Control Signals for Binary Multiplier

Block Diagram Module	Microoperation	Control Signal Name	Control Expression
Register A:	$A \leftarrow 0$	Initialize	$IDLE \cdot G$
	$A \leftarrow A + B$	Load	$MUL0 \cdot Q_0$
	$C \| A \| Q \leftarrow \text{sr } C \| A \| Q$	Shift_dec	$MUL1$
Register B:	$B \leftarrow IN$	Load_B	$LOADB$
Flip-Flop C:	$C \leftarrow 0$	Clear_C	$IDLE \cdot G + MUL1$
	$C \leftarrow C_{out}$	Load	—
Register Q:	$Q \leftarrow IN$	Load_Q	$LOADQ$
	$C \| A \| Q \leftarrow \text{sr } C \| A \| Q$	Shift_dec	—
Counter P:	$P \leftarrow n - 1$	Initialize	—
	$P \leftarrow P - 1$	Shift_dec	—

State Table for Sequence Register and Decoder Part of Multiplier Control Unit

Present state			Inputs		Next state		Decoder Outputs		
Name	M_1	M_0	G	Z	M_1	M_0	IDLE	MUL0	MUL1
IDLE	0	0	0	×	0	0	1	0	0
	0	0	1	×	0	1	1	0	0
MUL0	0	1	×	×	1	0	0	1	0
MUL1	1	0	×	0	0	1	0	0	1
	1	0	×	1	0	0	0	0	1
—	1	1	×	×	×	×	×	×	×

Control Signals for Microprogrammed Multiplier Control

Control Signal	Register Transfers	States in Which Signal is Active	Micro-instruction Bit Position	Symbolic Notation
Initialize	$A \leftarrow 0, P \leftarrow n-1$	INIT	0	IT
Load	$A \leftarrow A + B, C \leftarrow C_{\text{out}}$	ADD	1	LD
Clear_C	$C \leftarrow 0$	INIT, MUL1	2	CC
Shift_dec	$C \| A \| Q \leftarrow \text{sr } C \| A \| Q, P \leftarrow P-1$	MUL1	3	SD

SEL Field Definition for Binary Multiplier Control Sequencing

SEL		
Symbolic notation	**Binary Code**	**Sequencing Microoperations**
NXT	00	$CAR \leftarrow NXTADD0$
DG	01	$\overline{G}: CAR \leftarrow NXTADD0$ $G: CAR \leftarrow NXTADD1$
DQ	10	$\overline{Q_0}: CAR \leftarrow NXTADD0$ $Q_0: CAR \leftarrow NXTADD1$
DZ	11	$\overline{Z}: \ CAR \leftarrow NXTADD0$ $Z: CAR \leftarrow NXTADD1$

Register Transfer Description of Binary Multiplier Microprogram

Address	Symbolic transfer statement
IDLE	$G\colon CAR \leftarrow \text{INIT}, \overline{G}\colon CAR \leftarrow \text{IDLE}$
INIT	$C \leftarrow 0, A \leftarrow 0, P \leftarrow n-1, CAR \leftarrow \text{MUL0}$
MUL0	$Q_0\colon CAR \leftarrow \text{ADD}, \overline{Q_0}\colon CAR \leftarrow \text{MUL1}$
ADD	$A \leftarrow A + B, C \leftarrow C_{\text{out}}, CAR \leftarrow \text{MUL1}$
MUL1	$C \leftarrow 0, C \Vert A \Vert Q \leftarrow \text{sr}\ C \Vert A \Vert Q,\ Z\colon CAR \leftarrow \text{IDLE}, \overline{Z}\colon CAR \leftarrow \text{MUL0},$ $P \leftarrow P-1$

Symbolic Microprogram and Binary Microprogram for Multiplier

Address	NXTADD1	NXTADD0	SEL	DATAPATH	Address	NXTADD1	NXTADD0	SEL	DATAPATH
IDLE	INIT	IDLE	DS	None	000	001	000	01	0000
INIT	—	MUL0	NXT	IT, CC	001	000	010	00	0101
MUL0	ADD	MUL1	DQ	None	010	011	100	10	0000
ADD	—	MUL1	NXT	LD	011	000	100	00	0010
MUL1	IDLE	MUL0	DZ	CC, SD	100	010	000	11	1100

Truth Table for Instruction Decoder Logic

Instruction Bits			Control Word Bits				
Bit 15	Bit 14	Bit 13	MB	MD	RW	MW	Typical Operation Category
0	0	0	0	0	1	0	ALU function using registers
0	0	1	0	0	1	0	Shifter function using registers
0	1	0	0	1	0	1	Memory write using register data
0	1	1	0	1	1	0	Memory read using register data
1	0	0	1	0	1	0	ALU operation using a constant
1	0	1	1	0	1	0	Shifter function using registers
1	1	0	1	1	0	1	Memory write using constant data
1	1	1	1	1	1	0	Memory read using constant data

Six Instructions for the Single-Cycle Computer

Operation code	Symbolic name	Format	Description	Function	MB	MD	RW	MW
1000010	ADI	Immediate	Add immediate operand	$R[DR] \leftarrow R[SA] + zf\ I(2:0)$	1	0	1	0
0110000	LD	Register	Load memory content into register	$R[DR] \leftarrow M[R[SA]]$	0	1	1	0
0100000	ST	Register	Store register content in memory	$M[R[SA]] \leftarrow R[SB]$	0	1	0	1
0000001	INC	Register	Increment register	$R[DR] \leftarrow R[SA] + 1$	0	0	1	0
0001110	NOT	Register	Complement register	$R[DR] \leftarrow \overline{R[SA]}$	0	0	1	0
0000010	ADD	Register	Add registers	$R[DR] \leftarrow R[SA] + R[SB]$	0	0	1	0

Control Word Information for Datapath

TD	TA	TB	MB		FS		MD	RW	MM	MW	
Select	Select	Select	Select	Code	Function	Code	Select	Function	Select	Function	Code
R[DR]	R[SA]	R[SB]	Register	0	$F = A$	00000	FnUt	No write(NW)	Address	No write(NW)	0
R8	R8	R8	Constant	1	$F = A + 1$	00001	Data In	Write(WR)	PC	Write(WR)	1
					$F = A + B$	00010					
					$F = A + B + 1$	00011					
					$F = A + \overline{B}$	00100					
					$F = A + \overline{B} + 1$	00101					
					$F = A - 1$	00110					
					$F = A$	00111					
					$F = A \wedge B$	01000					
					$F = A \vee B$	01010					
					$F = A \oplus B$	01100					
					$F = \overline{A}$	01110					
					$F = A$	10000					
					$F =$ sl A	10010					
					$F =$ sr A	10100					
					$F = 0$	10110					

Control Information for Sequence Control Fields

MS			MC		IL		PI		
Action	Symbolic Notation	Code	Select	Symbolic Notation	Action	Symbolic Notation	Action	Symbolic Notation	Code
Increment CAR	CNT	000	NA	NXA	No load	NLI	No load	NLP	0
Load CAR	NXT	001	Opcode	OPC	Load instr.	LDI	Increment PC	INP	1
If $C = 1$, load CAR; else increment CAR	BC	010							
If $V = 1$, load CAR; else increment CAR	BV	011							
If $Z = 1$, load CAR; else increment CAR	BZ	100							
If $N = 1$, load CAR; else increment CAR	BN	101							
If $C = 0$, load CAR; else increment CAR	BNC	110							
If $Z = 0$, load CAR, else increment CAR	BNZ	111							

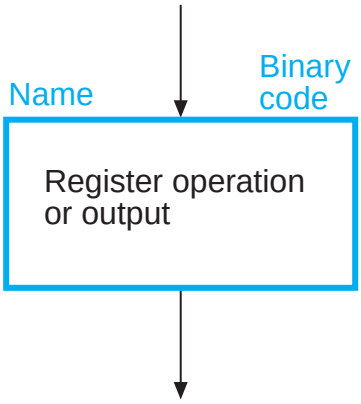
Symbolic Microprogram for Fetch and Execution of Six Instructions

Address	NXT ADD	MS	MC	IL	PI	TD	TA	TB	MB	FS	MD	RW	MM	MW
IF	EX0	CNT	—	LDI	INP	—	—	—	—	—	—	NW	PC	NW
EXO	—	NXT	OPC	NLI	NLP	—	—	—	—	—	—	NW	—	NW
ADI	IF	NXT	NXA	NLI	NLP	DR	SA	—	Constant	$F = A + B$	FnUt	WR	—	NW
LD	IF	NXT	NXA	NLI	NLP	DR	SA	—	—	—	Data	WR	MA	NW
ST	IF	NXT	NXA	NLI	NLP	—	SA	SB	Register	—	—	NW	MA	WR
INC	IF	NXT	NXA	NLI	NLP	DR	SA	—	—	$F = A + 1$	FnUt	WR	—	NW
NOT	IF	NXT	NXA	NLI	NLP	DR	SA	—	—	$F = \bar{A}$	FnUt	WR	—	NW
ADD	IF	NXT	NXA	NLI	NLP	DR	SA	SB	Register	$F = A + B$	FnUt	WR	—	NW

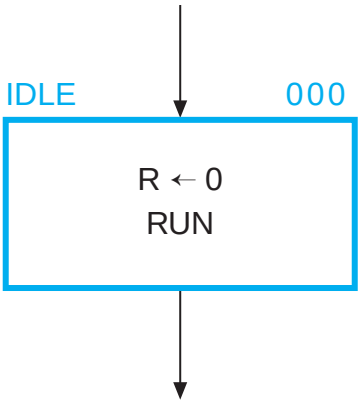
Binary Microprogram for Fetch and Execution of Six Instructions

Address	NXT ADD	MS	MC	IL	PI	TD	TA	TB	MB	FS	MD	RW	MM	MW
192	193	000	0	1	1	0	0	0	0	00000	0	0	1	0
193	000	001	1	0	0	0	0	0	0	00000	0	0	0	0
000	192	001	0	0	0	0	0	0	1	00010	0	1	0	0
001	192	001	0	0	0	0	0	0	0	00000	1	1	0	0
002	192	001	0	0	0	0	0	0	0	00000	0	0	0	1
003	192	001	0	0	0	0	0	0	0	00001	0	1	0	0
004	192	001	0	0	0	0	0	0	0	01110	0	1	0	0
005	192	001	0	0	0	0	0	0	0	00010	0	1	0	0

