

Fig. 10-1 CPU Register Set Diagram for CISC

TABLE 10-1
CISC Instruction Operations

Instruction	Mnemonic	Opcode	Status Effect	Instruction	Mnemonic	Opcode	Status Effect
Zero-operand Instructions				Two-operand Instructions			
No operation	NOP	000000	None	Move	MOVE	100000	None
Push registers	PSHR	000001	None	Exchange	XCH	100001	None
Pop registers	POPR	000010	None	Add	ADD	100010	ZCNV
Move string	MVS	000011	None	Add with carry	ADDC	100011	ZCNV
Return from procedure	RET	000100	None	Subtract	SUB	100100	ZCNV
Return from interrupt	RTI	000101	From stack	Subtract with borrow	SUBB	100101	ZCNV
Invalid	000110 through 001111			Multiply	MUL	100110	ZCNV
				Divide	DIV	100111	ZCNV
One-operand Instructions				Compare	CMP	101000	ZCNV
				AND	AND	101001	ZN
Push	PUSH	010000	None	OR	OR	101010	ZN
Pop	POP	010001	None	Exclusive-OR	XOR	101011	ZN
Increment	INC	010010	ZCNV	Invalid	101100 through 101111		
Decrement	DEC	010011	ZCNV	Branch Instructions			
Negate	NEG	010100	ZCNV				
Complement	COM	010101	ZN	Jump	JMP	110000	None
Logical shift right	SHR	011000	ZC	Call procedure	CALL	110001	None
Logical shift left	SHL	011001	ZC	Branch if zero	BZ	111000	None
Arithmetic shift right	SHRA	011010	ZCNV	Branch if no zero	BNZ	111001	None
Arithmetic shift left	SHLA	011011	ZCNV	Branch if carry	BC	111010	None
Rotate right	ROR	011100	ZC	Branch if no carry	BNC	111011	None
Rotate left	ROL	011101	ZC	Branch if negative	BN	111100	None
Rotate right with carry	RORC	011110	ZC	Branch if no negative	BNN	111101	None
Rotate left with carry	ROLC	011111	ZC	Branch if overflow	BV	111110	None
Invalid	010110 through 010111			Branch if no overflow	BNV	111111	None
				Invalid	110010 through 110111		

