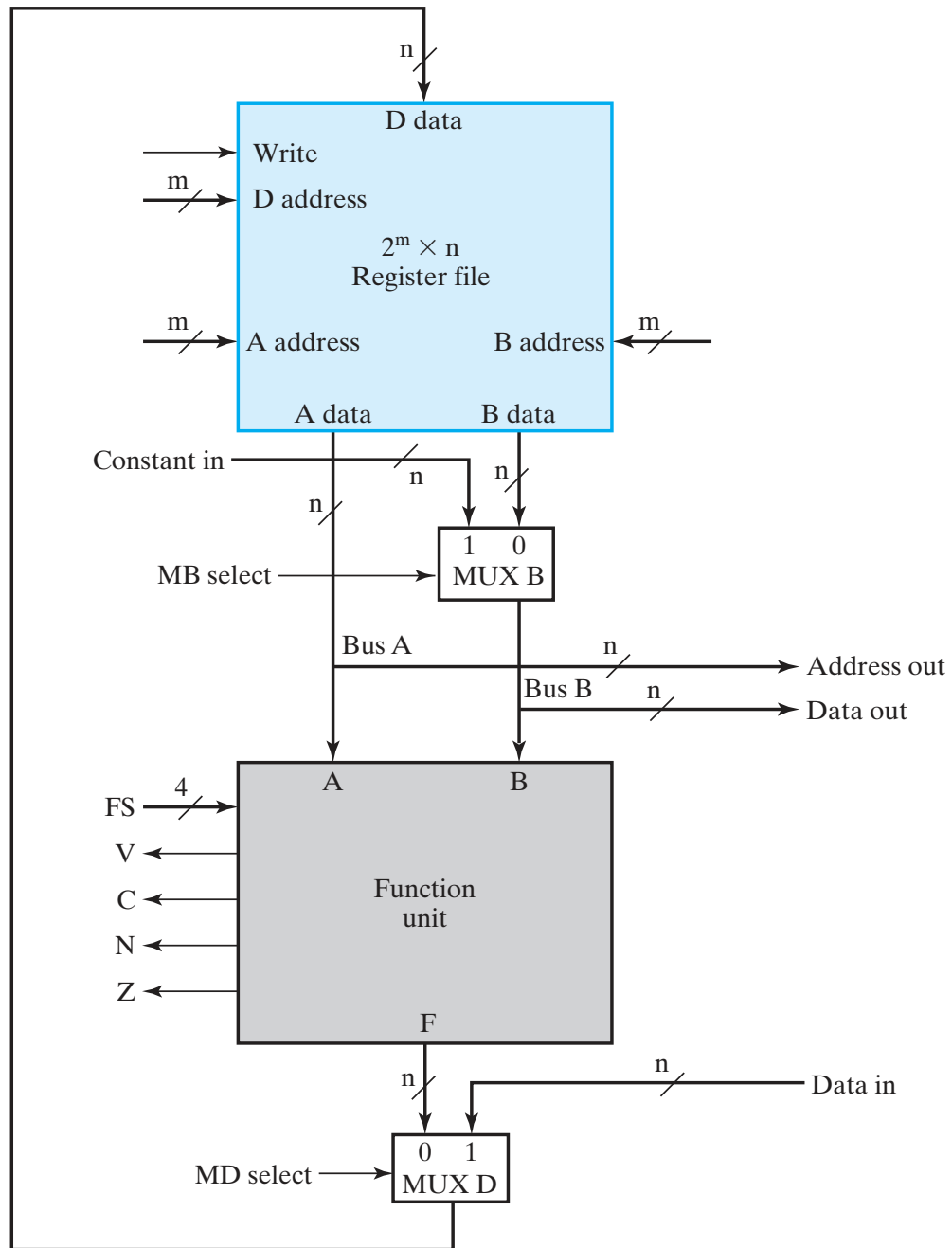


TABLE 9-3
Function Table for 4-Bit Barrel Shifter

Select		Output				Operation
S_1	S_0	Y_3	Y_2	Y_1	Y_0	
0	0	D_3	D_2	D_1	D_0	No rotation
0	1	D_2	D_1	D_0	D_3	Rotate one position
1	0	D_1	D_0	D_3	D_2	Rotate two positions
1	1	D_0	D_3	D_2	D_1	Rotate three positions