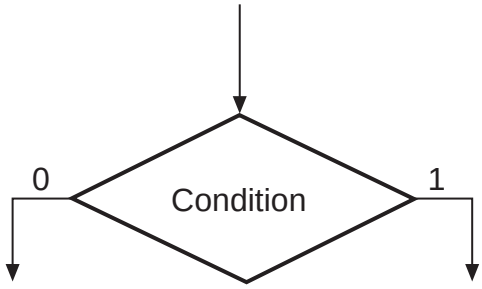
ASM Chart Elements



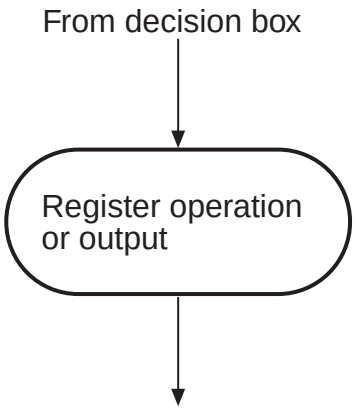
(a) State box



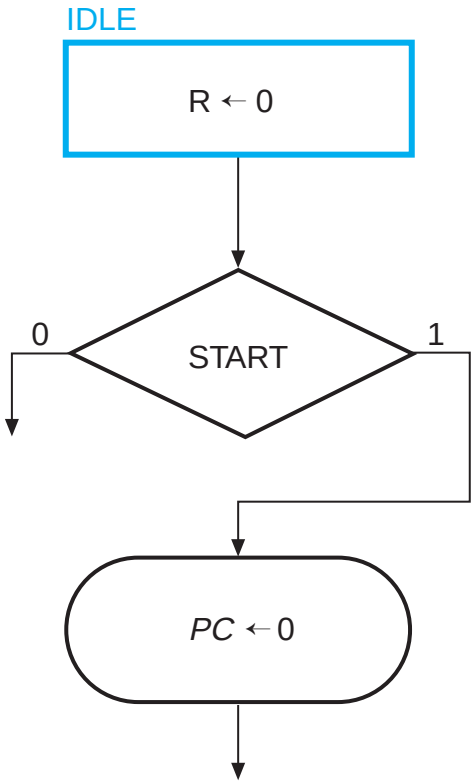
(b) Example of state box



(c) Decision box

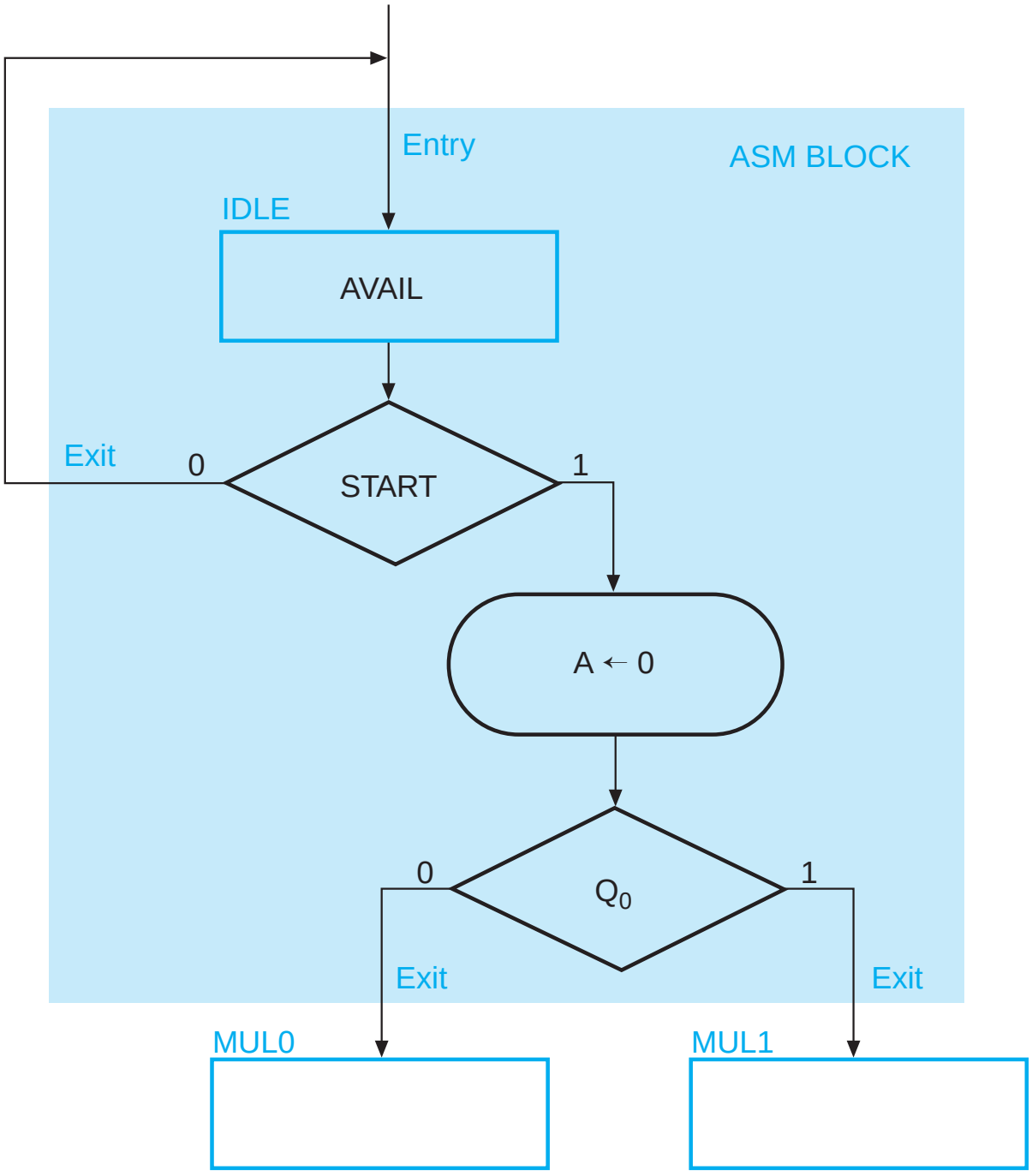


(d) Conditional output box

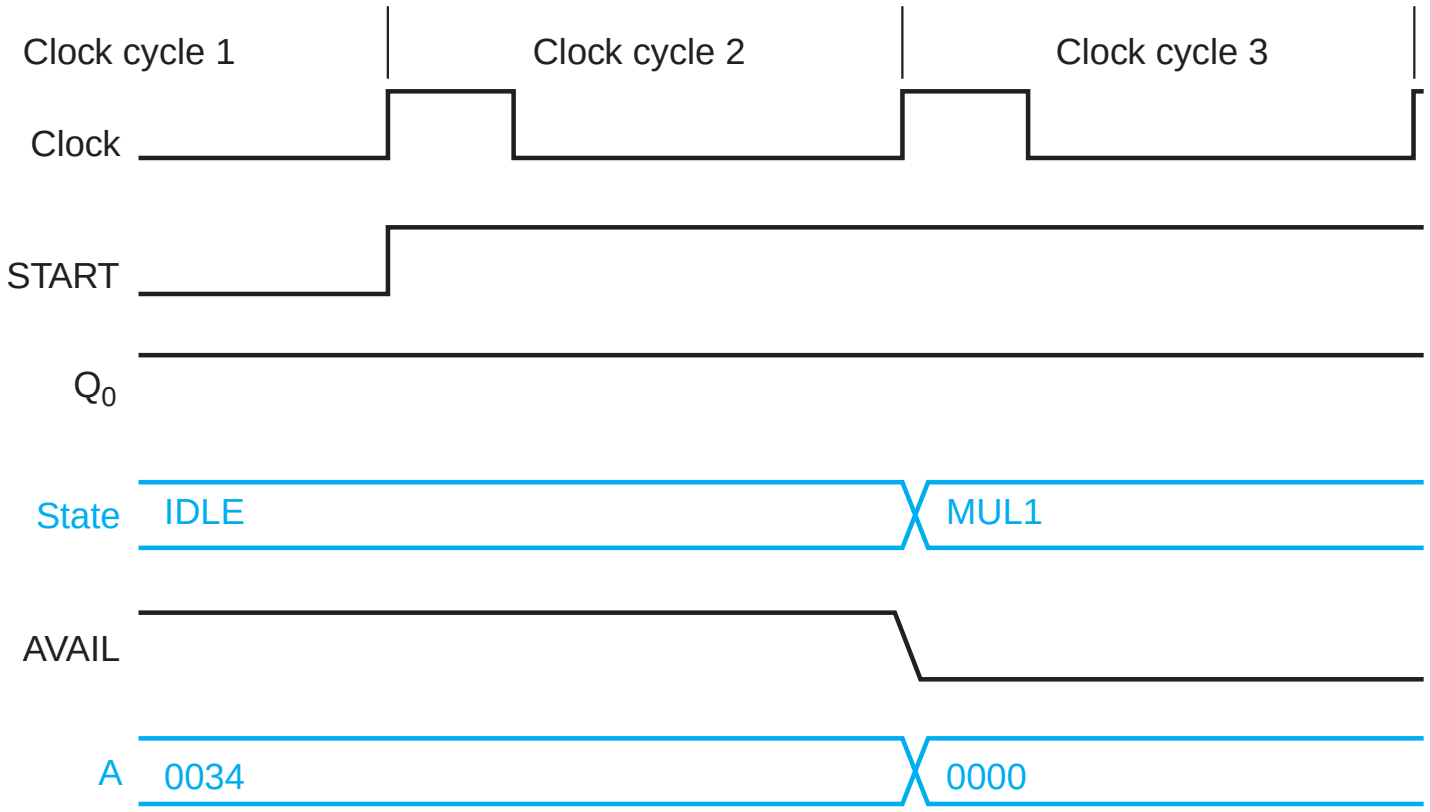


(e) Example of decision and condition output box

ASM Block



ASM Timing Behavior



Hand Multiplication Example

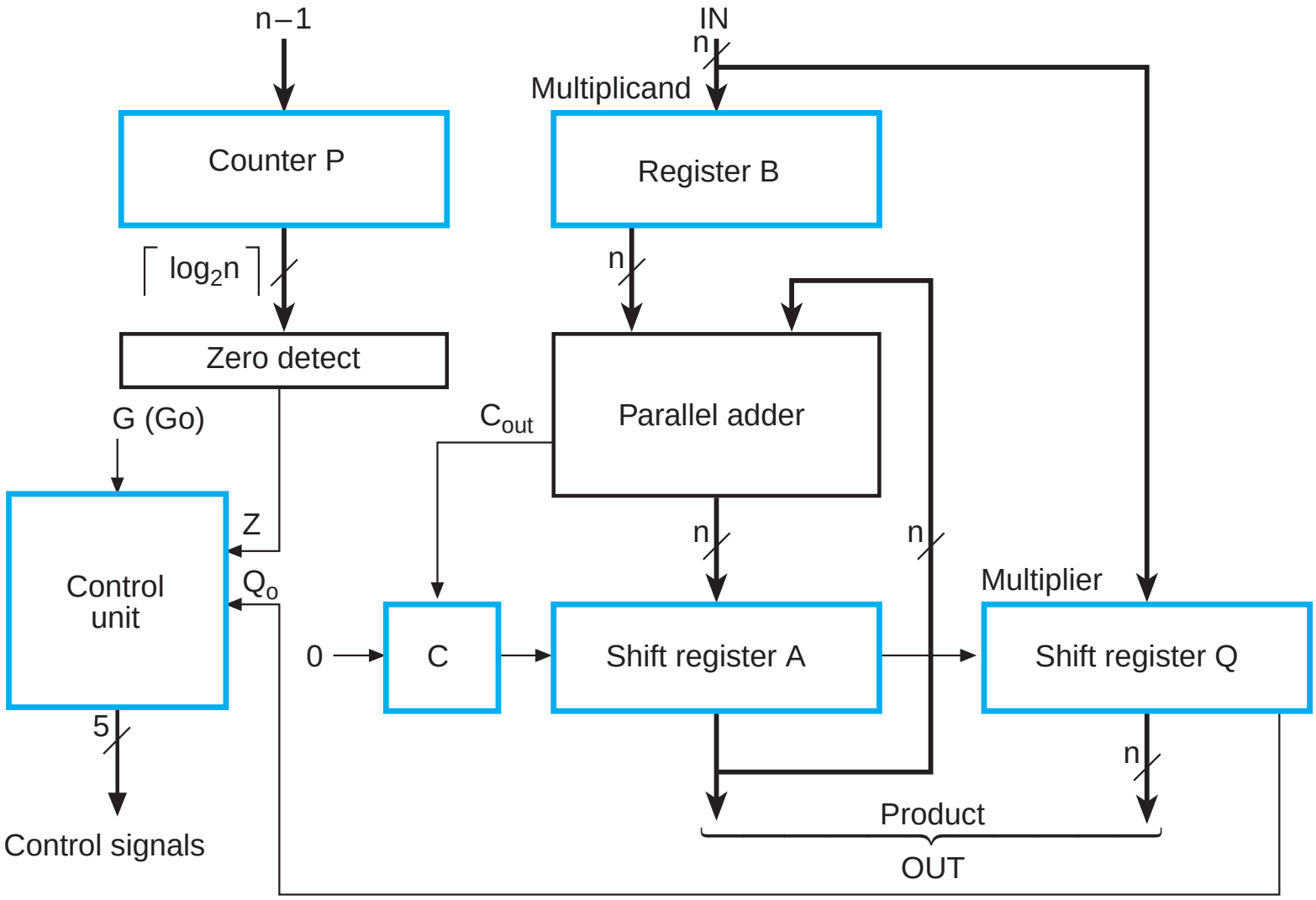
23	10111	Multiplicand
19	10011	Multiplier
	10111	
	10111	
	00000	
	00000	
	10111	
437	110110101	Product