Table 10-1 CISC Instruction Operations

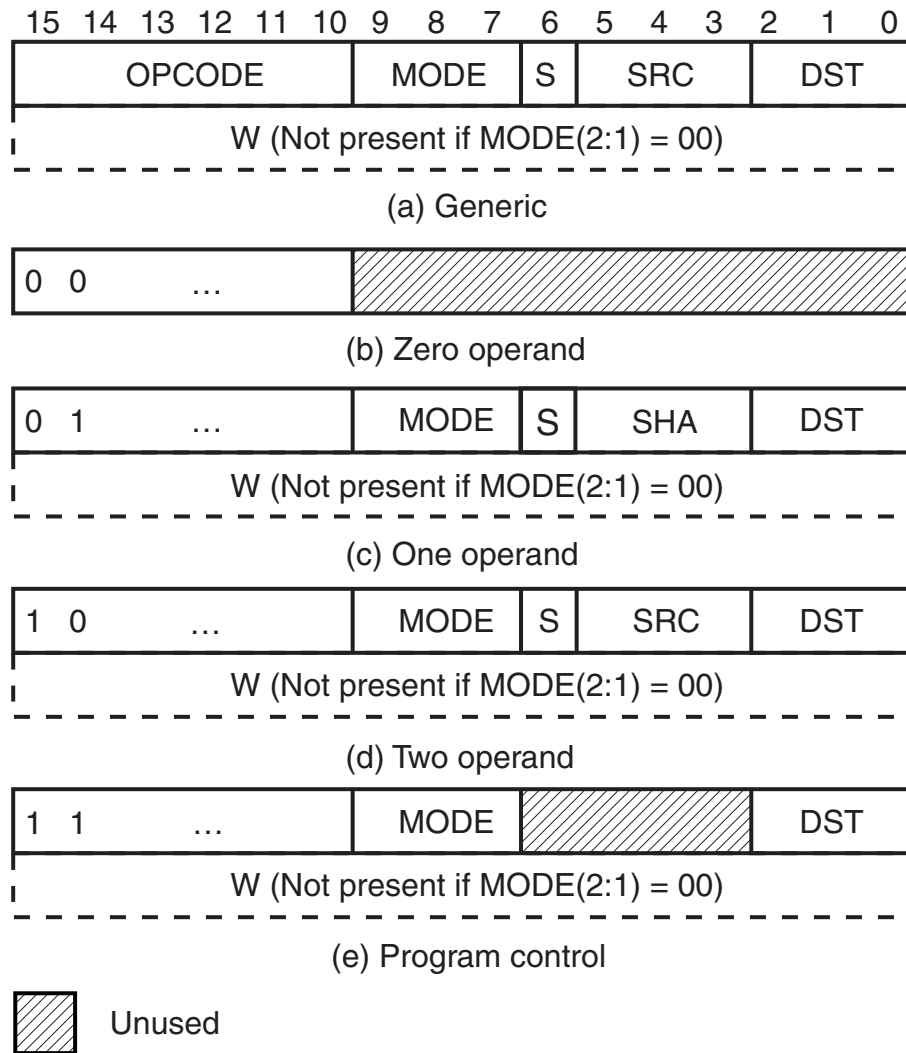


Fig. 10-2 Instruction Formats

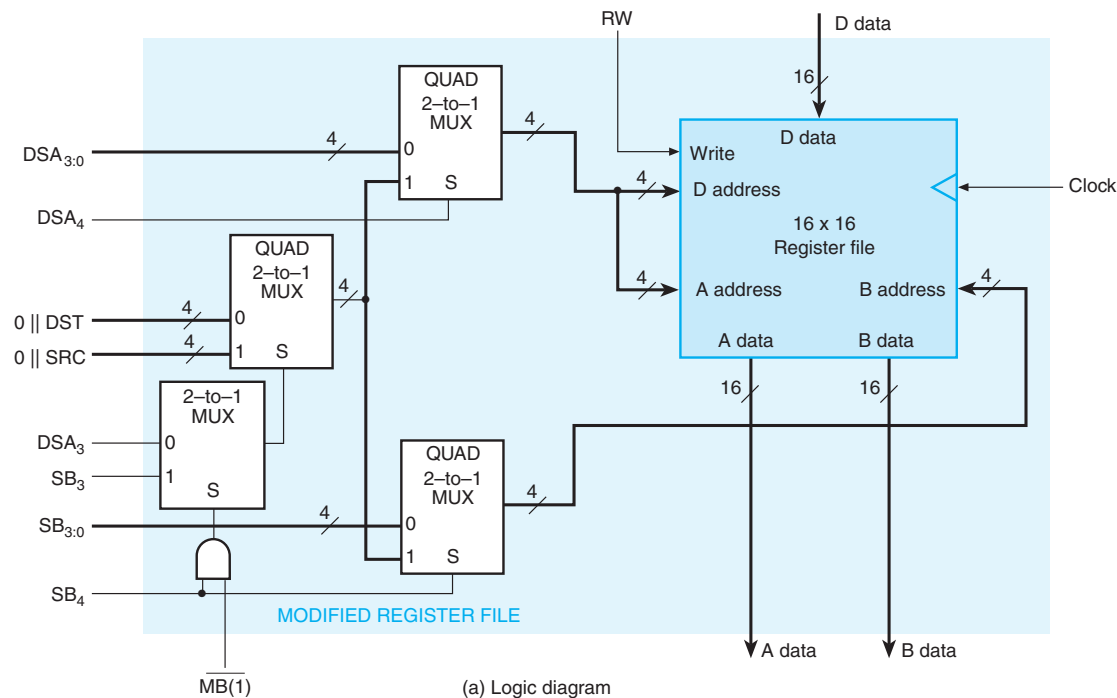
TABLE 10-2
Addressing Modes

MODE	Address Mode	Register Transfer Description of Operands		
		IR(15:14) = 01 or 11	IR(15:14) = 10, S = 0	IR(15:14) = 10, S = 1
000	Register	$R[DST]$	$R[Src], R[DST]$	$R[DST], R[Src]$
001	Register Indirect	$M[R[DST]]$	$M[R[Src]], R[DST]$	$M[R[DST]], R[Src]$
010	Immediate	W	$W, R[DST]$	$W, R[Src]$
011	Direct	$M[W]$	$M[W], R[DST]$	$M[W], R[Src]$
100	Indexed	$M[R[DST] + W]$	$M[R[Src] + W], R[DST]$	$M[R[DST] + W], R[Src]$
101	Indexed Indirect	$M[M[R[DST] + W]]$	$M[M[R[Src] + W]], R[DST]$	$M[M[R[DST] + W]],$ $R[Src]$
110	Relative	$M[PC + W]$	$M[PC + W], R[DST]$	$M[PC + W], R[Src]$
111	Relative Indirect	$M[M[PC + W]]$	$M[M[PC + W]], R[DST]$	$M[M[PC + W]], R[Src]$

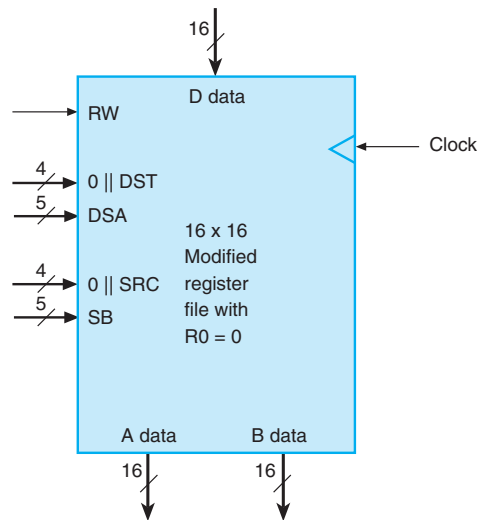
Table 10-2 Addressing Modes

0	R0 = 0
1	R1
2	R2
3	R3
4	R4
5	R5
6	R6
7	R7
8	R8
9	R9
10	R10
11	R11
12	Source Address SA
13	Source Data SD
14	Destination Address DA
15	Destination Data DD