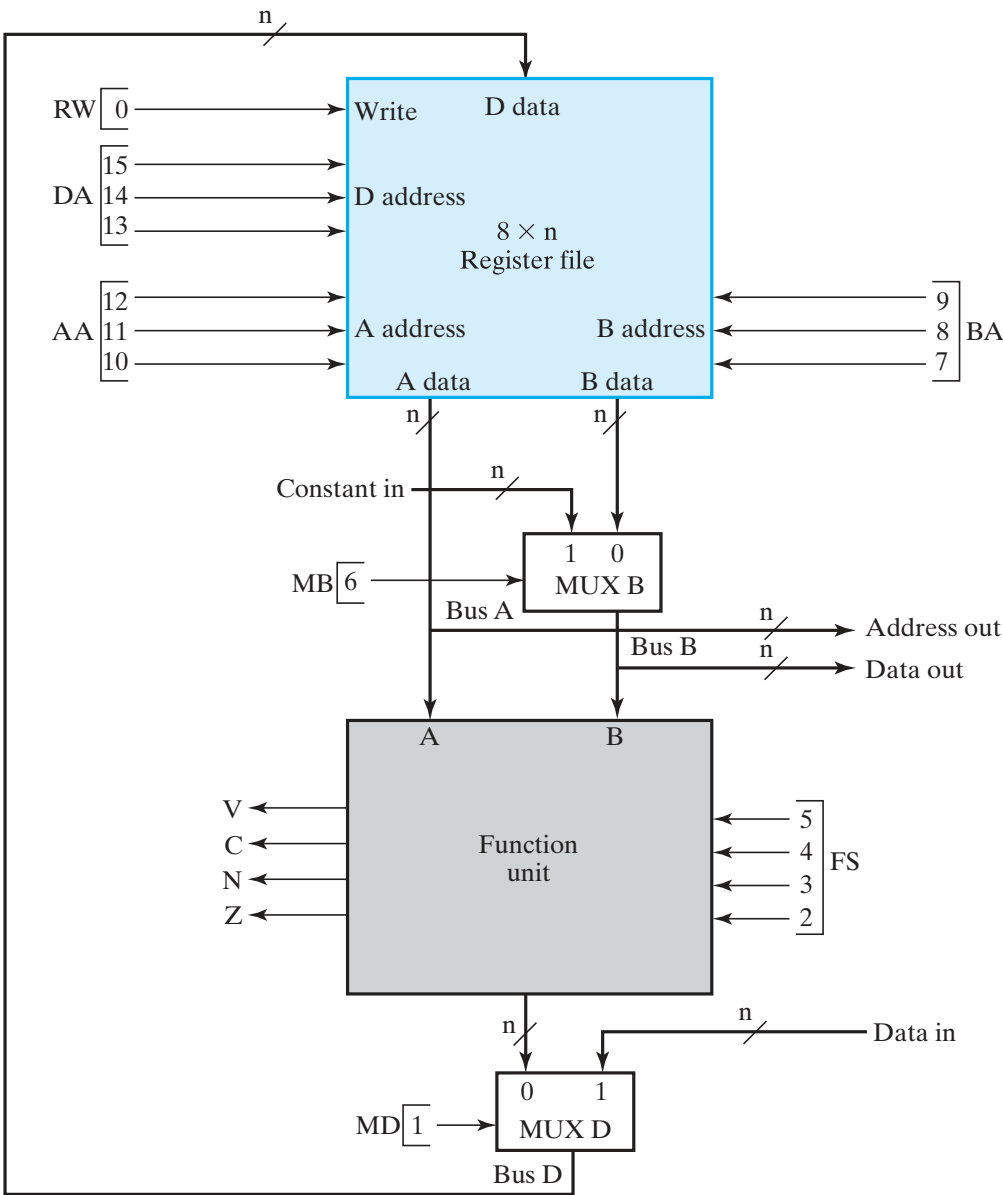




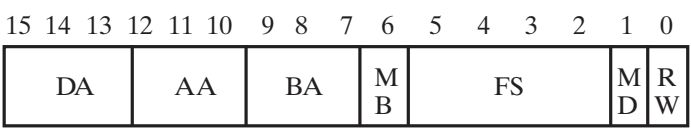
□ **TABLE 9-4**

***G* Select, *H* Select, and *MF* Select Codes Defined
in Terms of *FS* Codes**

FS(3:0)	MF Select	G Select(3:0)	H Select(3:0)	Microoperation
0000	0	0000	XX	$F = A$
0001	0	0001	XX	$F = A + 1$
0010	0	0010	XX	$F = A + B$
0011	0	0011	XX	$F = A + B + 1$
0100	0	0100	XX	$F = A + \overline{B}$
0101	0	0101	XX	$F = A + \overline{B} + 1$
0110	0	0110	XX	$F = A - 1$
0111	0	0111	XX	$F = A$
1000	0	1X00	XX	$F = A \wedge B$
1001	0	1X01	XX	$F = A \vee B$
1010	0	1X10	XX	$F = A \oplus B$
1011	0	1X11	XX	$F = \overline{A}$
1100	1	XXXX	0 0	$F = B$
1101	1	XXXX	0 1	$F = \text{sr } B$
1110	1	XXXX	1 0	$F = \text{sl } B$



(a) Block diagram



(b) Control word

TABLE 9-5
Encoding of Control Word for the Datapath

DA, AA, BA		MB		FS		MD		RW	
Function	Code	Function	Code	Function	Code	Function	Code	Function	Code
$R0$	000	Register	0	$F = A$	0000	Function	0	No Write	0
$R1$	001	Constant	1	$F = A + 1$	0001	Data in	1	Write	1
$R2$	010			$F = A + B$	0010				
$R3$	011			$F = A + \overline{B} + 1$	0011				
$R4$	100			$F = A + \overline{B}$	0100				
$R5$	101			$F = A + \overline{B} + 1$	0101				
$R6$	110			$F = A - 1$	0110				
$R7$	111			$F = A$	0111				
				$F = A \wedge B$	1000				
				$F = A \vee B$	1001				
				$F = A \oplus B$	1010				
				$F = \overline{A}$	1011				
				$F = B$	1100				
				$F = \text{sr } B$	1101				
				$F = \text{sl } B$	1110				

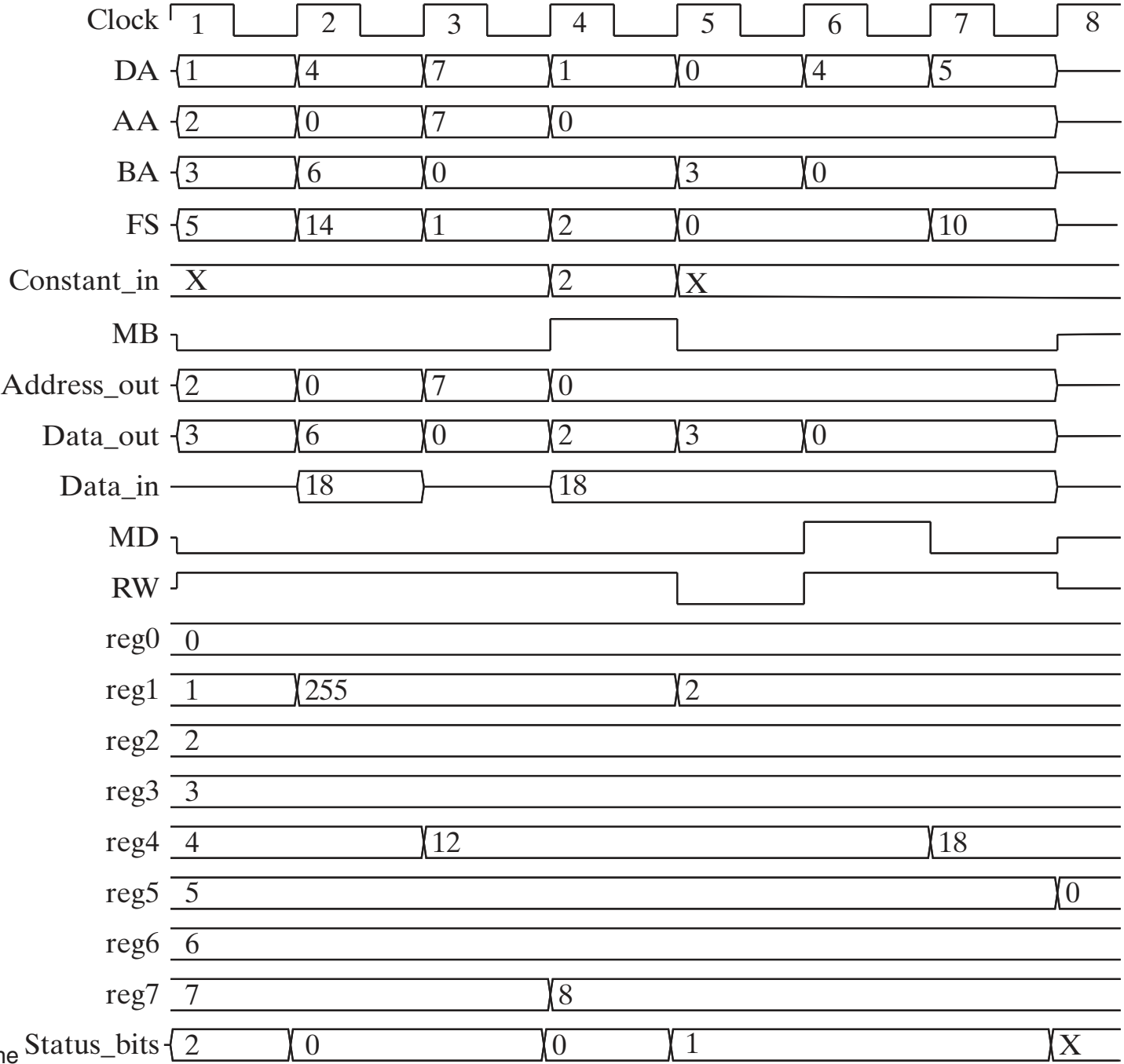
TABLE 9-6
Examples of Microoperations for the Datapath, Using Symbolic Notation