Hardware Multiplication Example

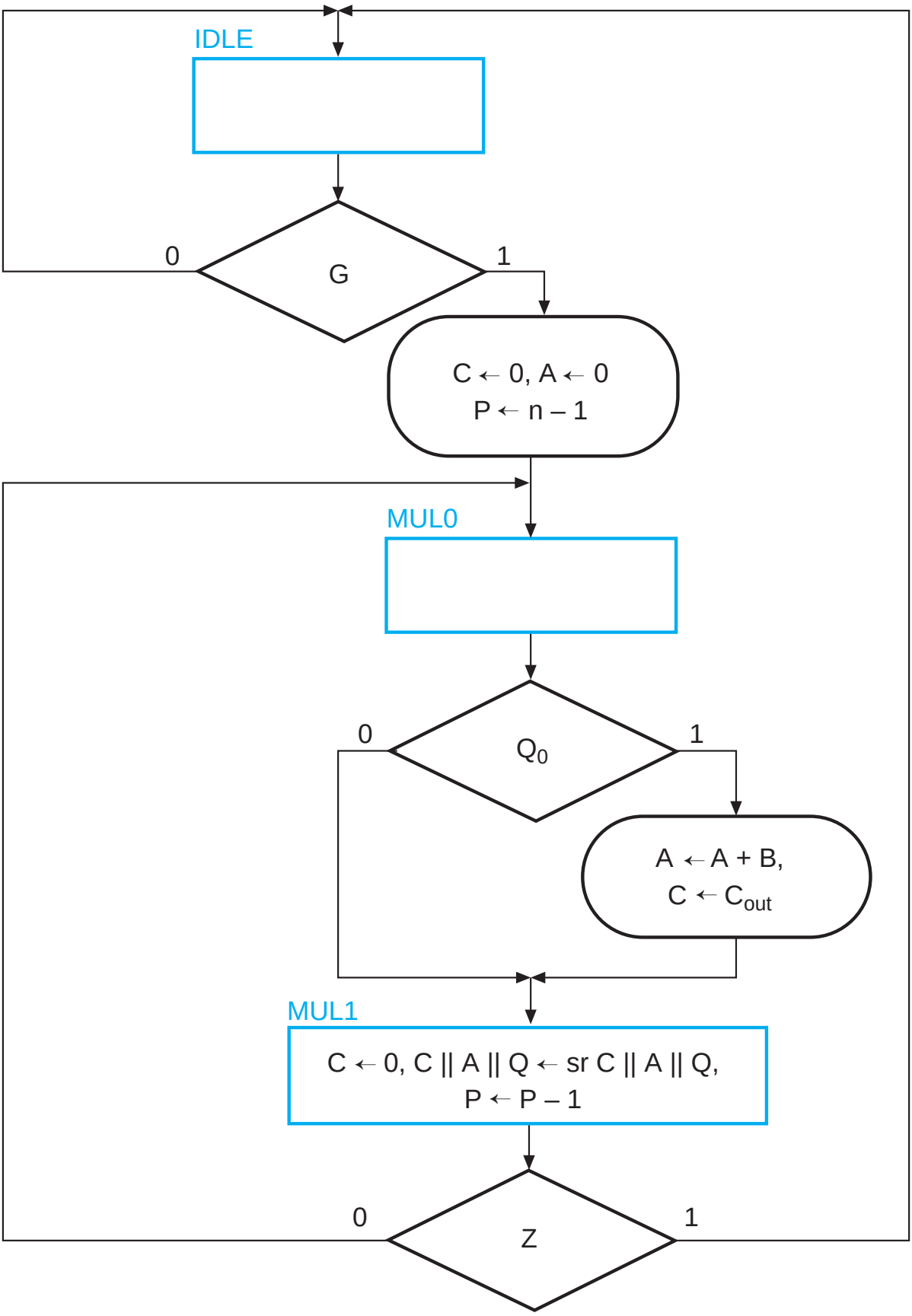
23	10111	Multiplicand
19	10011	Multiplier
	00000	Initial partial product
	10111	Add multiplicand, since multiplier bit is 1
	10111	Partial product after add and before shift
	010111	Partial product after shift
	10111	Add multiplicand, since multiplier bit is 1
	1000101	Partial product after add and before shift[a]
	1000101	Partial product after shift
	01000101	Partial product after shift
	001000101	Partial product after shift
	10111	Add multiplicand, since multiplier bit is 1
	110110101	Partial product after add and before shift
437	0110110101	Product after final shift

a. Note that overflow temporarily occurred.

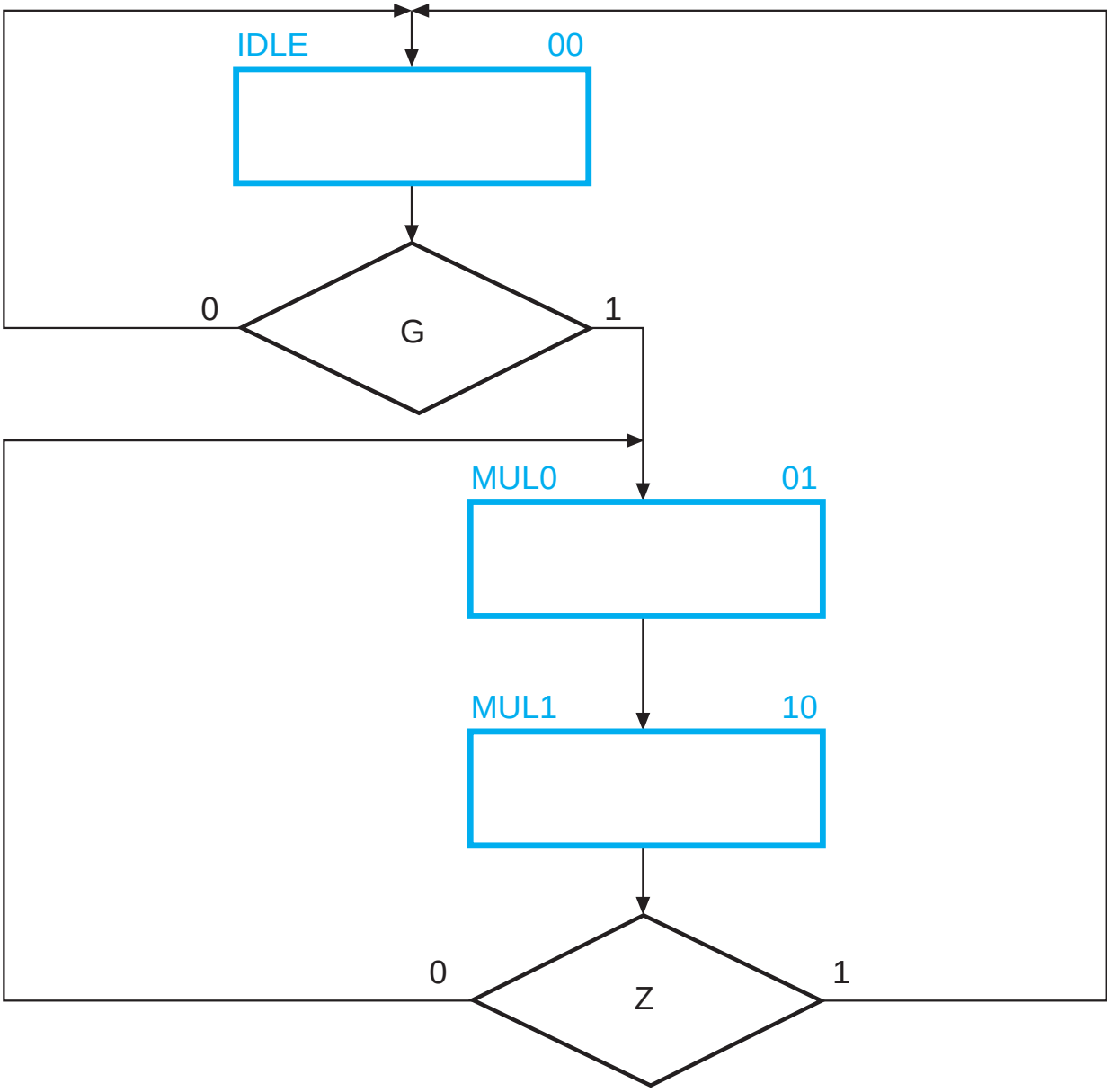
Block Diagram for Binary Multiplier



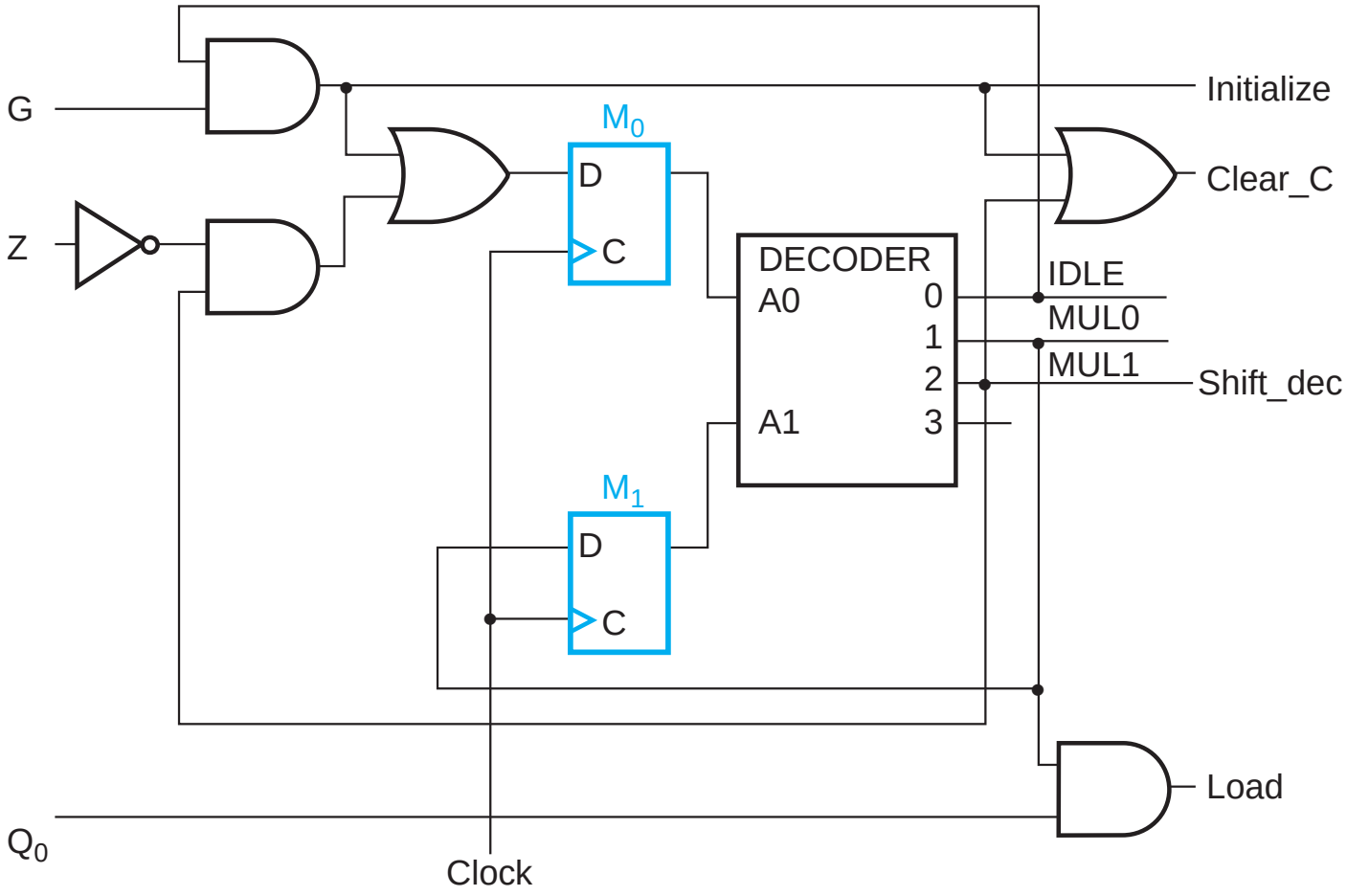
ASM Chart for Binary Multiplier



T-211 Sequencing Part of ASM Chart for the Binary Multiplier

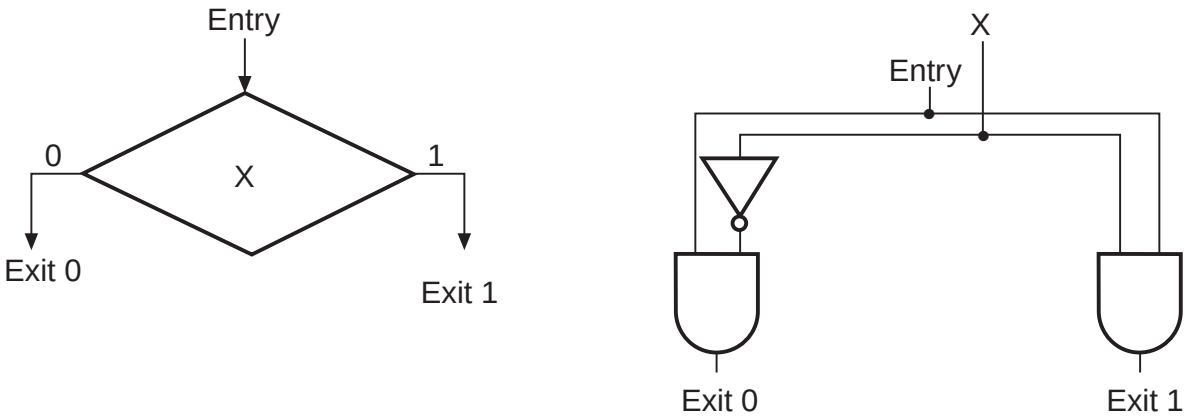


Control Unit for Binary Multiplier Using a Sequence Register and a Decoder



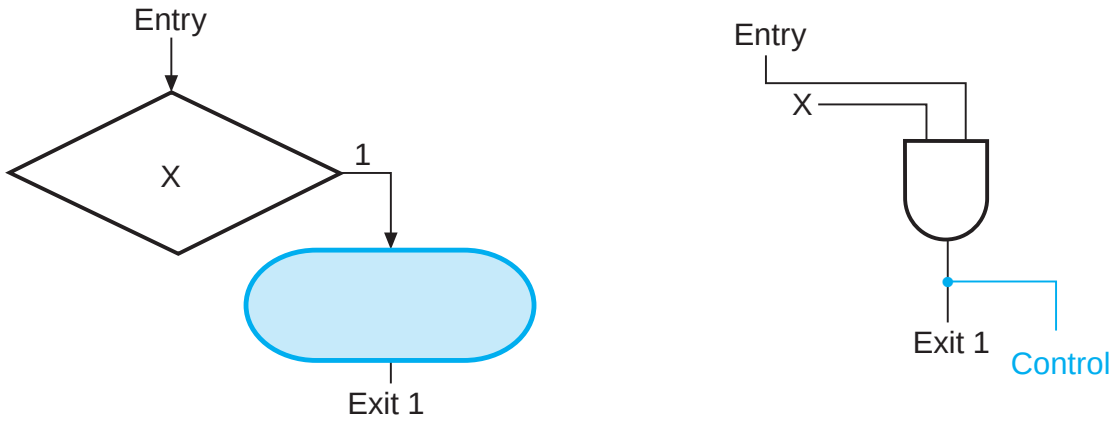
Transformation Rules for Control Unit with One Flip-Flop per State