Fig. 10-3 Register File Map

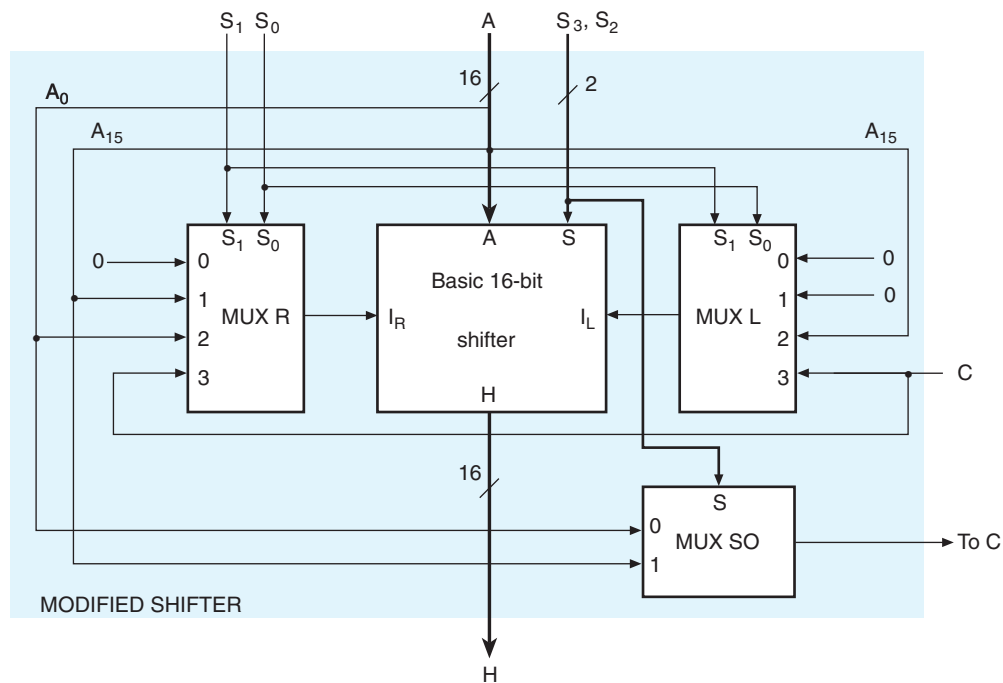


(a) Logic diagram

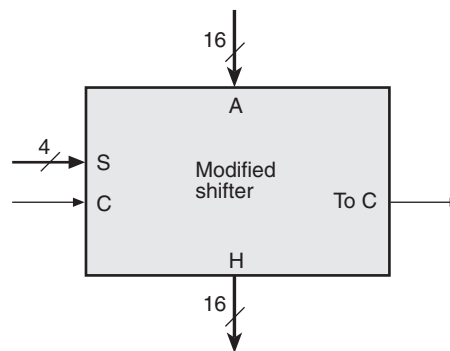


(b) Symbol

Fig. 10-4 Modifications to Register File



(a) Logic Diagram



(b) Symbol

Fig. 10-5 16-Bit Shifter and Symbol

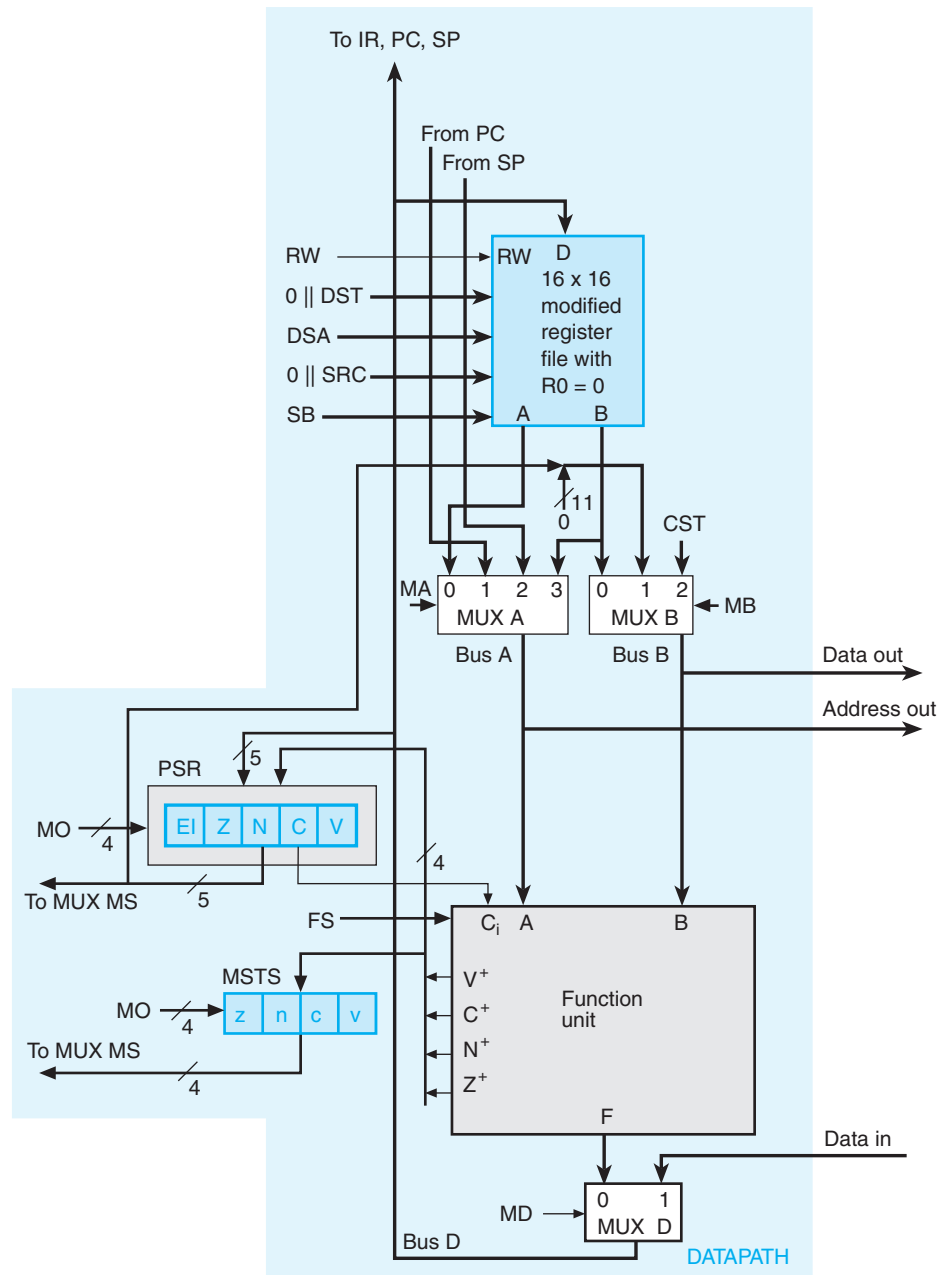


Fig. 10-6 CISC Datapath

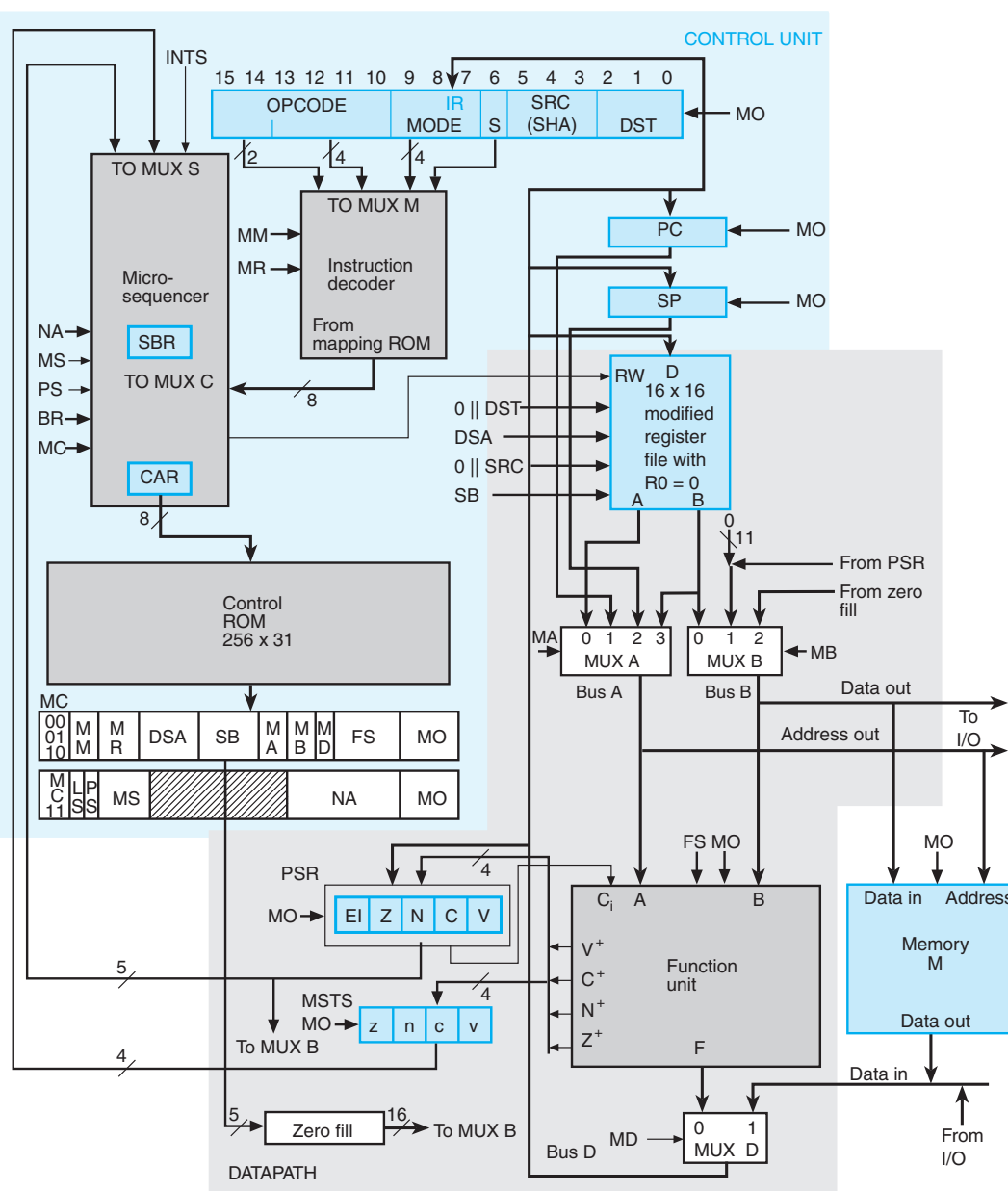
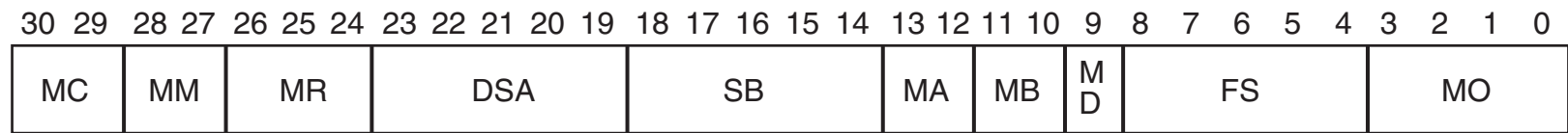


Fig. 10-7 CISC CPU



(a) Format A (MC = 00, 01, 11)



(b) Format B (MC = 11)

Fig. 10-8 Microinstruction Formats

TABLE 10-3
Control Word Encoding for Microinstruction Format A: Datapath Part

DSA, SB		MA	MB	MD			FS	
$R0 = 0$	00000	Register A	Register B	00	Function	0	$F = A$	00000
$R1$	00001	PC	PSR	01	Data in	1	$F = A + 1$	00001
$R2$	00010	SP	MCST	10			$F = A + B$	00010
$R3$	00011						$F = A + \underline{B} + 1$	00011
$R4$	00100						$F = A + \overline{B}$	00100
$R5$	00101						$F = A + \overline{B} + 1$	00101
$R6$	00110						$F = A - 1$	00110
$R7$	00111						$F = A$	00111
$R8$	01000						$F = A \wedge B$	01000
$R9$	01001						$F = A \vee B$	01010
$R10$	01010						$F = \underline{A} \oplus B$	01100
$R11$	01011						$F = A$	01110
$R12 (SA)$	01100						$F = B$	10000
$R13 (SD)$	01101						$F = \text{lsl } B$	11000
$R14 (DA)$	01110						$F = \text{lsr } B$	10100
$R15 (DD)$	01111						$F = \text{asl } B$	11001
$R[DST]$	10XXX*						$F = \text{asr } B$	10101
$R[SRC]$	11XXX*						$F = \text{rol } B$	11010
							$F = \text{ror } B$	10110
							$F = \text{rolc } B$	11011
							$F = \text{rorc } B$	10111

*Only one of *DSA* and *SB* may contain either of these patterns in any microinstruction. The other must contain a pattern beginning with a 0.