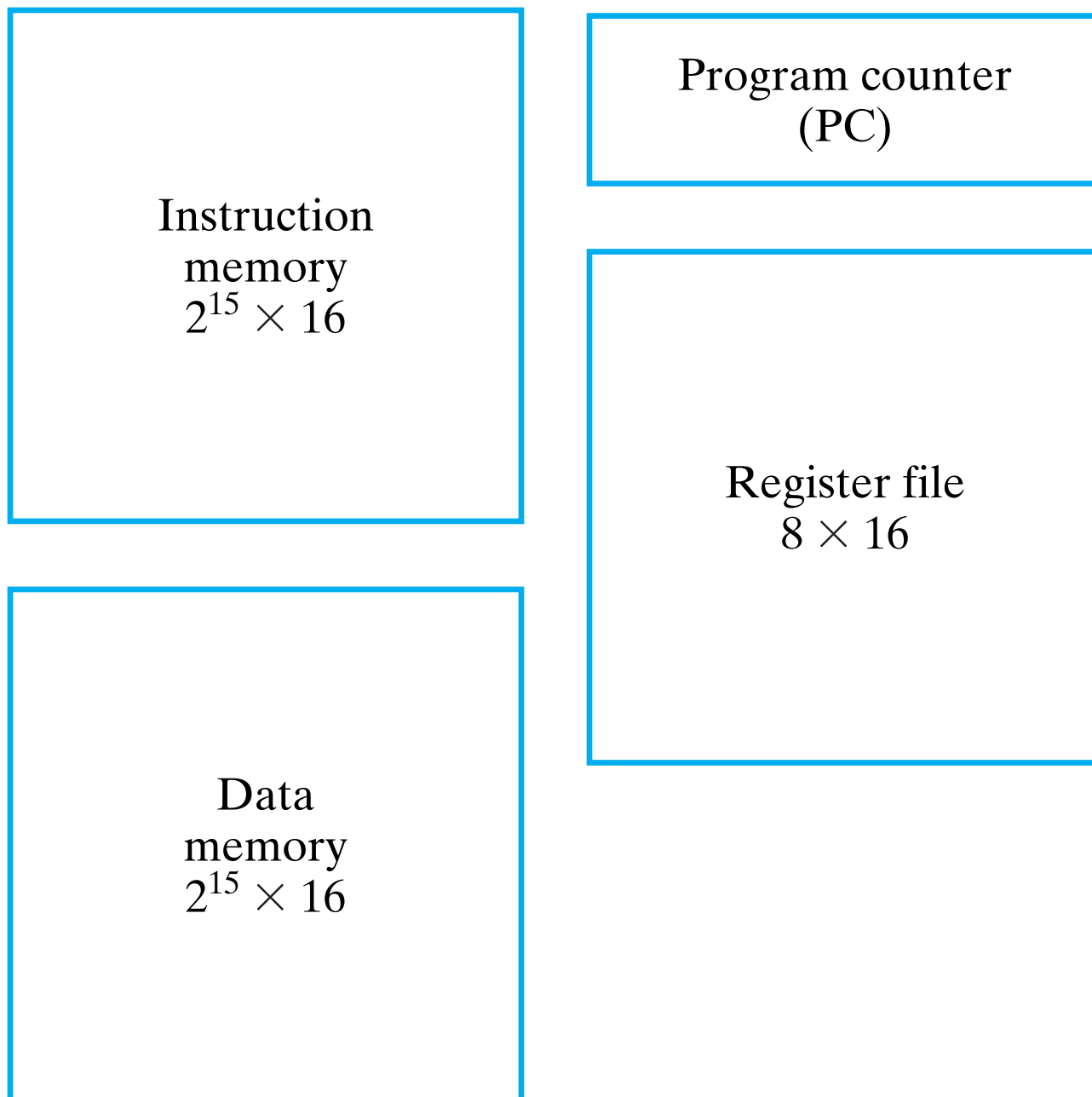
Micro-operation	DA	AA	BA	MB	FS	MD	RW
$R1 \leftarrow R2 - R3$	$R1$	$R2$	$R3$	Register	$F = A + \overline{B} + 1$	Function	Write
$R4 \leftarrow \text{sl } R6$	$R4$	—	$R6$	Register	$F = \text{sl } B$	Function	Write
$R7 \leftarrow R7 + 1$	$R7$	$R7$	—	—	$F = A + 1$	Function	Write
$R1 \leftarrow R0 + 2$	$R1$	$R0$	—	Constant	$F = A + B$	Function	Write
$\text{Data out} \leftarrow R3$	—	—	$R3$	Register	—	—	No Write
$R4 \leftarrow \text{Data in}$	$R4$	—	—	—	—	Data in	Write
$R5 \leftarrow 0$	$R5$	$R0$	$R0$	Register	$F = A \oplus B$	Function	Write

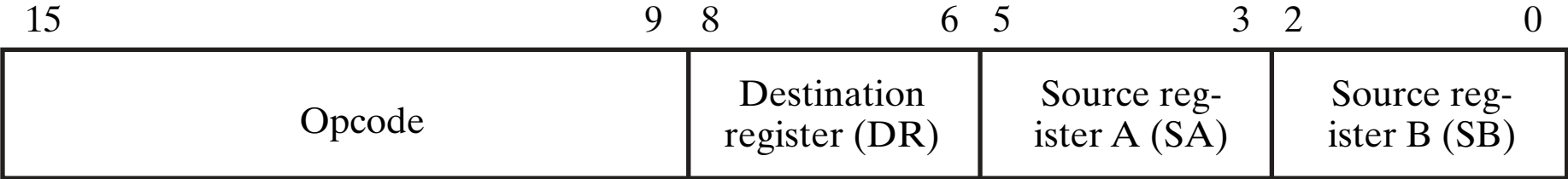
TABLE 9-7

Examples of Microoperations from Table 9-6, Using Binary Control Words

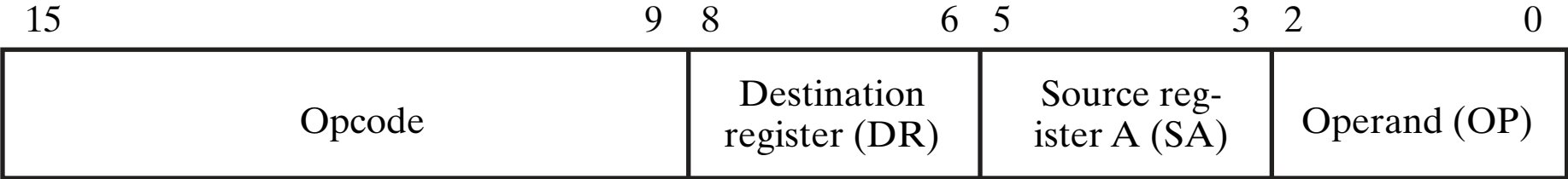
Micro-operation	DA	AA	BA	MB	FS	MD	RW
$R1 \leftarrow R2 - R3$	001	010	011	0	0101	0	1
$R4 \leftarrow s1 R6$	100	XXX	110	0	1110	0	1
$R7 \leftarrow R7 + 1$	111	111	XXX	X	0001	0	1
$R1 \leftarrow R0 + 2$	001	000	XXX	1	0010	0	1
Data out $\leftarrow R3$	XXX	XXX	011	0	XXXX	X	0
$R4 \leftarrow$ Data in	100	XXX	XXX	X	XXXX	1	1
$R5 \leftarrow 0$	101	000	000	0	1010	0	1