(a) State box



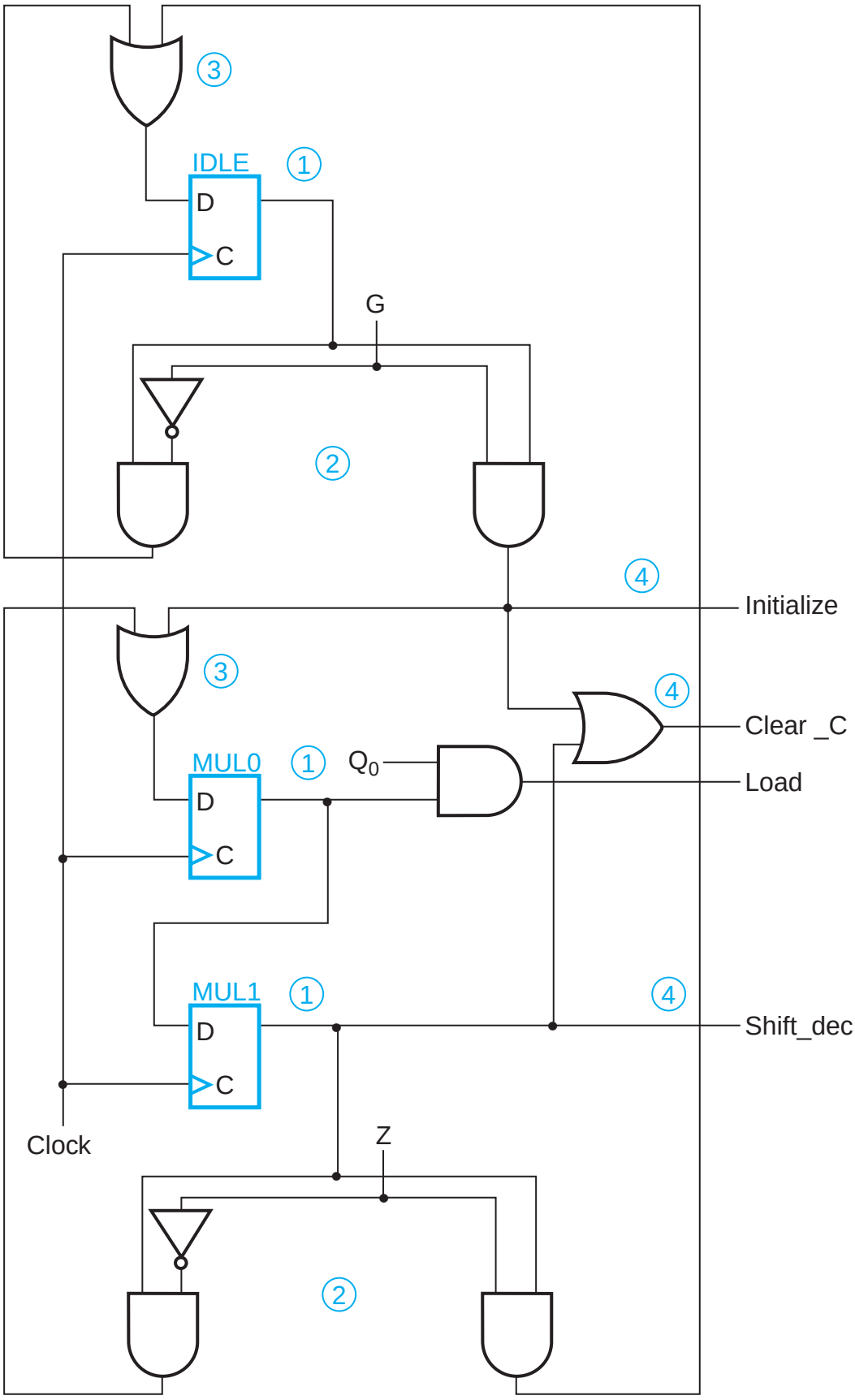
(b) Decision box

(c) Junction

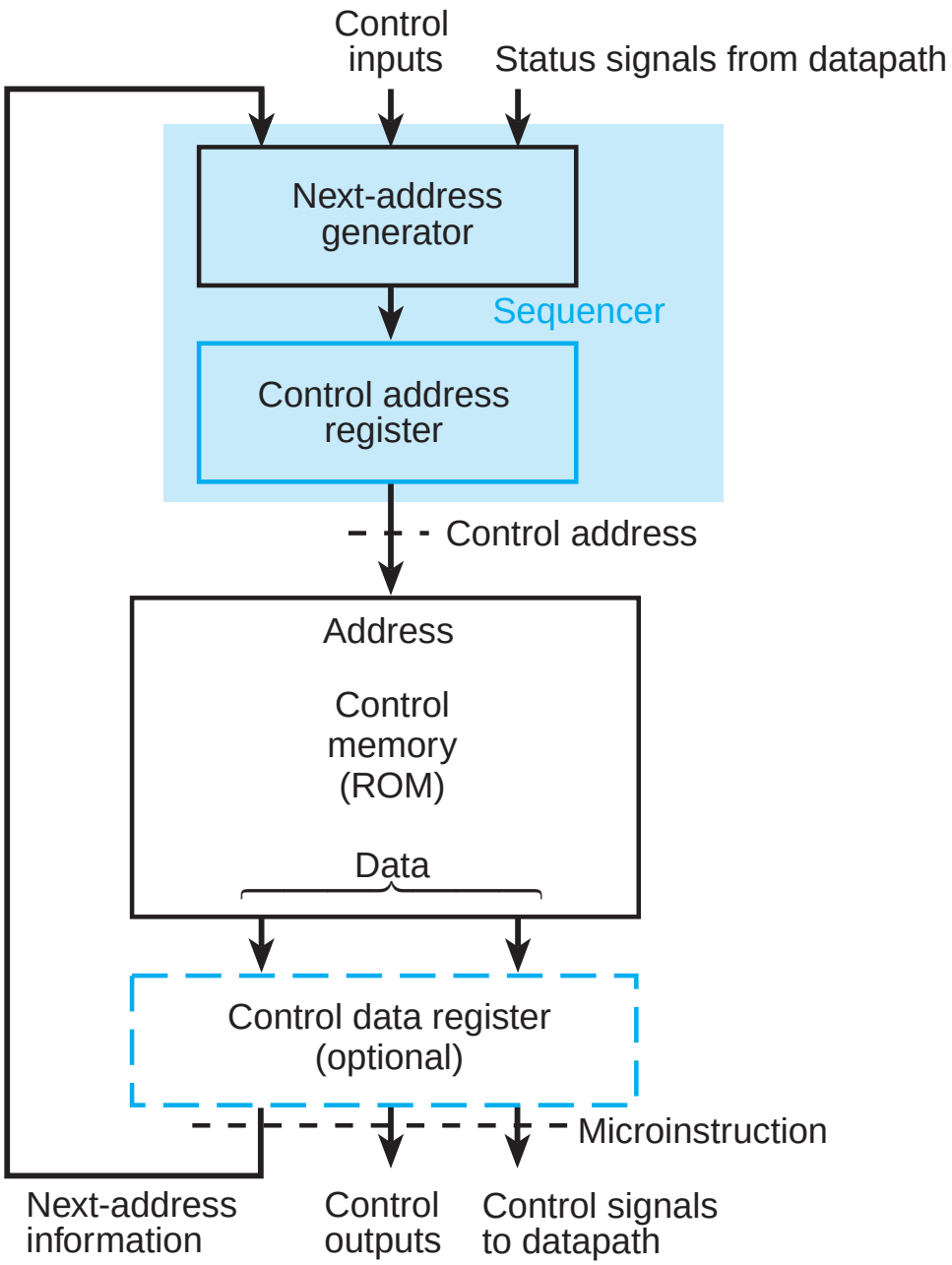


(d) Conditional output box

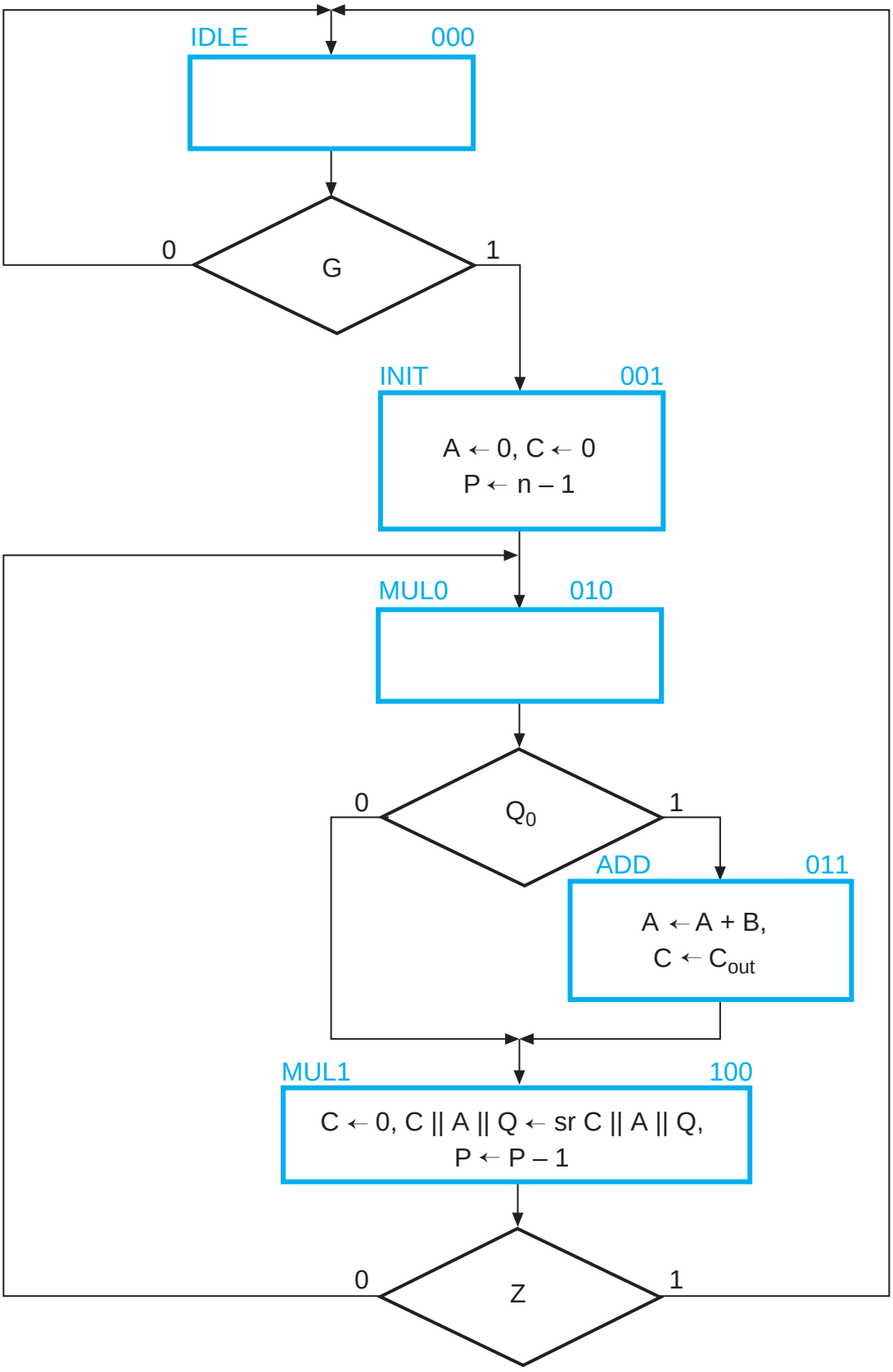
Control Unit with One Flip-Flop per State for the Binary Multiplier