Table 10-3 Control Word Encoding for Microinstruction Format A: Datapath Part

TABLE 10-4
Control Word Information for MC and LS

Action	Format	Symbolic Notation	Codes	
			MC	LS
Increment <i>CAR</i>	A	NXT	00	—
Return from subroutine	A	RET	01	—
Map instruction into <i>CAR</i>	A	MAP	10	—
Jump to <i>NA</i> if <i>ST</i> bit is satisfied; else increment <i>CAR</i>	B	BR	11	0
Call subroutine at <i>NA</i> if <i>ST</i> bit is satisfied; else increment <i>CAR</i>	B	CALL	11	1

Table 10-4 Control Word Information for MC and LS

TABLE 10-5
Control Word Information for Polarity Bit and Multiplexer S
in Format B Microinstructions

PS			MS		
Action	Symbolic Notation	Code	Condition Status Signal	Symbolic Notation	Code
Pass status bit unchanged	TS	0	Constant 1 for unconditional transfer	BU	0000
Complement status bit	CS	1	Zero <i>PSR</i> bit	BZ	0001
			Negative <i>PSR</i> bit	BN	0010
			Carry <i>PSR</i> bit	BC	0011
			Overflow <i>PSR</i> bit	BV	0100
			Enable-interrupt <i>PSR</i> bit	EI	0101
			Zero <i>MSTS</i> bit	Bz	0110
			Negative <i>MSTS</i> bit	Bn	0111
			Carry <i>MSTS</i> bit	Bc	1000
			Overflow <i>MSTS</i> bit	Bv	1001
			Interrupt signal <i>INTS</i>	BI	1010

Table 10-5 Control Word Information for Polarity Bit and Multiplexer S in Format B Microinstructions

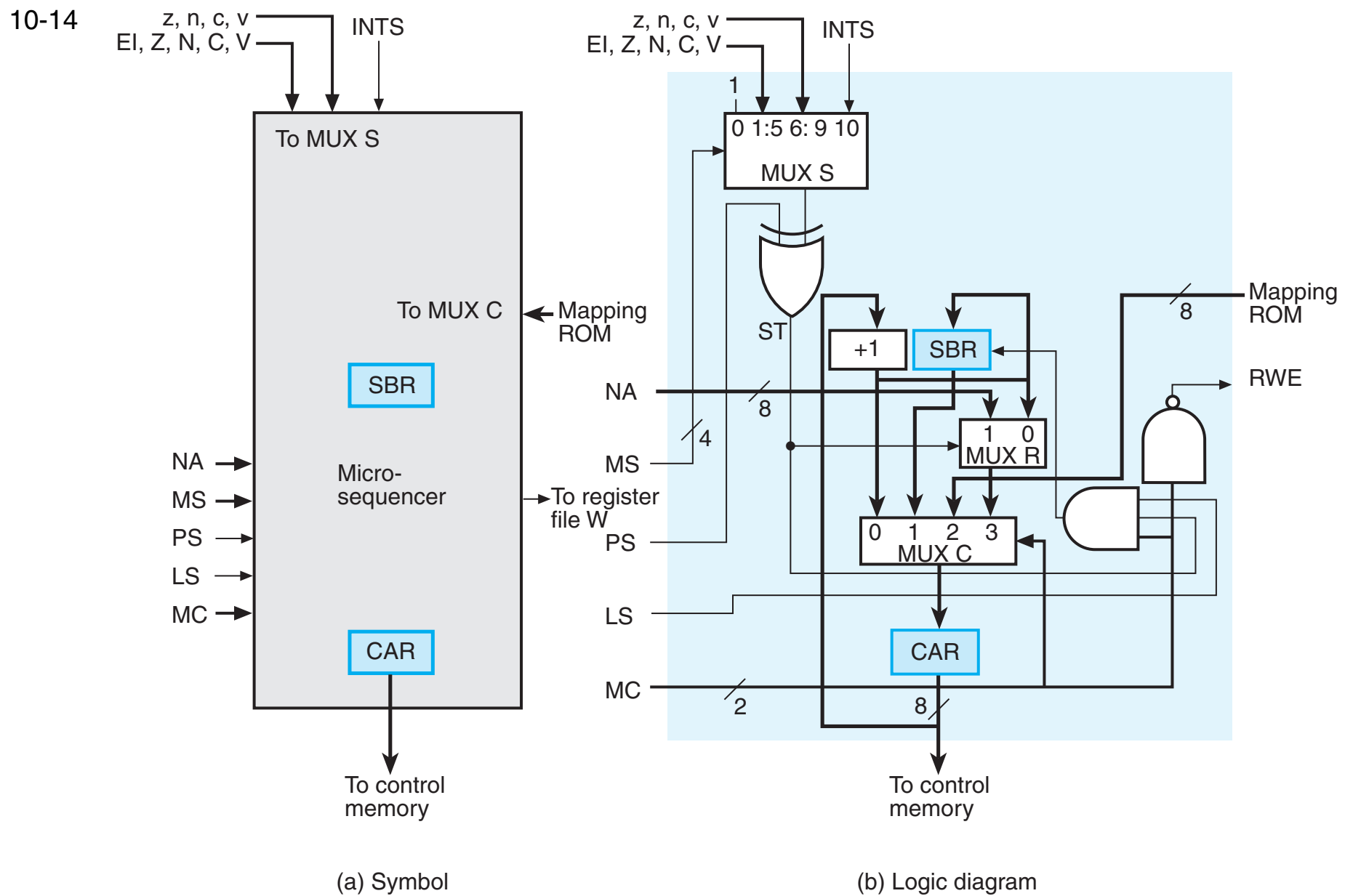


Fig. 10-9 The Microsequencer

TABLE 10-6
Control Word Encoding for Instruction Decoder

MM		MR	
Select	Code	Region	Code
OPCODE(5:3)	00	0	000
OPCODE(3:0)	01	1	001
MODE S	10	2	010
MODE S	11	3	011
		4	100
		5	101
		6	110
		7	111