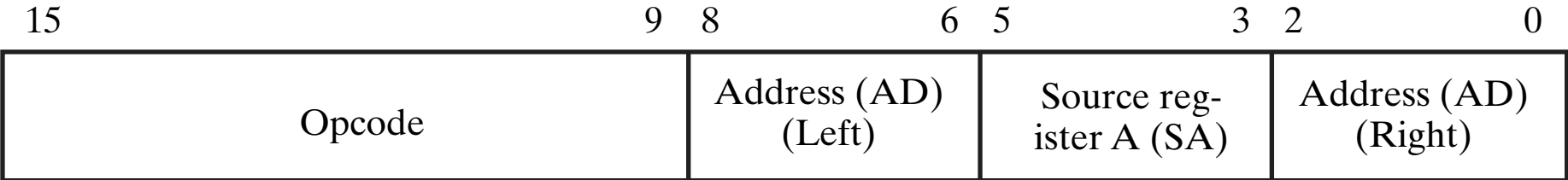




(a) Register



(b) Immediate



(c) Jump and Branch

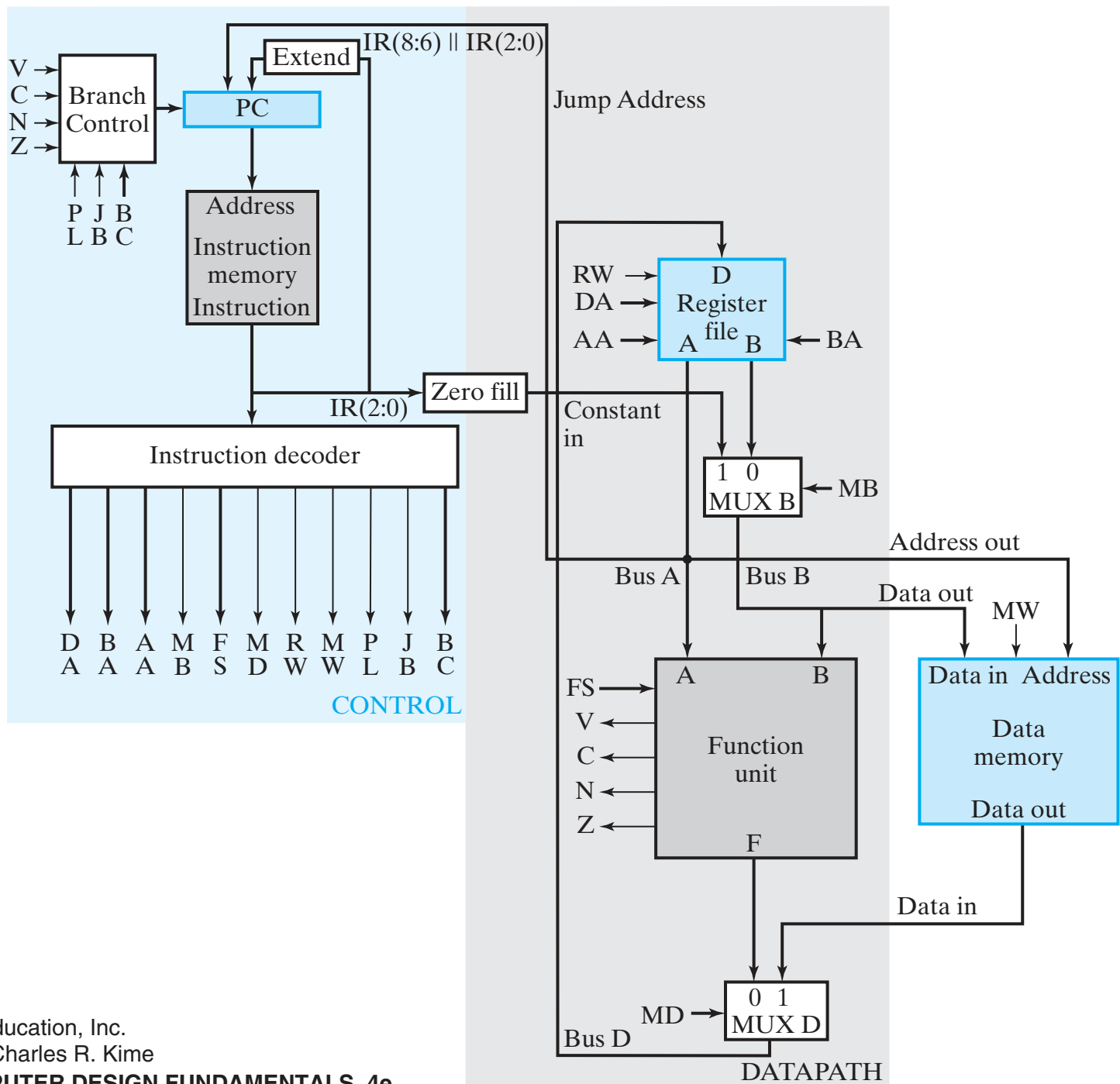
TABLE 9-8
Instruction Specifications for the Simple Computer

Instruction	Opcode	Mne- monic	Format	Description	Status Bits
Move A	0000000	MOVA	RD, RA	$R[DR] \leftarrow R[SA]^*$	N, Z
Increment	0000001	INC	RD, RA	$R[DR] \leftarrow R[SA] + 1^*$	N, Z
Add	0000010	ADD	RD, RA, RB	$R[DR] \leftarrow R[SA] + R[SB]^*$	N, Z
Subtract	0000101	SUB	RD, RA, RB	$R[DR] \leftarrow R[SA] - R[SB]^*$	N, Z
Decrement	0000110	DEC	RD, RA	$R[DR] \leftarrow R[SA] - 1^*$	N, Z
AND	0001000	AND	RD, RA, RB	$R[DR] \leftarrow R[SA] \wedge R[SB]^*$	N, Z
OR	0001001	OR	RD, RA, RB	$R[DR] \leftarrow R[SA] \vee R[SB]^*$	N, Z
Exclusive OR	0001010	XOR	RD, RA, RB	$R[DR] \leftarrow R[SA] \oplus R[SB]^*$	N, Z
NOT	0001011	NOT	RD, RA	$R[DR] \leftarrow \overline{R[SA]}^*$	N, Z
Move B	0001100	MOVB	RD, RB	$R[DR] \leftarrow R[SB]^*$	
Shift Right	0001101	SHR	RD, RB	$R[DR] \leftarrow sr R[SB]^*$	
Shift Left	0001110	SHL	RD, RB	$R[DR] \leftarrow sl R[SB]^*$	
Load Immediate	1001100	LDI	RD, OP	$R[DR] \leftarrow zf OP^*$	
Add Immediate	1000010	ADI	RD, RA, OP	$R[DR] \leftarrow R[SA] + zf OP^*$	N, Z
Load	0010000	LD	RD, RA	$R[DR] \leftarrow M[SA]^*$	
Store	0100000	ST	RA, RB	$M[SA] \leftarrow R[SB]^*$	
Branch on Zero	1100000	BRZ	RA, AD	if ($R[SA] = 0$) $PC \leftarrow PC + se AD$, if ($R[SA] \neq 0$) $PC \leftarrow PC + 1$	N, Z
Branch on Negative	1100001	BRN	RA, AD	if ($R[SA] < 0$) $PC \leftarrow PC + se AD$, if ($R[SA] \geq 0$) $PC \leftarrow PC + 1$	N, Z
Jump	1110000	JMP	RA	$PC \leftarrow R[SA]$	

* For all of these instructions, $PC \leftarrow PC + 1$ is also executed to prepare for the next cycle.

TABLE 9-9
Memory Representation of Instructions and Data

Decimal Address	Memory Contents	Decimal Opcode	Other Fields	Operation
25		5 (Subtract)	DR:1, SA:2, SB:3	$R1 \leftarrow R2 - R3$
	0000101 001 010 011			
35		32 (Store)	SA:4, SB:5	$M[R4] \leftarrow R5$
	0100000 000 100 101			
45		66 (Add Immediate)	DR:2, SA:7, OP:3	$R2 \leftarrow R7 + 3$
	1000010 010 111 011			
55		96 (Branch on Zero)	AD: 44, SA:6	If $R6 = 0$, $PC \leftarrow PC - 20$
	1100000 101 110 100			
70		Data = 192. After execution of instruction in 35, Data = 80.		
	00000000011000000			