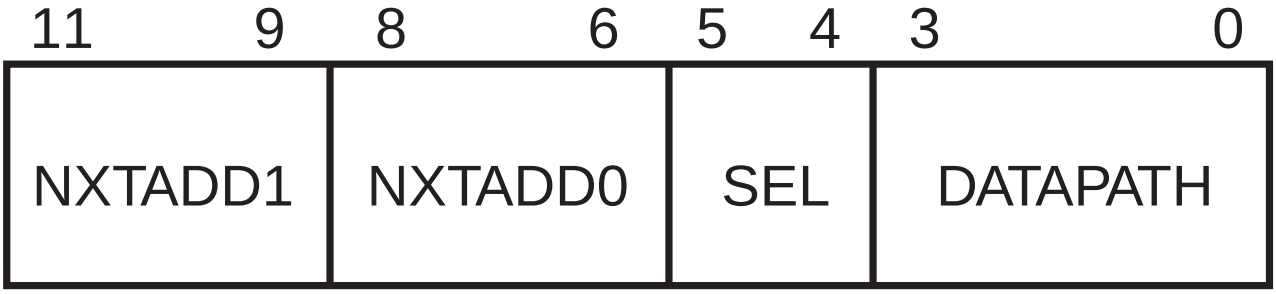
Microprogrammed Control Unit Organization



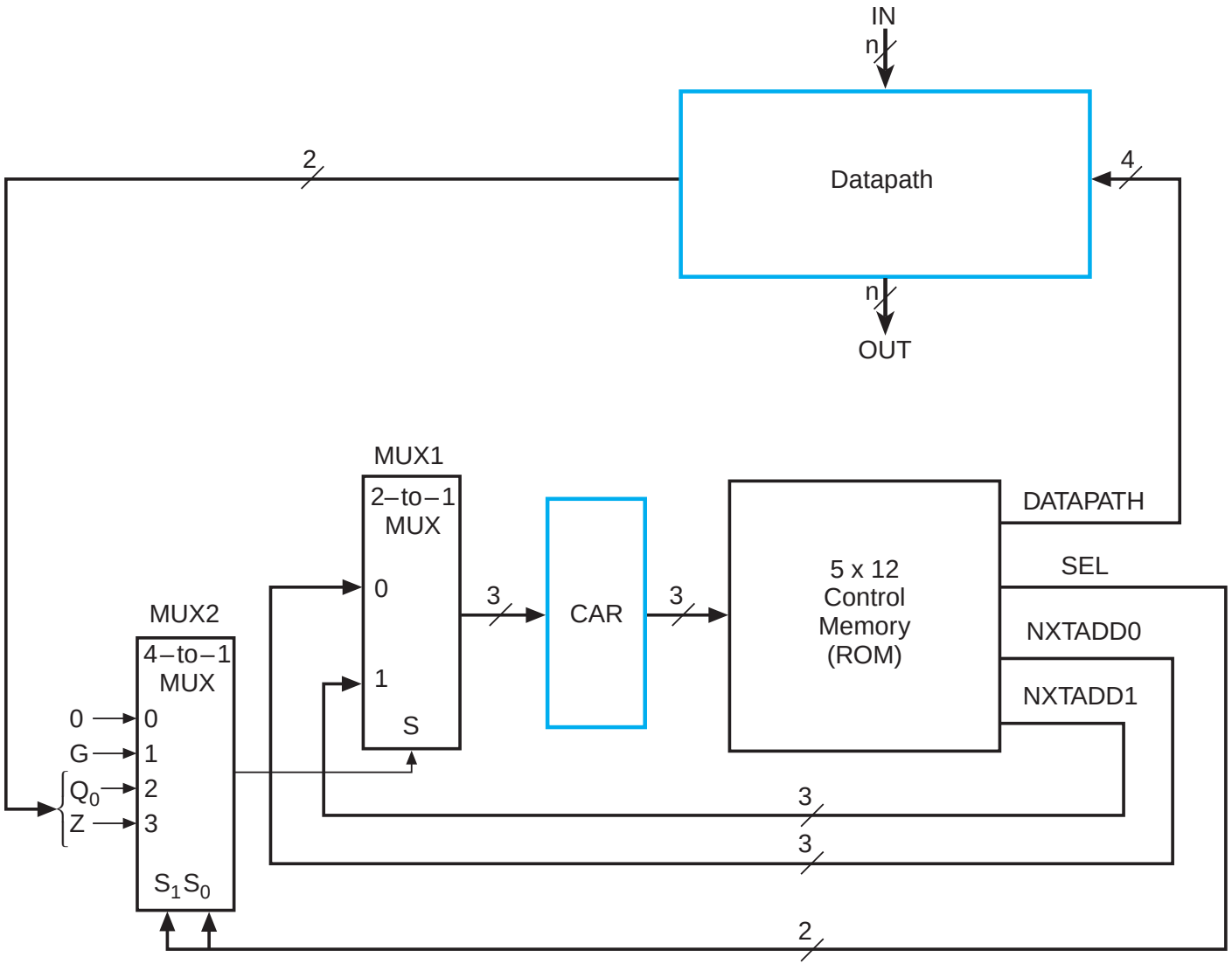
ASM Chart for Microprogrammed Binary Multiplier Control Unit



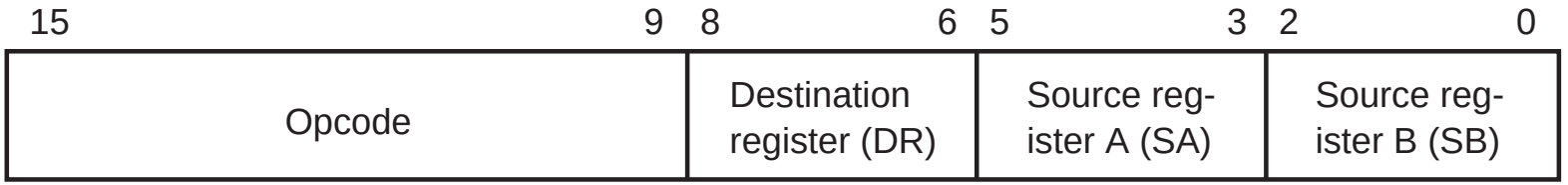
Microinstruction Control Unit Word Format



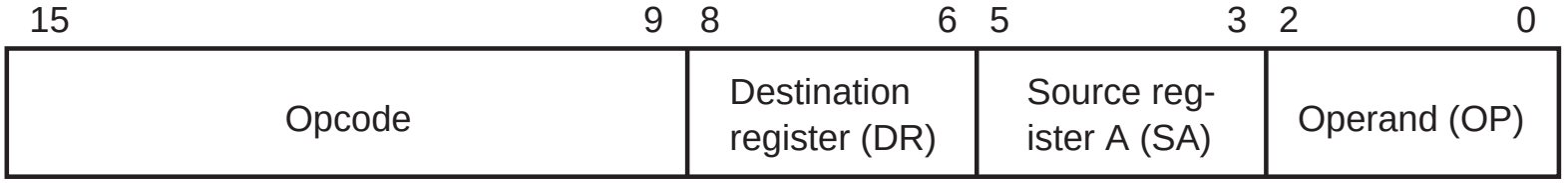
Microprogrammed Control Unit for Multiplier



Two Instruction Formats



(a) Register



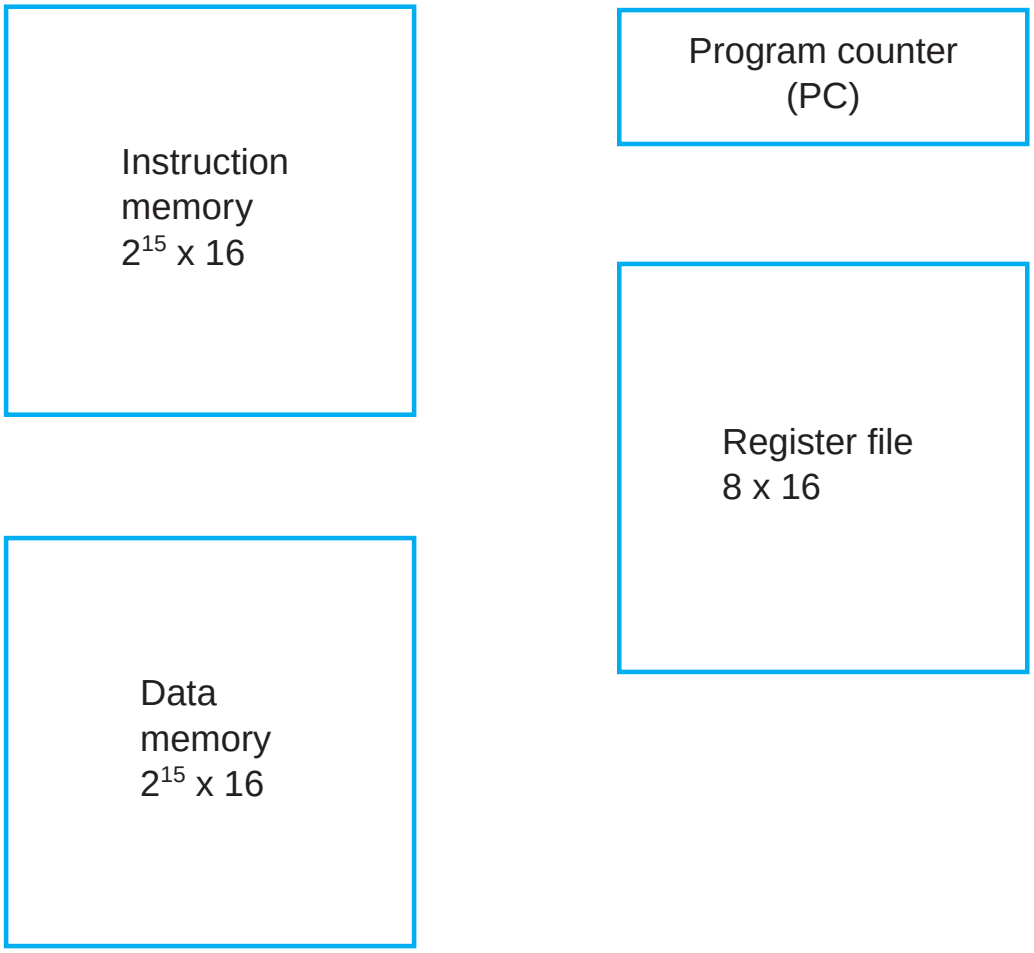
(b) Immediate

Memory Representation of Instructions and Data

Decimal address	Memory contents	Decimal opcode	Other specified fields	Operation
25	0000101 001 010 011	5 (Subtract)	DR:1, SA:2 SB:3	R1 ← R2 – R3
35	0100000 000 100 101	32 (Store)	SA:4 SB:5	M [R4] ← R5
45	1000010 010 111 011	66 (Add Immediate)	DR:2 SA:7 OP:3	R2 ← R7 + 3
70	0000000 011 000 000	Data = 192. After execution of instruction in 35, Data = 80.		