Table 10-6 Control Word Encoding for Instruction Decoder

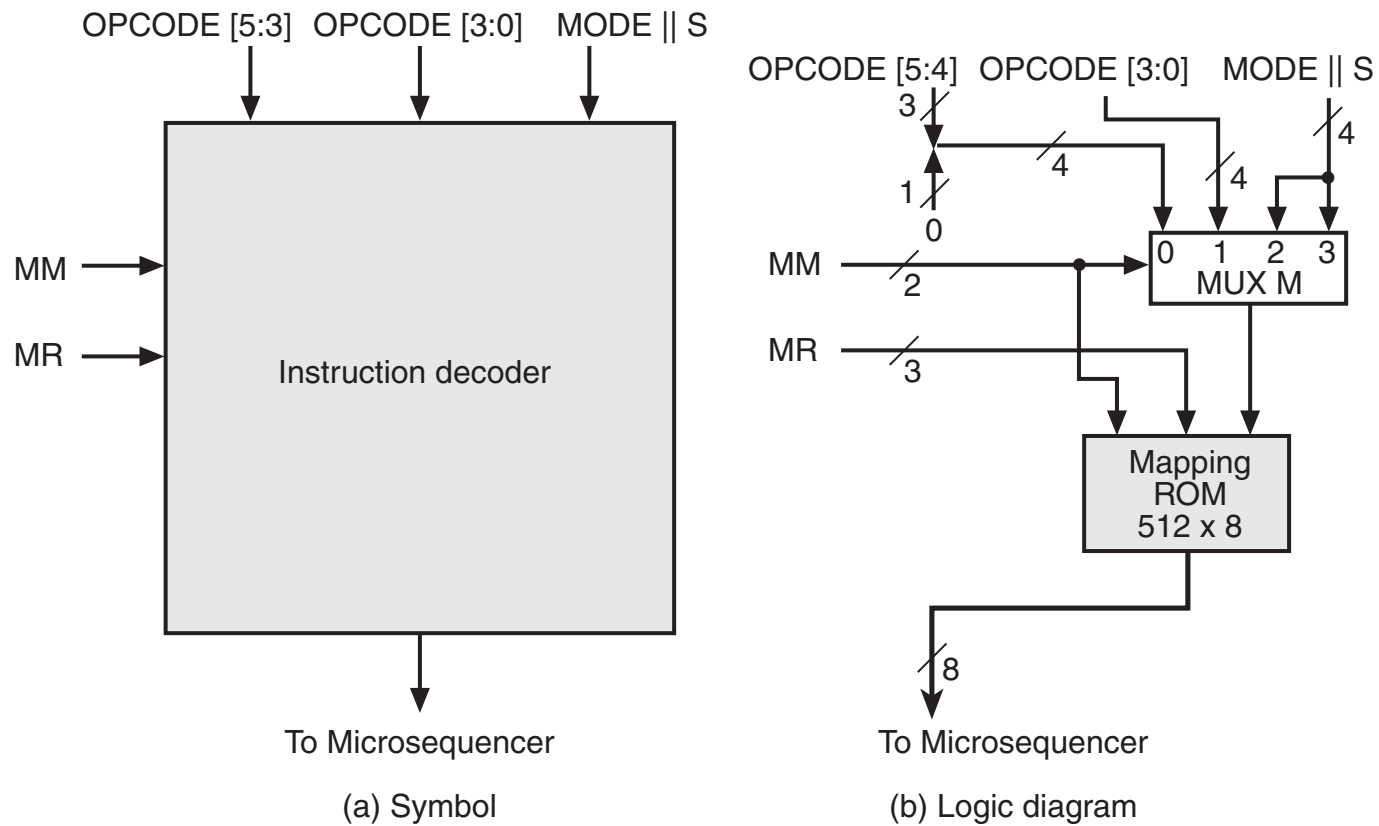


Fig. 10-10 Instruction Decoder

TABLE 10-7
Control Information for Miscellaneous Operations for
Format A and B Microinstructions

MO		
Operations	Symbolic Notation	Code
No operation	—	0000
$INACK = 1$	INCK	0001
Memory write*	WRITE	0010
Load PC^*	LPC	0011
Load IR and increment PC^*	DPC	0100
Increment PC	IPC	0101
Load PSR^*	LST	0110
Load SP^*	LSP	0111
Decrement SP	DSP	1000
Decrement SP and memory write*	DSM	1001
Increment SP	ISP	1010
Select C as C_{in} for arithmetic and action EST as follows	CIN	1011
Enable update of status bits Z , N , C , and V	EST	1100
Enable update of status bits Z and C	EZC	1101
Enable update of status bits N and Z	ENZ	1110
Enable update of microstatus bits z , n , c , and v	EMS	1111

* Prevents write to register file

Table 10-7 Control Information for Miscellaneous Operations for Format A and B Microinstructions