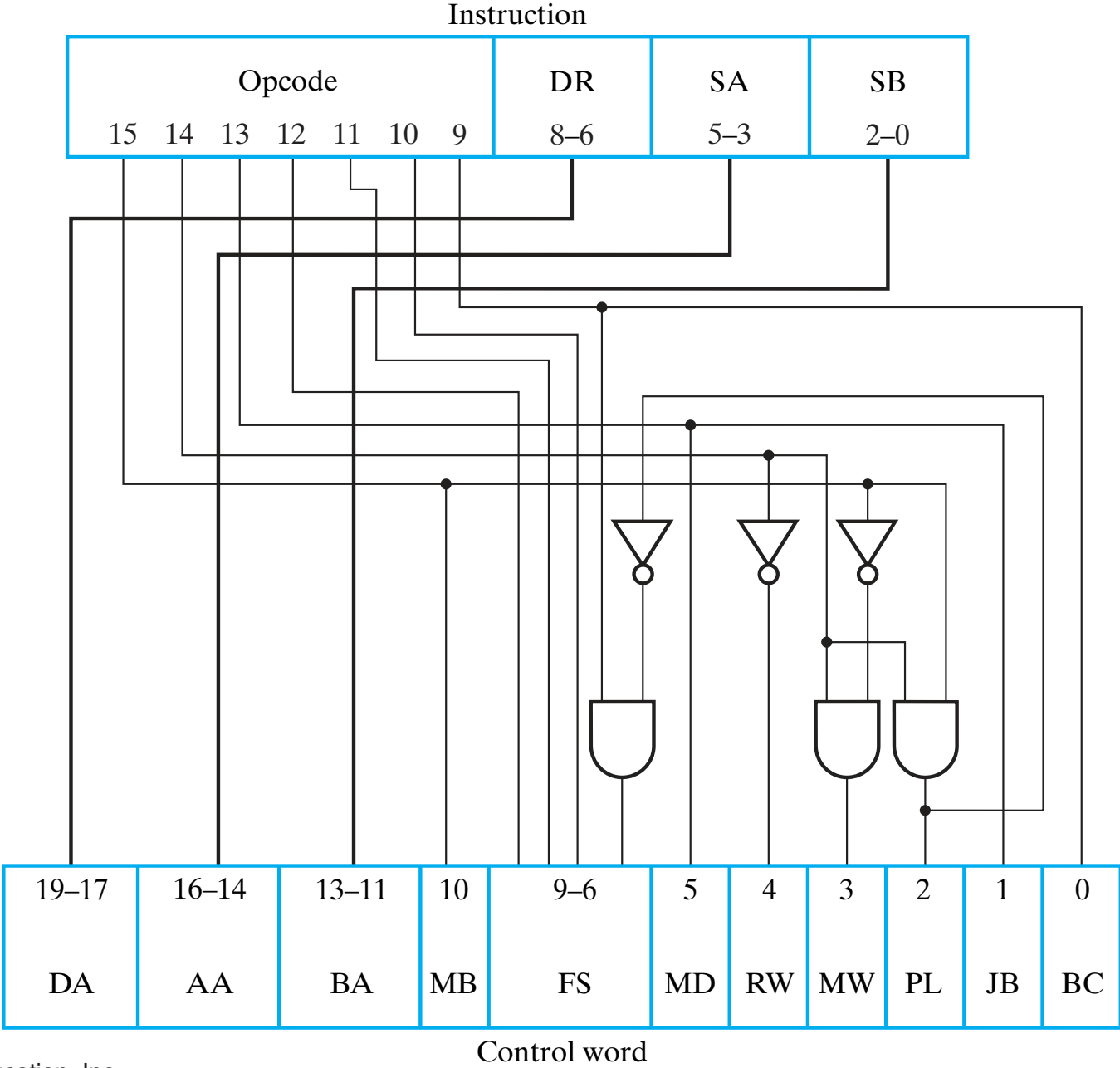
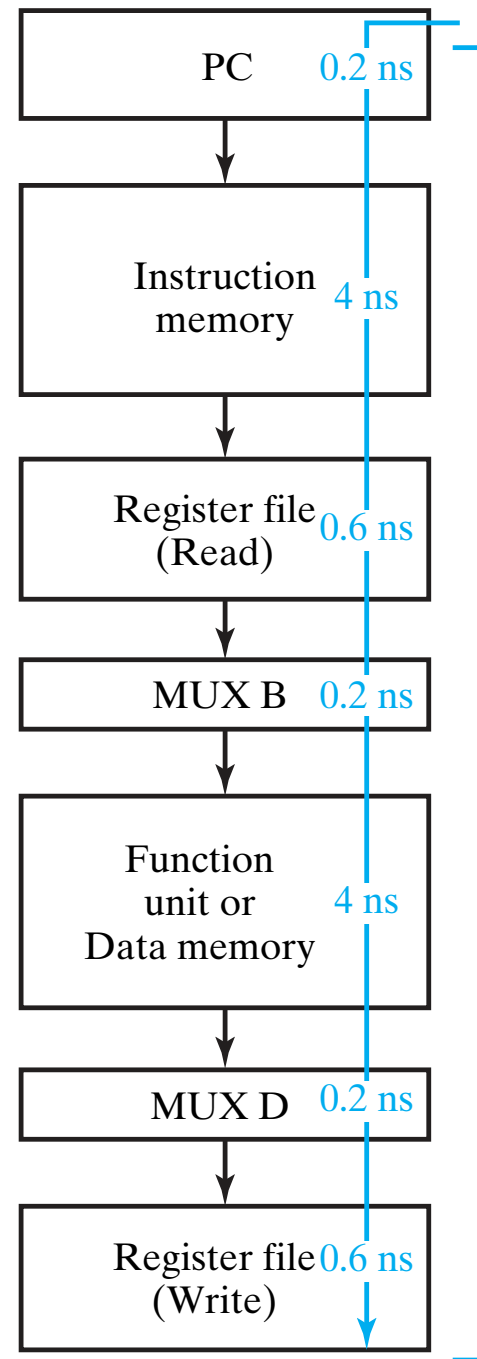


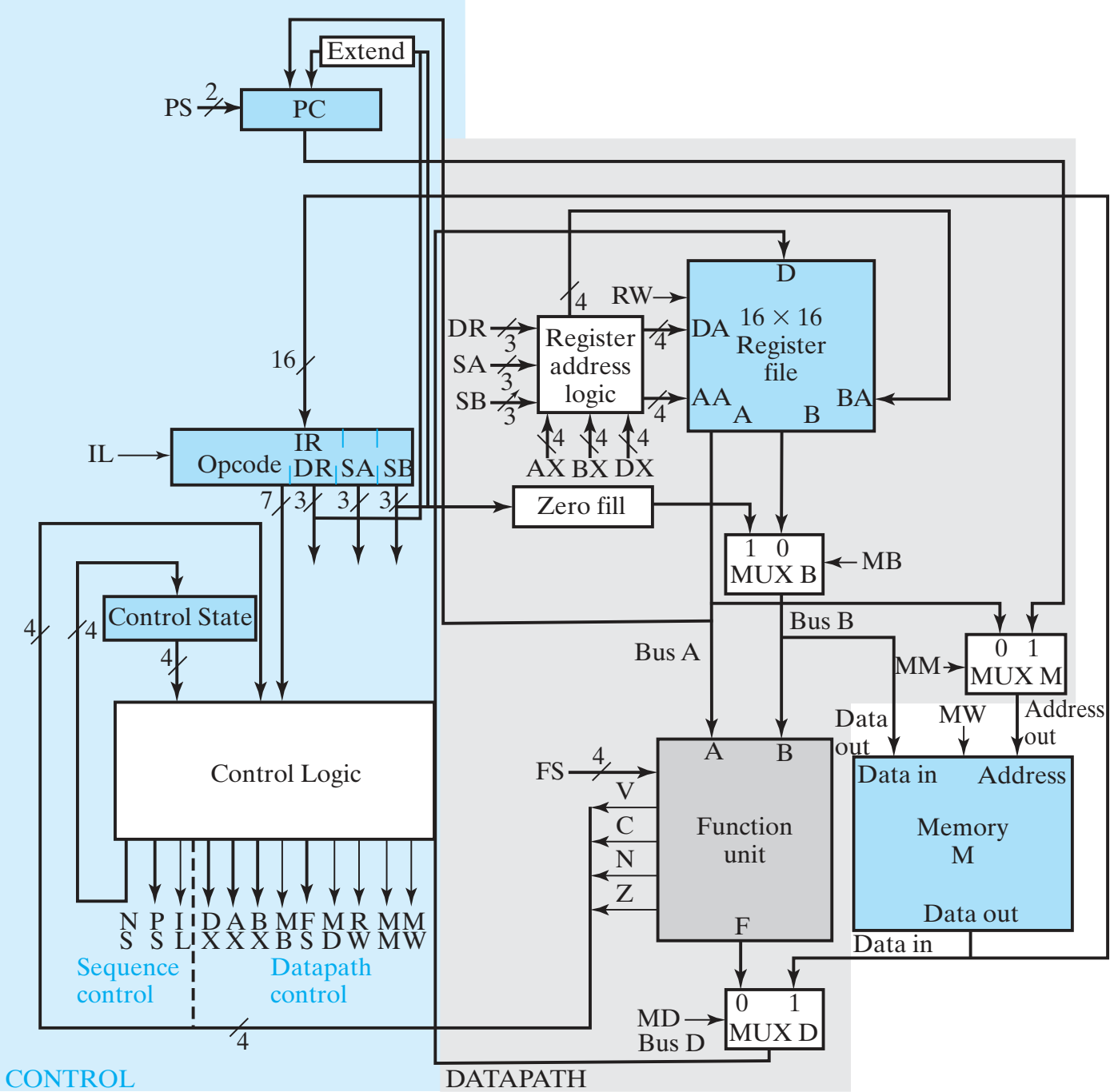
TABLE 9-10
Truth Table for Instruction Decoder Logic