T-221

Storage Resource Diagram for a Simple Computer



Block Diagram of a Single-Cycle Computer

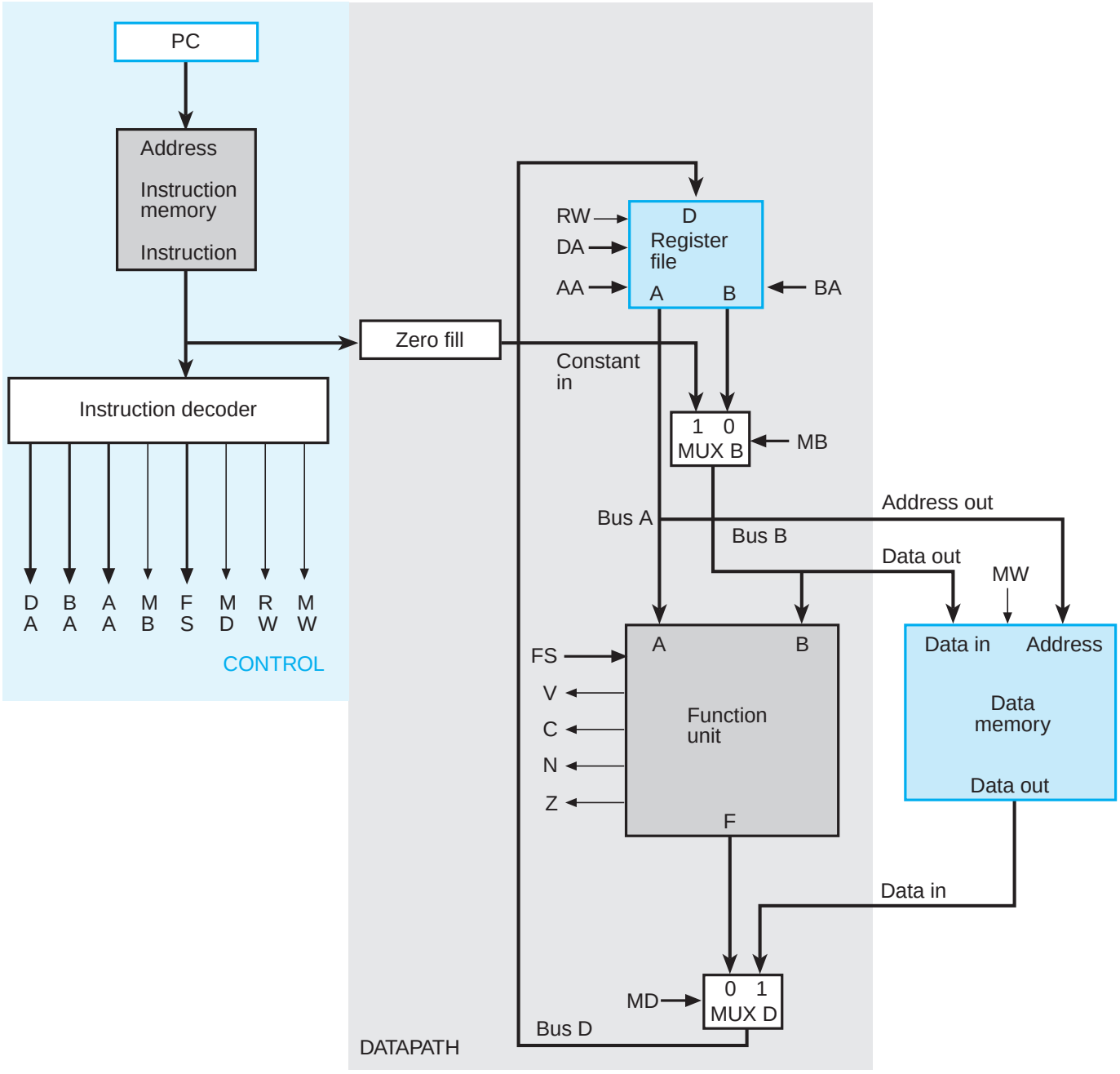
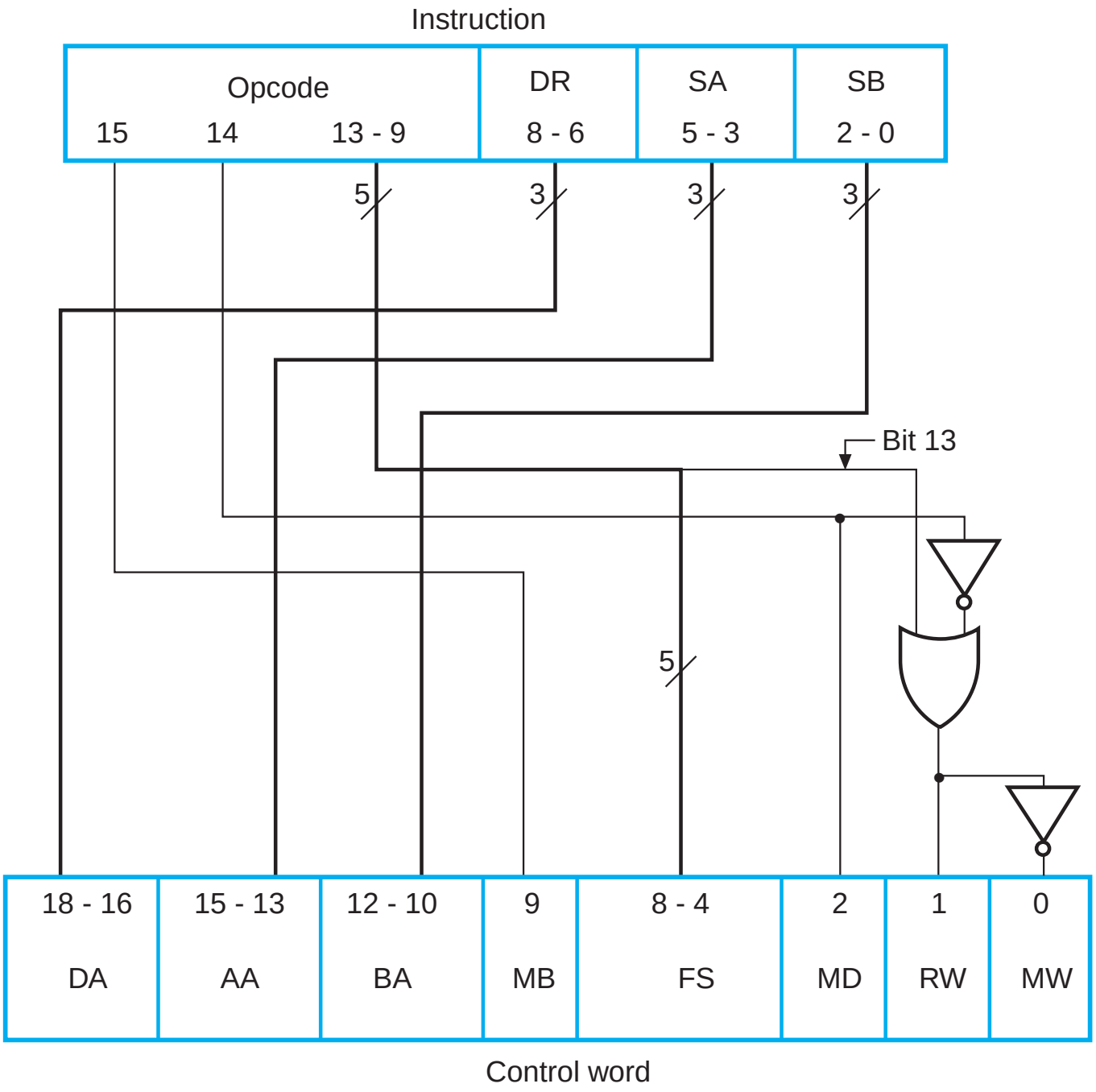
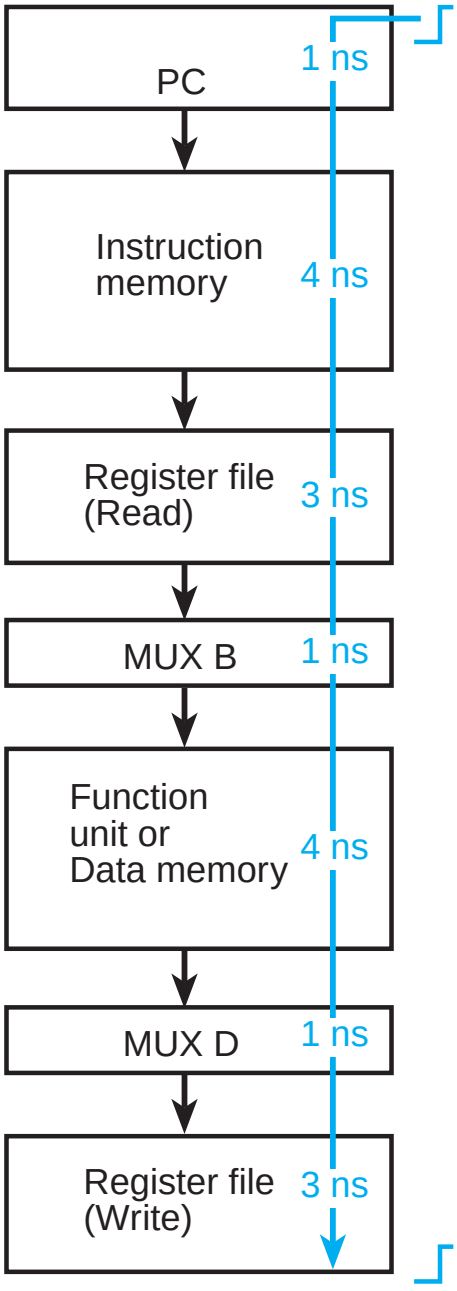


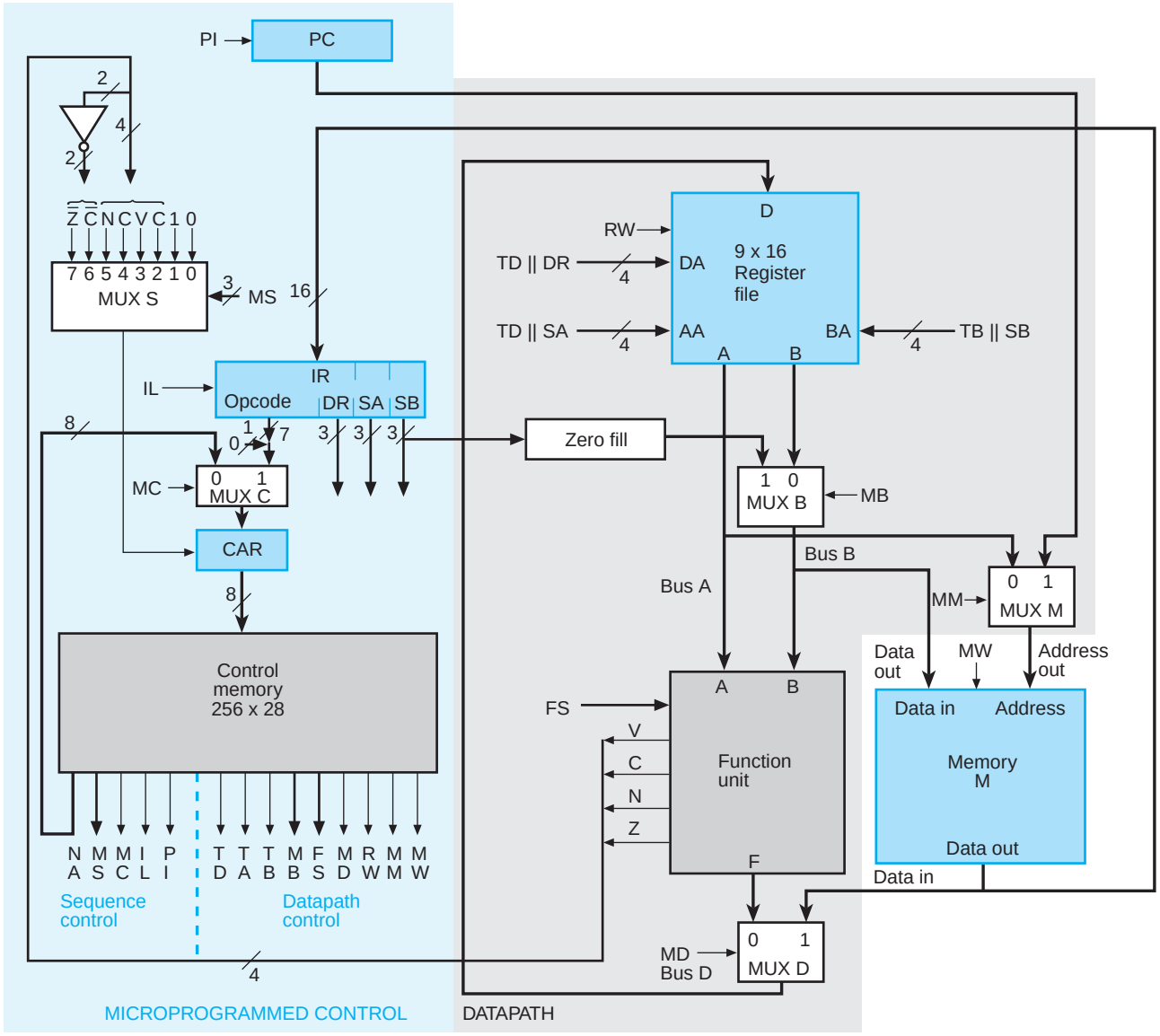
Diagram of Instruction Decoder



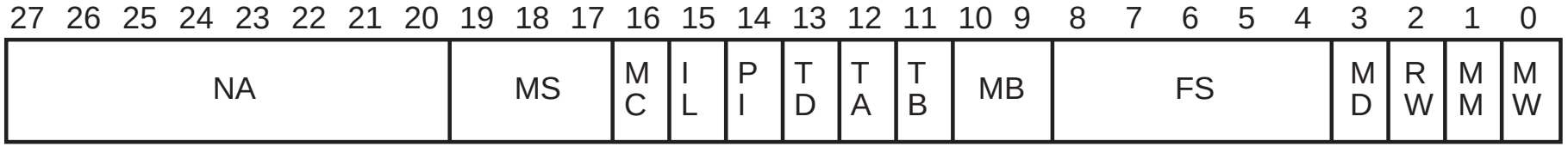
Worst Case Delay Path in Single-Cycle Computer