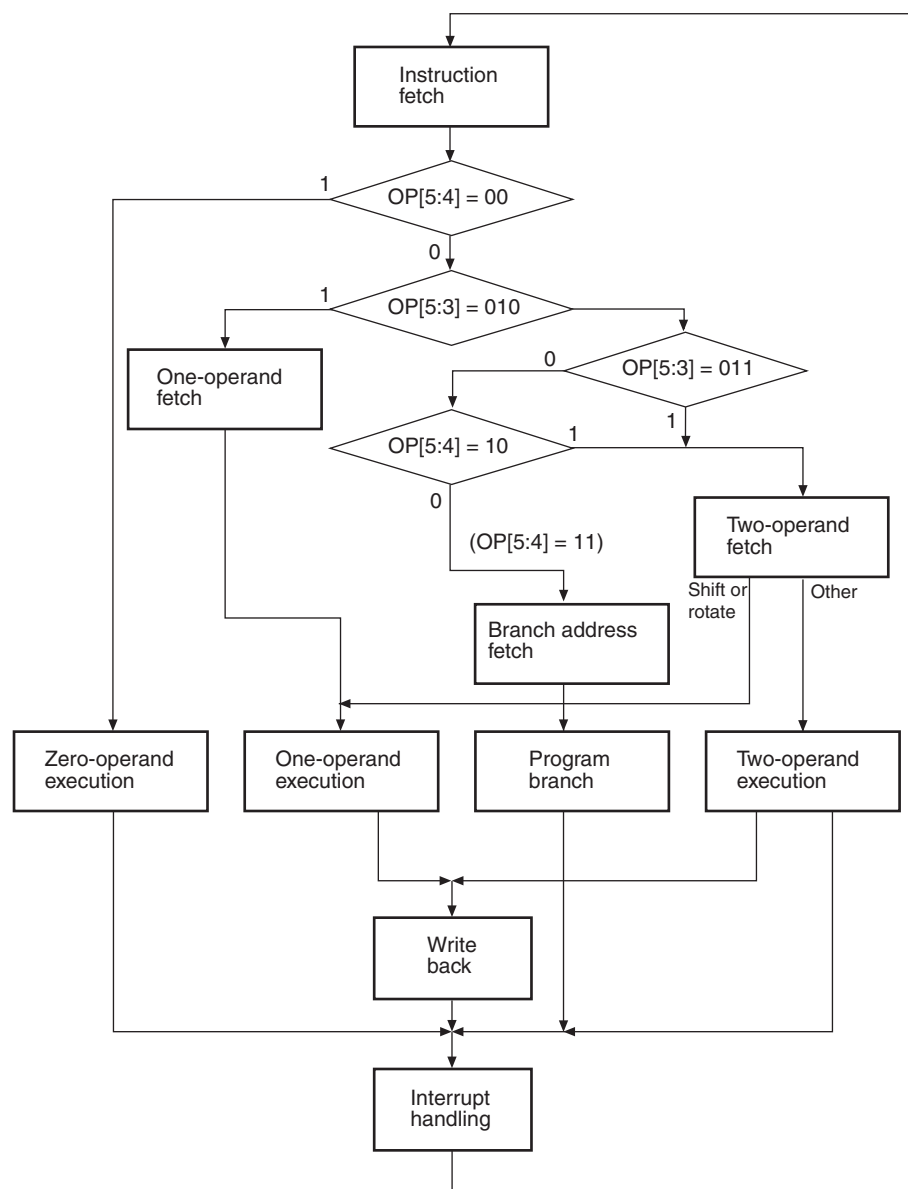


Fig. 10-11 ASM-like Chart of Microroutines

TABLE 10-8
 Programming Table for Mapping ROM

Location of Microinstructions Performing a Mapping	ROM Inputs			ROM Outputs—Location of Next Microinstruction
Symbolic Microaddress	MM	MR	IR Bits and Match Field	Symbolic Microaddress
IF1	00	000	OPCODE(5:3) 0 = 00X0	0EX
IF1	00	000	OPCODE(5:3) 0 = 0100	1OF
IF1	00	000	OPCODE(5:3) 0 = 0110	2OF
IF1	00	000	OPCODE(5:3) 0 = 10X0	2OF
IF1	00	000	OPCODE(5:3) 0 = 1011	BAF
In 1OF Microroutine	00	001	OPCODE(5:3) 0 = XXXX	1EX
In 2OF Microroutine	00	010	OPCODE(5:3) 0 = 0110	1EX
In 2OF Microroutine	00	010	OPCODE(5:3) 0 = 10X0	2EX
In BAF Microroutine	00	011		BEX
0EX	01	000	OPCODE(3:0) = O ₃ O ₂ O ₁ O ₀	16 0-Op EX Microinstruction Addresses
1EX	01	001	OPCODE(3:0) = O ₃ O ₂ O ₁ O ₀	16 1-Op EX Microinstruction Addresses
2EX	01	010	OPCODE(3:0) = O ₃ O ₂ O ₁ O ₀	16 2-Op Ex Microinstruction Addresses
BEX	01	011	OPCODE(3:0) = O ₃ O ₂ O ₁ O ₀	16 Br-Op EX Microinstruction Addresses
In EX Microroutines	01	100	OPCODE(3:0) = XXXX	WB0
In EX Microroutines	01	101	OPCODE(3:0) = XXXX	INT0
1OF	10	001	MODE S = M ₂ M ₁ M ₀ X	8 1-Op OF Microinstruction Addresses
2OF	10	010	MODE S = M ₂ M ₁ M ₀ S	16 2-Op OF Microinstruction Addresses
BAF	10	011	MODE S = M ₂ M ₁ M ₀ X	8 Br-Op Microinstruction Addresses
XCH2	10	100	MODE S = XXX1	XCH3
XCH2	10	100	MODE S = 0000	XCH3
XCH2	10	100	MODE S = XX10	XCH4
XCH2	10	100	MODE S = X1X0	XCH4
XCH2	10	100	MODE S = 1XX0	XCH4
WB0	11	000	MODE S = XXX0	WB1
WB0	11	000	MODE S = 0001	WB1
WB0	11	000	MODE S = 1XX1	WB2
WB0	11	000	MODE S = X1X1	WB2
WB0	11	000	MODE S = XX11	WB2
In WB microroutine	11	001	MODE S = XXXX	INT0
In INT microroutine	11	010	MODE S = XXXX	IF0

Table 10-8 Programming Table for Mapping ROM

TABLE 10-9
Instruction Fetch Microroutine

Sym			MM	MR	DSA					FS	
Add	Register Transfer Description	MC	/LS	/PS	/MS	SB	MA	MB	MD	/NA	MO
IF0	$IR \leftarrow M[PC], PC \leftarrow PC + 1$	0	0	0	00	00	1	0	1	00	4
IF1	$CAR \leftarrow ROM[00000_2 \parallel OPCODE(5:4) \parallel 00_2]$	2	0	0	00	00	0	0	0	00	0