Instruction Function Type	Instruction Bits				Control Word Bits						
	15	14	13	9	MB	MD	RW	MW	PL	JB	BC
Function-unit operations using registers	0	0	0	X	0	0	1	0	0	X	X
Memory read	0	0	1	X	0	1	1	0	0	X	X
Memory write	0	1	0	X	0	X	0	1	0	X	X
Function-unit operations using register and constant	1	0	0	X	1	0	1	0	0	X	X
Conditional branch on zero (<i>Z</i>)	1	1	0	0	X	X	0	0	1	0	0
Conditional branch on negative (<i>N</i>)	1	1	0	1	X	X	0	0	1	0	1
Unconditional jump	1	1	1	X	X	X	0	0	1	1	X

TABLE 9-11
Six Instructions for the Single-Cycle Computer

Operation Code	Symbolic Name	Format	Description	Function	MB	MD	RW	MW	PL	JB	BC
1000010	ADI	Immediate	Add immediate operand	$R[DR] \leftarrow R[SA] + zf\ I(2:0)$	1	0	1	0	0	0	0
0010000	LD	Register	Load memory content into register	$R[DR] \leftarrow M[R[SA]]$	0	1	1	0	0	1	0
0100000	ST	Register	Store register content in memory	$M[R[SA]] \leftarrow R[SB]$	0	1	0	1	0	0	0
0001110	SL	Register	Shift left	$R[DR] \leftarrow sl\ R[SB]$	0	0	1	0	0	1	0
0001011	NOT	Register	Complement register	$R[DR] \leftarrow \overline{R[SA]}$	0	0	1	0	0	0	1
1100000	BRZ	Jump/Branch	If $R[SA] = 0$, branch to $PC + se\ AD$	If $R[SA] = 0$, $PC \leftarrow PC + se\ AD$ If $R[SA] \neq 0$, $PC \leftarrow PC + 1$	1	0	0	0	1	0	0





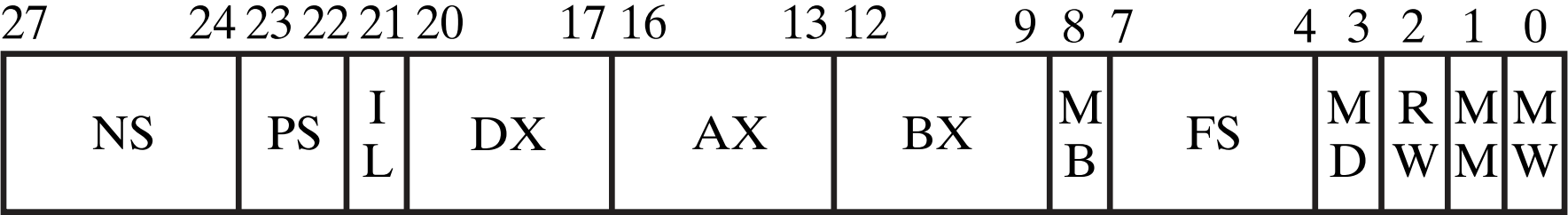


TABLE 9-12
Control-Word Information for Datapath

DX	AX	BX	Code	MB	Code	FS	Code	MD	RW	MM	MW	Code
$R[DR]$	$R[SA]$	$R[SB]$	0XXX	Register	0	$F = A$	0000	FnUt	No Write	Address out	No Write	0
$R8$	$R8$	$R8$	1000	Constant	1	$F = A + 1$	0001	Data in	Write	PC	Write	1
$R9$	$R9$	$R9$	1001			$F = A + B$	0010					
$R10$	$R10$	$R10$	1010			Unused	0011					
$R11$	$R11$	$R11$	1011			Unused	0100					
$R12$	$R12$	$R12$	1100			$F = A + \overline{B} + 1$	0101					
$R13$	$R13$	$R13$	1101			$F = A - 1$	0110					
$R14$	$R14$	$R14$	1110			Unused	0111					
$R15$	$R15$	$R15$	1111			$F = A \wedge B$	1000					
						$F = A \vee B$	1001					
						$F = A \oplus B$	1010					
						$F = \overline{A}$	1011					
						$F = B$	1100					
						$F = \text{sr } B$	1101					
						$F = \text{sl } B$	1110					
						Unused	1111					

TABLE 9-13
Control Information for Sequence Control

NS		PS		IL	
Next State		Action	Code	Action	Code
Gives next state of control state register		Hold PC	00	No load	0
		Inc PC	01	Load IR	1
		Branch	10		
		Jump	11		