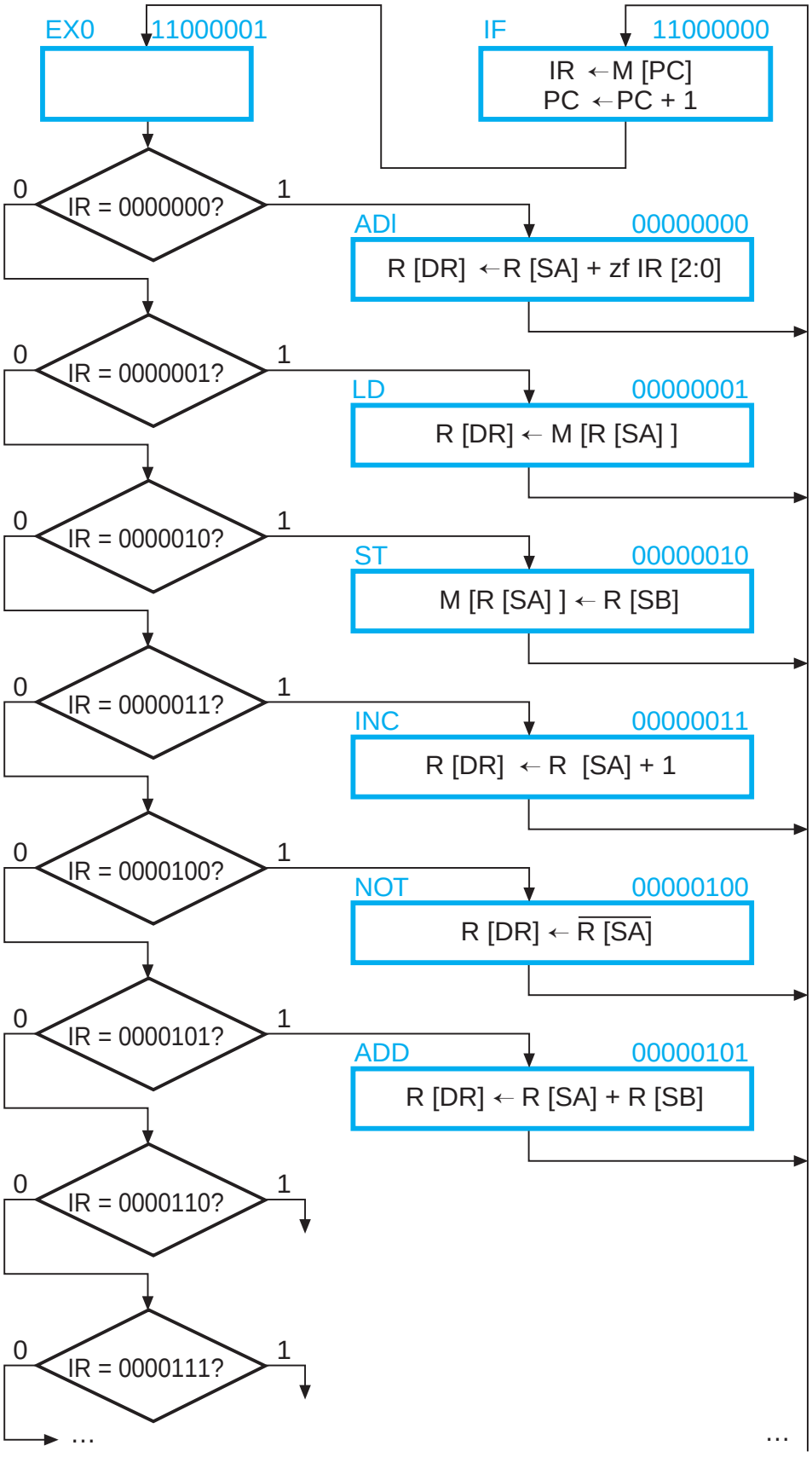
Multiple-Cycle Microprogrammed Computer



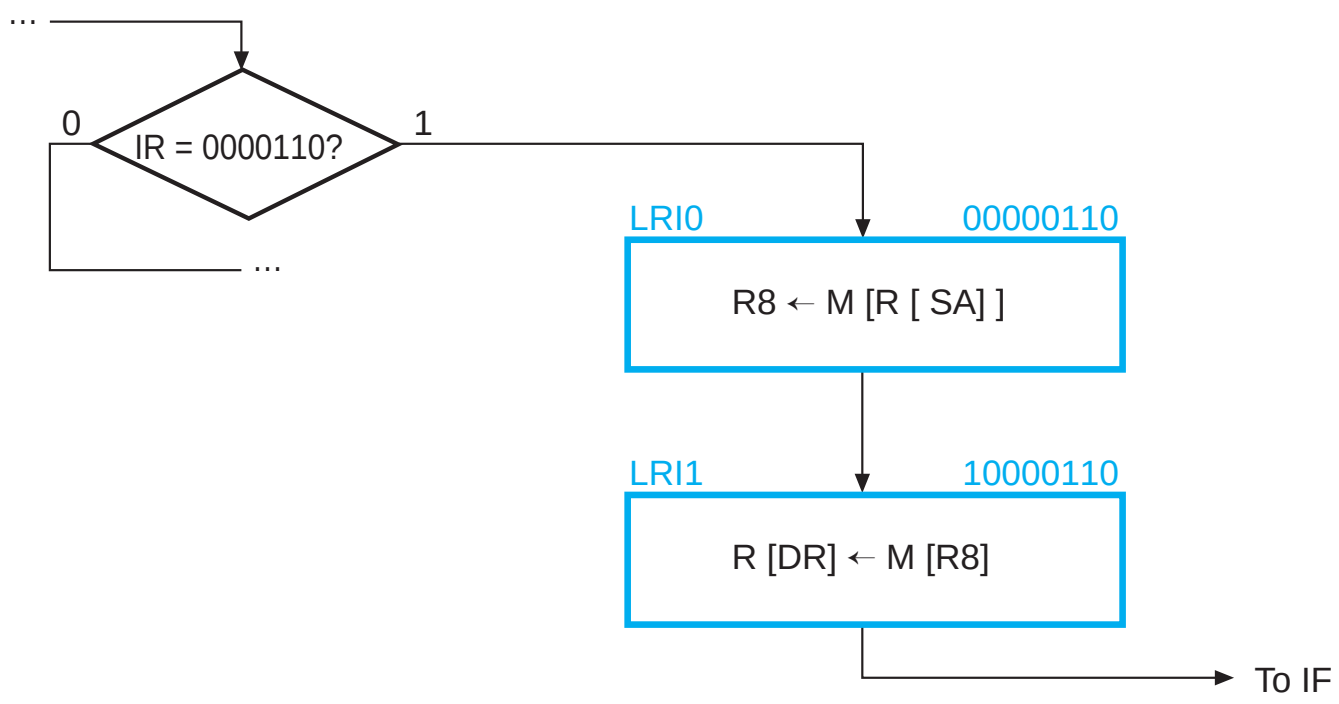
Format for Microinstruction



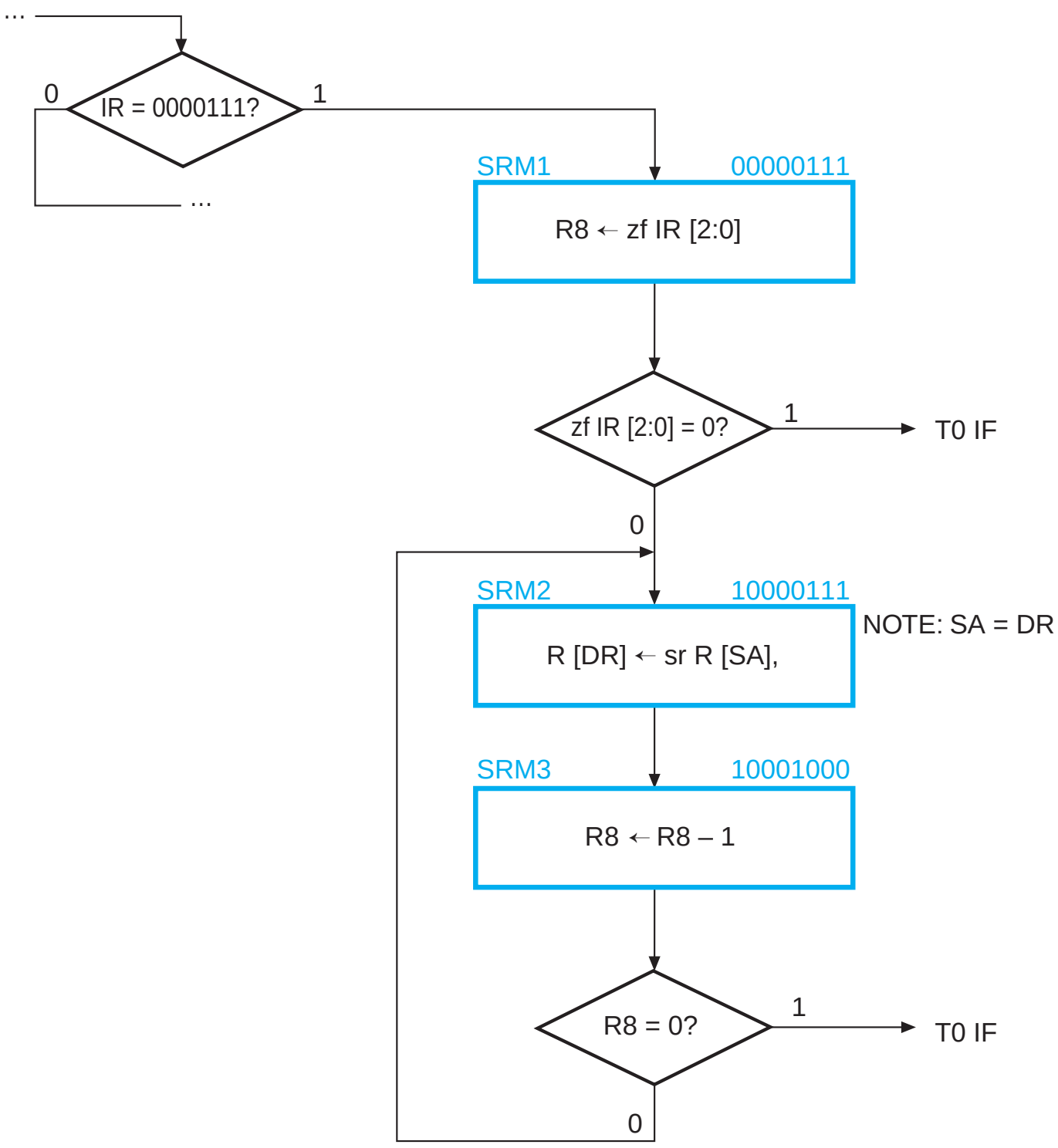
ASM Chart for Multiple-Cycle Microprogrammed Computer



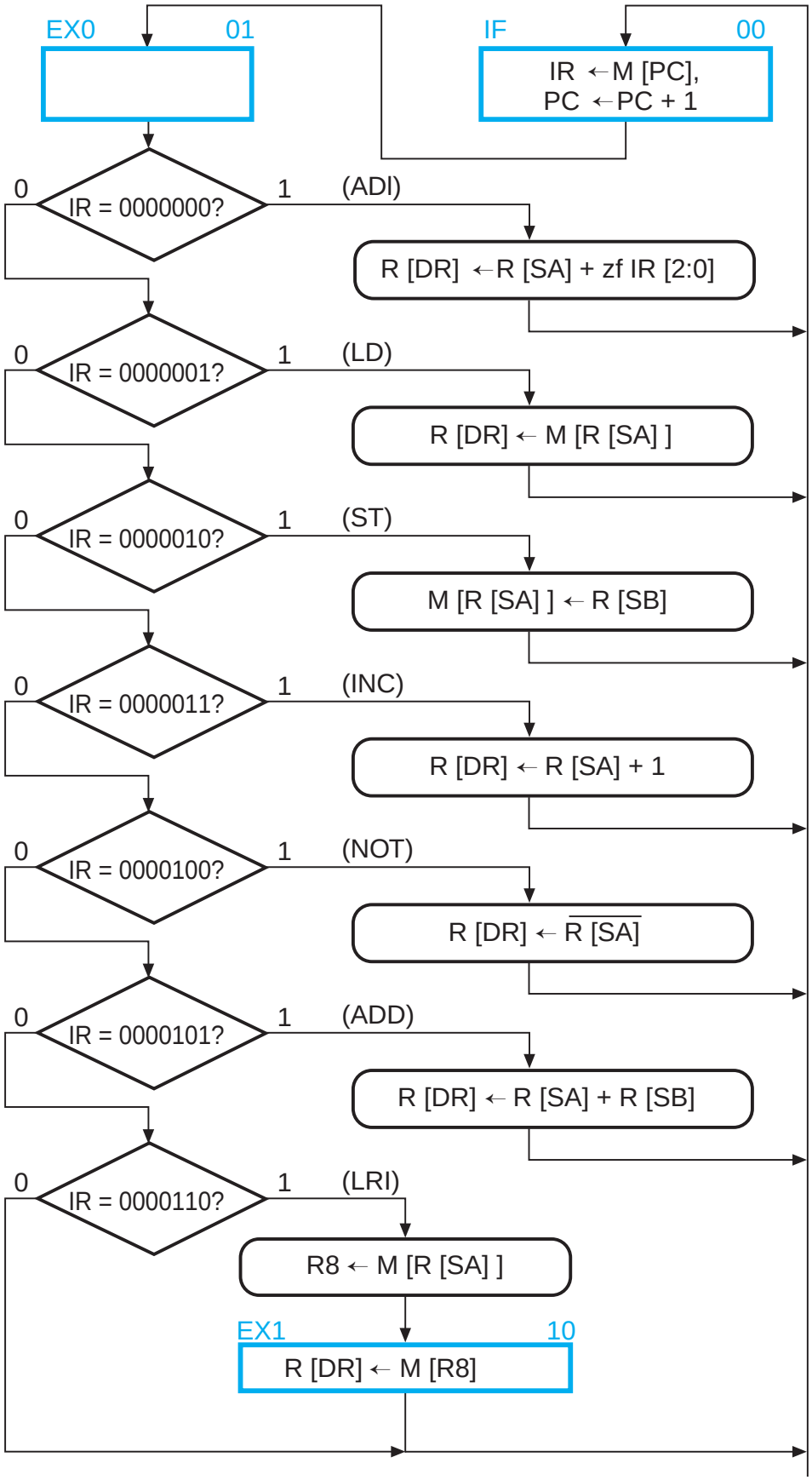
ASM Chart for Register Indirect Instruction



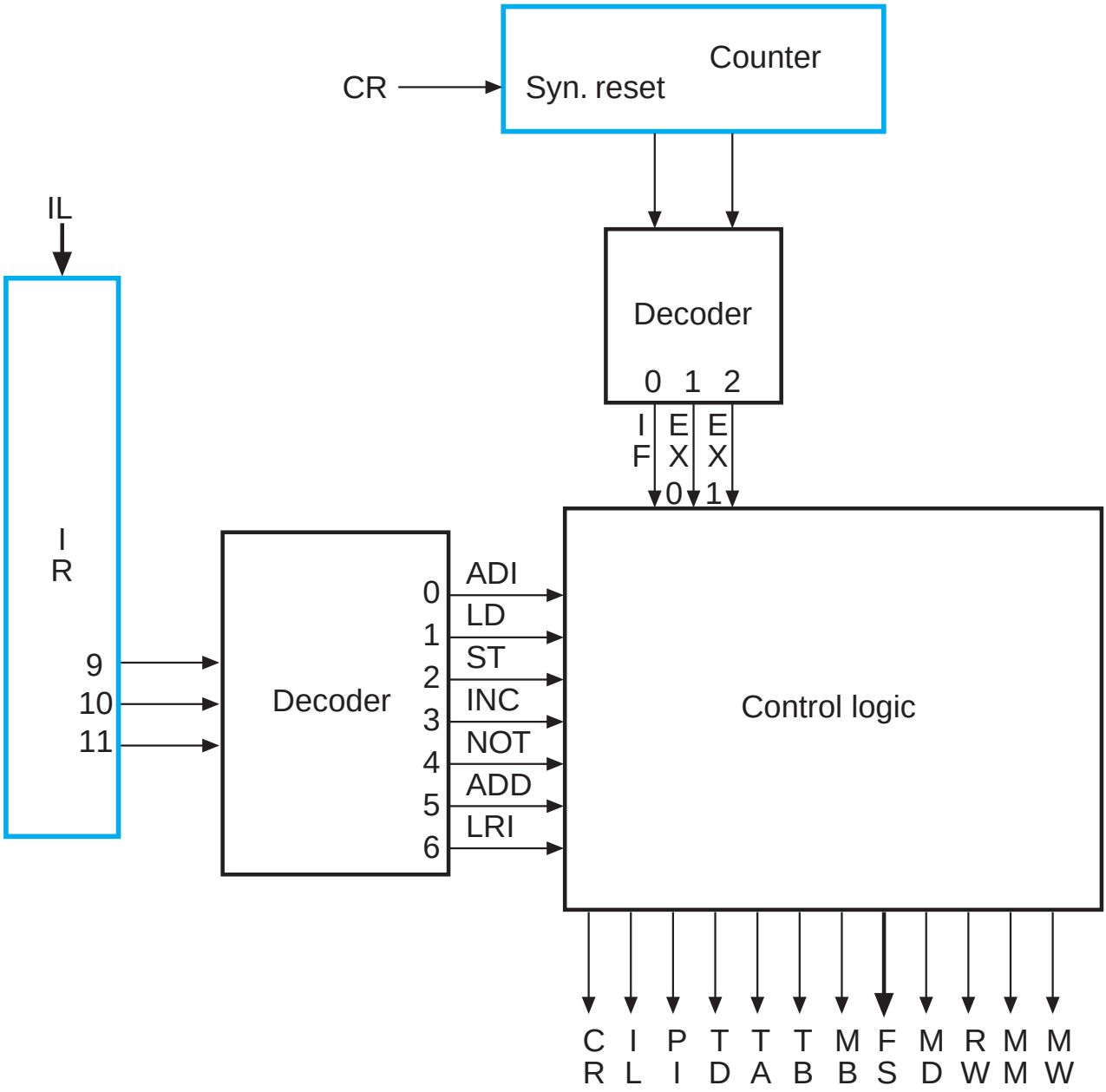
ASM Chart for Right-Shift Multiple Instruction



ASM Chart for Multiple-Cycle, Decoder-Based Computer

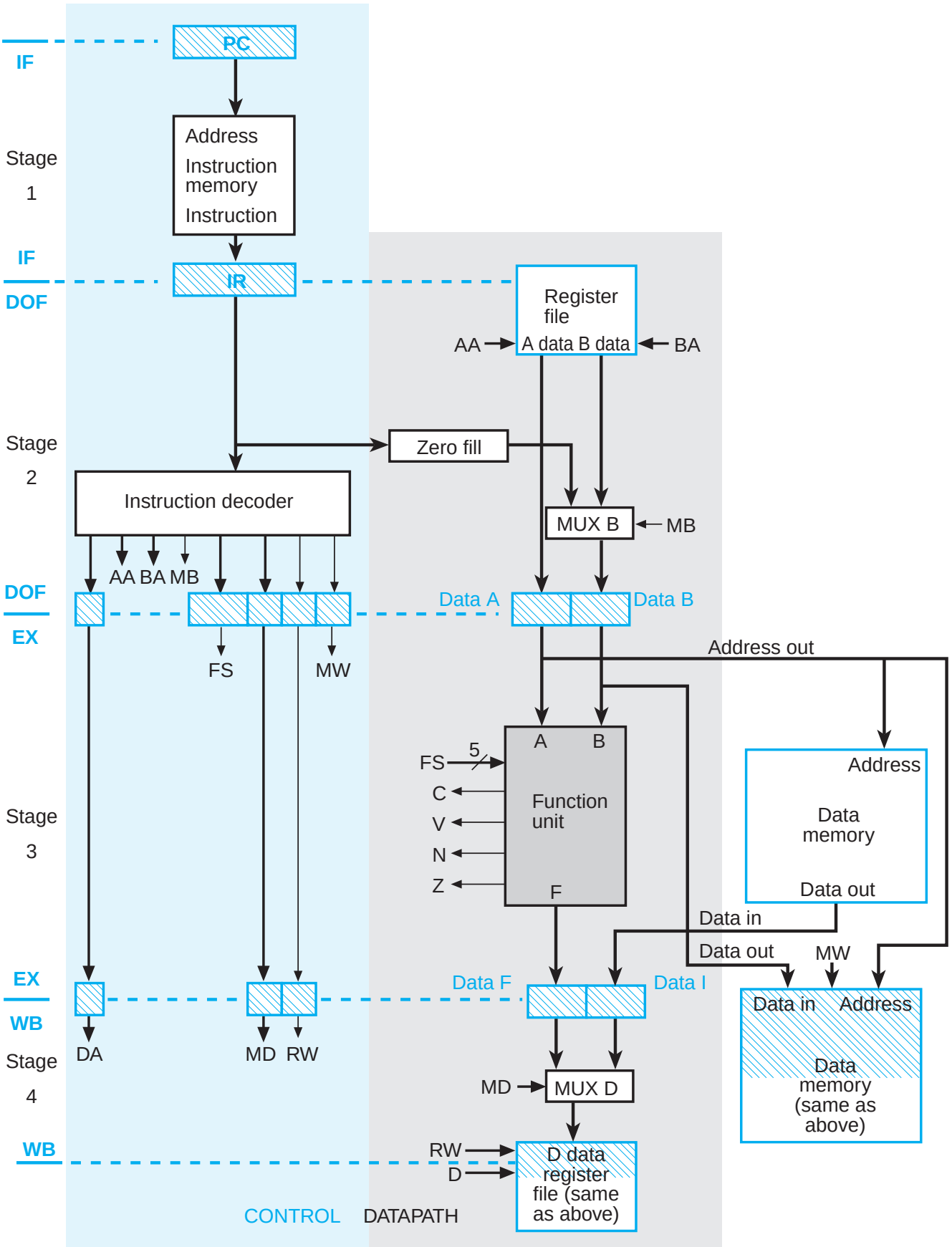


Block Diagram of Hardwired Counter and Decoder-Based, Multiple-Cycle Control Unit

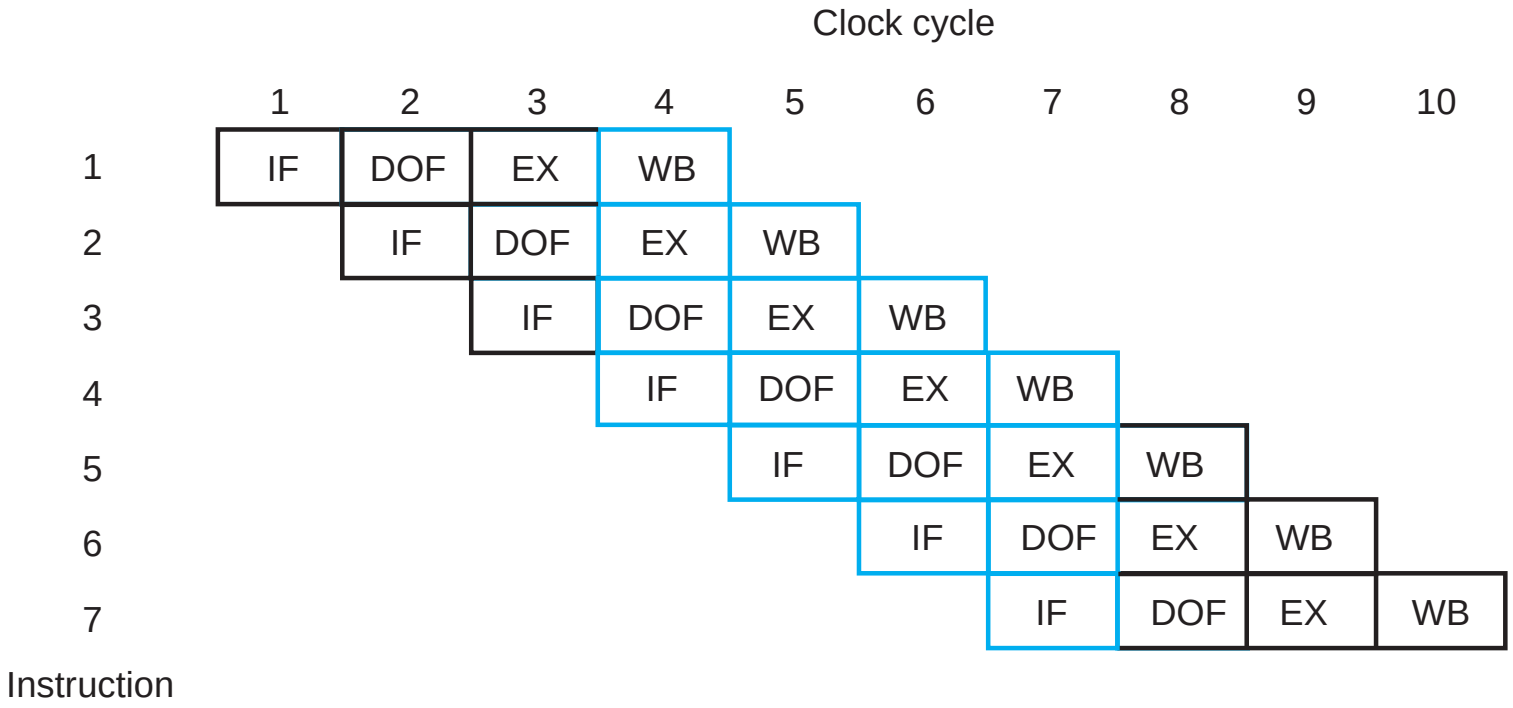


Assembly Line Analogy to Computer Pipeline

Block Diagram of Pipelined Computer



Pipeline Execution Pattern of Register Number Program



Pipeline Execution Pattern of Register Sum Program