$PC \leftarrow PC + 1$
 Σ transition conditions on merged arcs

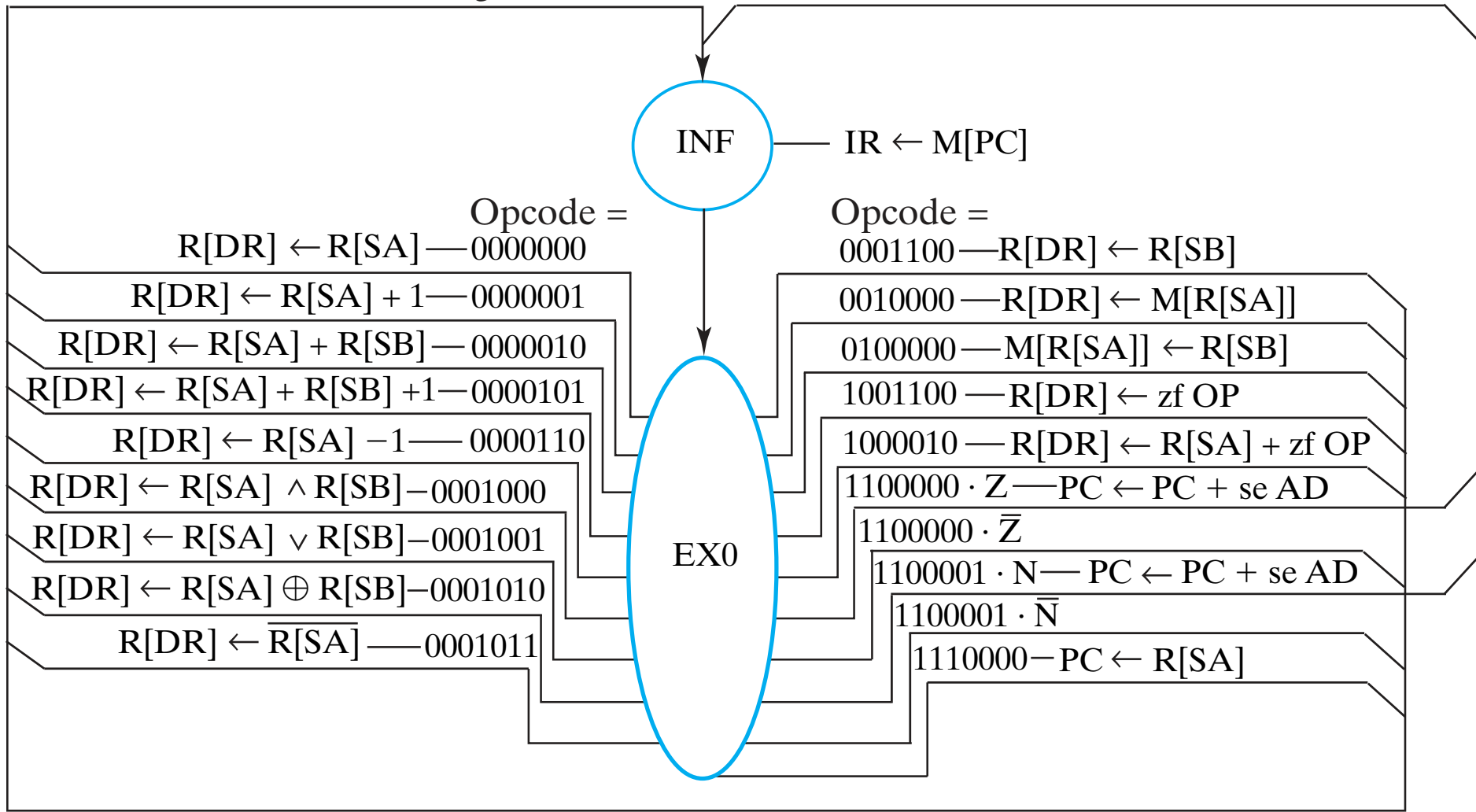
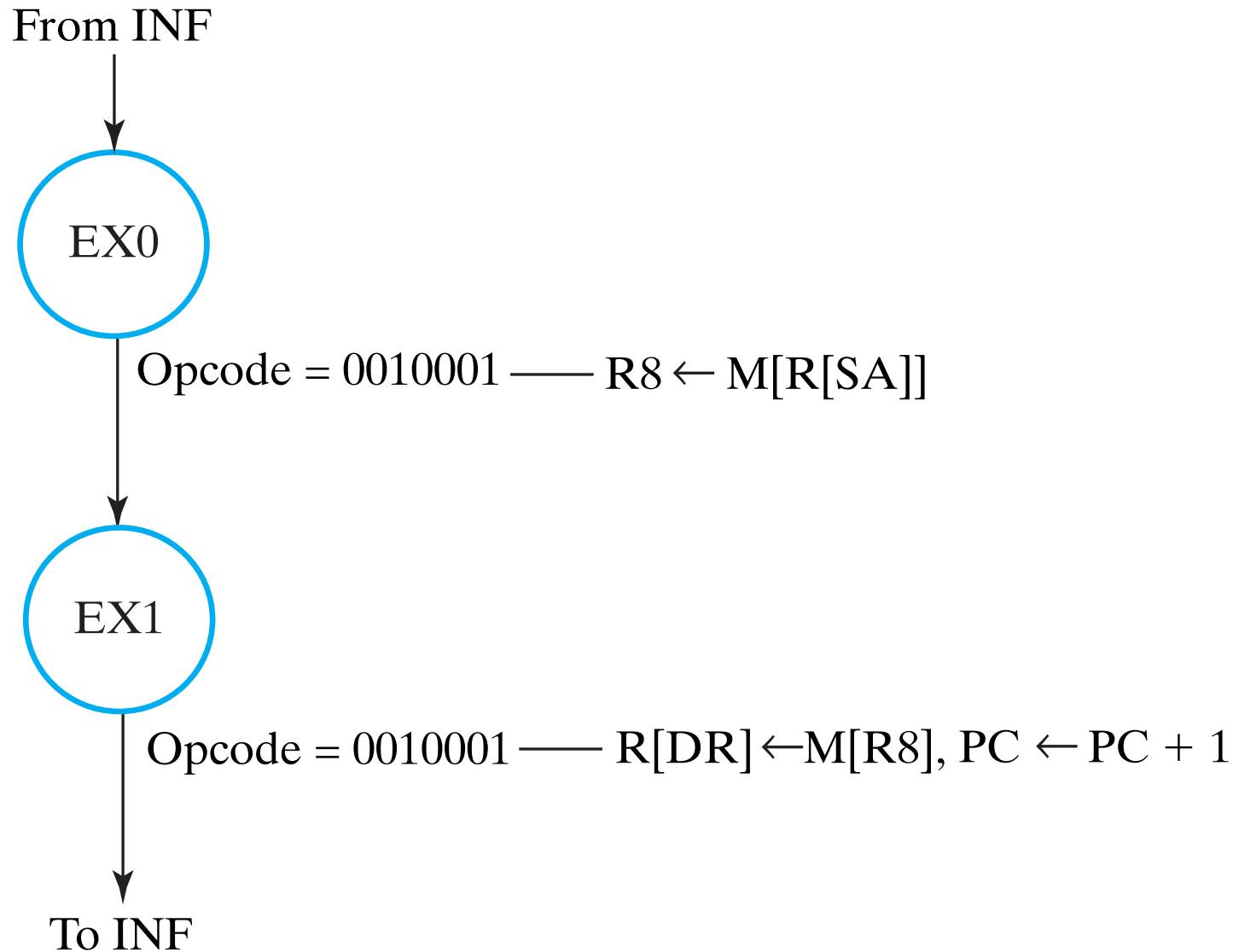


TABLE 9-14
 State Table for Two-Cycle Instructions

State	Inputs		Next State	Outputs											Comments
	Opcode	VCNZ		I L	P S	DX	AX	BX	M B	FS	M D	R W	M M	M W	
INF	XXXXXXX	XXXX	EX0	1	00	XXXX	XXXX	XXXX	X	XXXX	X	0	1	0	$IR \leftarrow M[PC]$
EX0	0000000	XXXX	INF	0	01	0XXX	0XXX	XXXX	X	0000	0	1	X	0	MOVA $R[DR] \leftarrow R[SA]^*$
EX0	0000001	XXXX	INF	0	01	0XXX	0XXX	XXXX	X	0001	0	1	X	0	INC $R[DR] \leftarrow R[SA] + 1^*$
EX0	0000010	XXXX	INF	0	01	0XXX	0XXX	0XXX	0	0010	0	1	X	0	ADD $R[DR] \leftarrow R[SA] + R[SB]^*$
EX0	0000101	XXXX	INF	0	01	0XXX	0XXX	0XXX	0	0101	0	1	X	0	SUB $R[DR] \leftarrow R[SA] + \overline{R[SB]} + 1^*$
EX0	0000110	XXXX	INF	0	01	0XXX	0XXX	XXXX	X	0110	0	1	X	0	DEC $R[DR] \leftarrow R[SA] + (-1)^*$
EX0	0001000	XXXX	INF	0	01	0XXX	0XXX	0XXX	0	1000	0	1	X	0	AND $R[DR] \leftarrow R[SA] \wedge R[SB]^*$
EX0	0001001	XXXX	INF	0	01	0XXX	0XXX	0XXX	0	1001	0	1	X	0	OR $R[DR] \leftarrow R[SA] \vee R[SB]^*$
EX0	0001010	XXXX	INF	0	01	0XXX	0XXX	0XXX	0	1010	0	1	X	0	XOR $R[DR] \leftarrow R[SA] \oplus R[SB]^*$
EX0	0001011	XXXX	INF	0	01	0XXX	0XXX	XXXX	X	1011	0	1	X	0	NOT $R[DR] \leftarrow \overline{R[SA]}^*$
EX0	0001100	XXXX	INF	0	01	0XXX	XXXX	0XXX	0	1100	0	1	X	0	MOVB $R[DR] \leftarrow R[SB]^*$
EX0	0010000	XXXX	INF	0	01	0XXX	0XXX	XXXX	X	XXXX	1	1	0	0	LD $R[DR] \leftarrow M[R[SA]]^*$
EX0	0100000	XXXX	INF	0	01	XXXX	0XXX	0XXX	0	XXXX	X	0	0	1	ST $M[R[SA]] \leftarrow R[SB]^*$
EX0	1001100	XXXX	INF	0	01	0XXX	XXXX	XXXX	1	1100	0	1	0	0	LDI $R[DR] \leftarrow zf\ OP^*$
EX0	1000010	XXXX	INF	0	01	0XXX	0XXX	XXXX	1	0010	0	1	0	0	ADI $R[DR] \leftarrow R[SA] + zf\ OP^*$
EX0	1100000	XXX1	INF	0	10	XXXX	0XXX	XXXX	X	0000	X	0	0	0	BRZ $PC \leftarrow PC + se\ AD$
EX0	1100000	XXX0	INF	0	01	XXXX	0XXX	XXXX	X	0000	X	0	0	0	BRZ $PC \leftarrow PC + 1$
EX0	1100001	XX1X	INF	0	10	XXXX	0XXX	XXXX	X	0000	X	0	0	0	BRN $PC \leftarrow PC + se\ AD$
EX0	1100001	XX0X	INF	0	01	XXXX	0XXX	XXXX	X	0000	X	0	0	0	BRN $PC \leftarrow PC + 1$
EX0	1110000	XXXX	INF	0	11	XXXX	0XXX	XXXX	X	0000	X	0	0	0	JMP $PC \leftarrow R[SA]$

* For this state and input combination, $PC \leftarrow PC + 1$ also occurs.



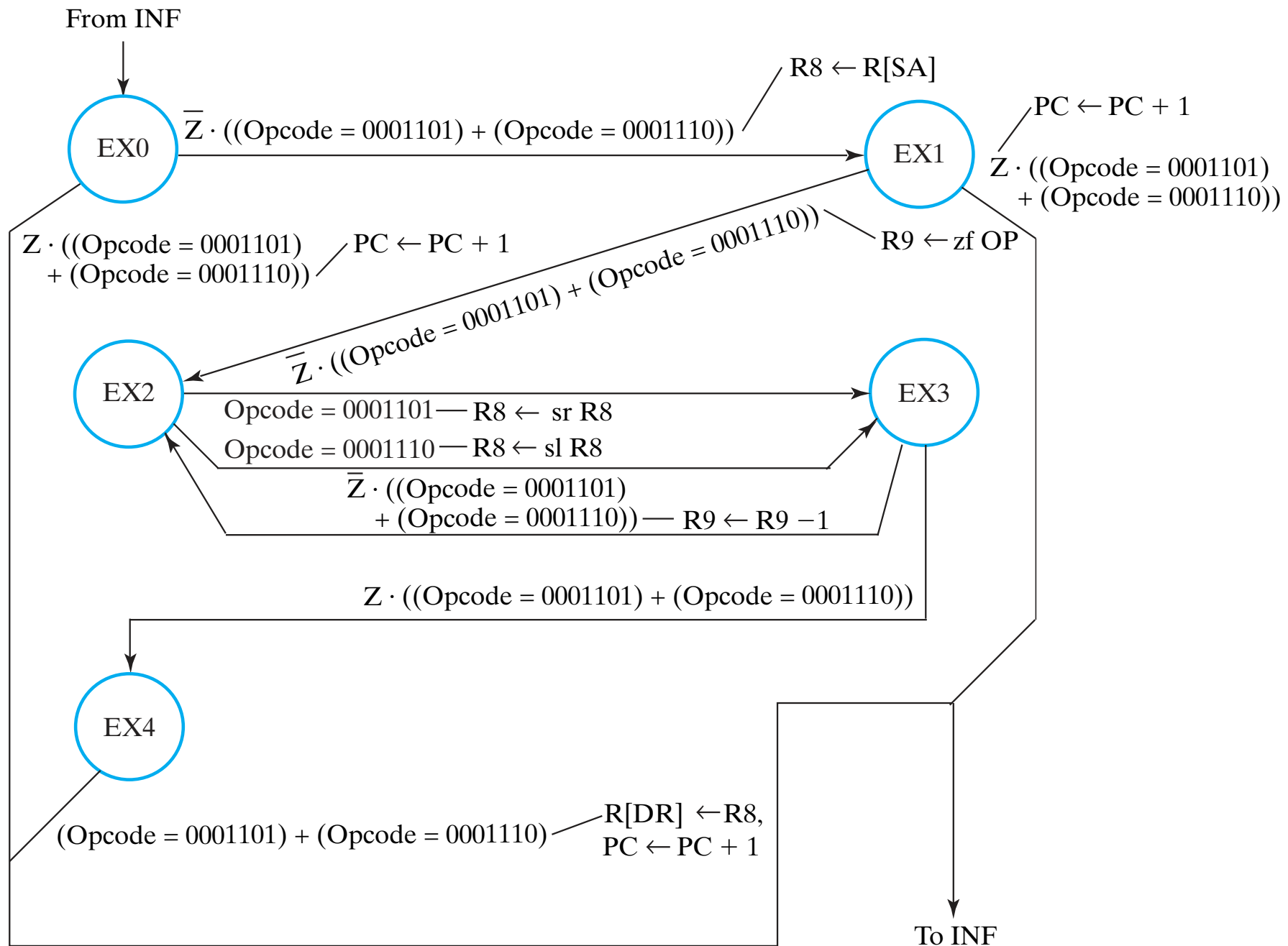


TABLE 9-15
State Table for Illustration of Instructions Having Three or More Cycles

State	Inputs		Next state	Outputs											Comments	
	Opcode	VCNZ		I L	PS	DX	AX	BX	MB	FS	MD	RW	MM	M W		
EX0	0010001	XXXX	EX1	0	00	1000	0XXX	XXXX	X	0000	1	1	X	0	LRI	$R8 \leftarrow M[R[SA]], \rightarrow EX1$
EX1	0010001	XXXX	INF	0	01	0XXX	1000	XXXX	X	0000	1	1	X	0	LRI	$R[DR] \leftarrow M[R8], \rightarrow INF^*$
EX0	0001101	XXX0	EX1	0	00	1000	0XXX	XXXX	X	0000	0	1	X	0	SRM	$R8 \leftarrow R[SA], \bar{Z}: \rightarrow EX1$
EX0	0001101	XXX1	INF	0	01	1000	0XXX	XXXX	X	0000	0	1	X	0	SRM	$R8 \leftarrow R[SA], Z: \rightarrow INF^*$
EX1	0001101	XXX0	EX2	0	00	1001	XXXX	XXXX	1	1100	0	1	X	0	SRM	$R9 \leftarrow zf\ OP, \bar{Z}: \rightarrow EX2$
EX1	0001101	XXX1	INF	0	01	1001	XXXX	XXXX	1	1100	0	1	X	0	SRM	$R9 \leftarrow zf\ OP, Z: \rightarrow INF^*$
EX2	0001101	XXXX	EX3	0	00	1000	XXXX	1000	0	1101	0	1	X	0	SRM	$R8 \leftarrow sr\ R8, \rightarrow EX3$
EX3	0001101	XXX0	EX2	0	00	1001	1001	XXXX	X	0110	0	1	X	0	SRM	$R9 \leftarrow R9 - 1, \bar{Z}: \rightarrow EX2$
EX3	0001101	XXX1	EX4	0	00	1001	1001	XXXX	X	0110	0	1	X	0	SRM	$R9 \leftarrow R9 - 1, Z: \rightarrow EX4$
EX4	0001101	XXXX	INF	0	01	0XXX	1000	XXXX	X	0000	0	1	X	0	SRM	$R[DR] \leftarrow R8, \rightarrow INF^*$
EX0	0001110	XXX0	EX1	0	00	1000	0XXX	XXXX	X	0000	0	1	X	0	SLM	$R8 \leftarrow R[SA], \bar{Z}: \rightarrow EX1$
EX0	0001110	XXX1	INF	0	01	1000	0XXX	XXXX	X	0000	0	1	X	0	SLM	$R8 \leftarrow R[SA], Z: \rightarrow INF^*$
EX1	0001110	XXX0	EX2	0	00	1001	XXXX	XXXX	1	1100	0	1	X	0	SLM	$R9 \leftarrow zf\ OP, \bar{Z}: \rightarrow EX2$
EX1	0001110	XXX1	INF	0	01	1001	XXXX	XXXX	1	1100	0	1	X	0	SLM	$R9 \leftarrow zf\ OP, Z: \rightarrow INF^*$
EX2	0001110	XXXX	EX3	0	00	1000	XXXX	1000	0	1110	0	1	X	0	SLM	$R8 \leftarrow sl\ R8, \rightarrow EX3$
EX3	0001110	XXX0	EX2	0	00	1001	1001	XXXX	X	0110	0	1	X	0	SLM	$R9 \leftarrow R9 - 1, \bar{Z}: \rightarrow EX2$
EX3	0001110	XXX1	EX4	0	00	1001	1001	XXXX	X	0110	0	1	X	0	SLM	$R9 \leftarrow R9 - 1, Z: \rightarrow EX4$
EX4	0001110	XXXX	INF	0	01	0XXX	1000	XXXX	X	0000	0	1	X	0	SLM	$R[DR] \leftarrow R8, \rightarrow IF^*$

*For this state and input combination, $PC \leftarrow PC + 1$ also occurs.