Table 10-9 Instruction Fetch Microroutine

TABLE 10-10
One-operand Fetch Microroutine

Sym Add	Register Transfer Description	MC	MM /LS	MR /PS	DSA /MS	SB	MA	MB	MD	FS /NA	MO
1OF	$CAR \leftarrow ROM[10001_2 \parallel MODE \parallel S]$	2	2	1	00	00	0	0	0	00	0
1RG	$DD \leftarrow R[DST], CAR \leftarrow 1EX(ROM)$	2	0	1	0F	10	0	0	0	00	0
1RGI0	$DA \leftarrow R[DST]$	0	0	0	0E	10	0	0	0	00	0
1RGI1	$DD \leftarrow M[DA], CAR \leftarrow 1EX(ROM)$	2	0	1	0F	0E	3	0	1	00	0
1IM	$DD \leftarrow M[PC], PC \leftarrow PC + 1,$ $CAR \leftarrow EX1(ROM)$	2	0	1	0F	00	1	0	1	00	5
1DR0	$DA \leftarrow M[PC], PC \leftarrow PC + 1$	0	0	0	0E	00	1	0	1	00	5
1DR1	$DD \leftarrow M[DA], CAR \leftarrow 1EX(ROM)$	2	0	1	0F	0E	3	0	1	00	0
1ID0	$DA \leftarrow M[PC], PC \leftarrow PC + 1$	0	0	0	0E	00	1	0	1	00	5
1ID1	$DA \leftarrow DA + R[DST]$	0	0	0	0E	10	0	0	0	02	0
1ID2	$DD \leftarrow M[DA], CAR \leftarrow 1EX(ROM)$	2	0	1	0F	0E	3	0	1	00	0
1IDI0	$DA \leftarrow M[PC], PC \leftarrow PC + 1$	0	0	0	0E	00	1	0	1	00	5
1IDI1	$DA \leftarrow DA + R[DST]$	0	0	0	0E	10	0	0	0	02	0
1IDI2	$DA \leftarrow M[DA]$	0	0	0	0E	00	0	0	1	00	0
1IDI3	$DD \leftarrow M[DA], CAR \leftarrow 1EX(ROM)$	2	0	1	0F	0E	3	0	1	00	0
1RL0	$DA \leftarrow M[PC], PC \leftarrow PC + 1$	0	0	0	0E	00	1	0	1	00	5
1RL1	$DA \leftarrow DA + PC$	0	0	0	0E	0E	1	0	0	02	0
1RL2	$DD \leftarrow M[DA], CAR \leftarrow 1EX(ROM)$	2	0	1	0F	0E	3	0	1	00	0
1RLI0	$DA \leftarrow M[PC], PC \leftarrow PC + 1$	0	0	0	0E	00	1	0	1	00	5
1RLI1	$DA \leftarrow DA + PC$	0	0	0	0E	0E	1	0	0	02	0
1RLI2	$DA \leftarrow M[DA]$	0	0	0	0E	00	0	0	1	00	0
1RLI3	$DD \leftarrow M[DA], CAR \leftarrow 1EX(ROM)$	2	0	1	0F	0E	3	0	1	00	0

Table 10-10 One-operand Fetch Microroutine

TABLE 10-11
Two-operand Fetch Microroutine Examples

Sym Add	Register Transfer Description	MC	MM /LS	MR /PS	DSA /MS	SB	MA	MB	MD	FS /NA	MO
2OF	$CAR \leftarrow ROM[10010_2 \parallel MODE \parallel S]$	2	2	2	00	00	0	0	0	00	0
2DR0	$SA \leftarrow M[PC], PC \leftarrow PC + 1$	0	0	0	0C	00	1	0	1	00	5
2DR1	$SD \leftarrow M[SA]$	0	0	0	0D	0C	0	0	1	10	0
2DR2	$DD \leftarrow R[DST], CAR \leftarrow 2EX(ROM)$	2	0	2	0F	10	0	0	0	10	0
2DRS0	$DA \leftarrow M[PC], PC \leftarrow PC + 1$	0	0	0	0E	00	1	0	1	00	5
2DRS1	$DD \leftarrow M[DA]$	0	0	0	0F	0E	3	0	1	00	0
2DRS2	$SD \leftarrow R[Src], CAR \leftarrow 2EX(ROM)$	2	0	2	0D	11	0	0	0	10	0

Table 10-11 Two-operand Fetch Microroutine Examples

TABLE 10-12
 Example of Branch Address Fetch

Sym Add	Register Transfer Description	MM/ MR/ DSA/								FS	
		MC	LS	PS	MS	SB	MA	MB	MD	/NA	MO
BAF	$CAR \leftarrow ROM[10011_2 \parallel MODE \parallel S]$	2	2	3	00	00	0	0	0	00	0
BDR0	$DA \leftarrow M[PC], PC \leftarrow PC + 1,$ $CAR \leftarrow BEX(ROM)$	2	0	3	0E	00	0	1	1	00	5

Table 10-12 Example of Branch Address Fetch

TABLE 10-13
Zero-operand Execution Microroutine Examples

Sym Add	Register Transfer Description	MC	MM /LS	MR /PS	DSA /MS	SB	MA	MB	MD	FS /NA	MO
0EX	$CAR \leftarrow ROM[01000_2 \parallel OPCODE[3:0]]$	2	1	0	00	00	0	0	0	00	0
PSHR0	$SP \leftarrow SP - 1$	0	0	0	00	00	0	0	0	00	8
PSHR1	$M[SP] \leftarrow R1, SP \leftarrow SP - 1$	0	0	0	00	01	2	0	0	00	9
PSHR2	$M[SP] \leftarrow R2, SP \leftarrow SP - 1$	0	0	0	00	02	2	0	0	00	9
PSHR3	$M[SP] \leftarrow R3, SP \leftarrow SP - 1$	0	0	0	00	03	2	0	0	00	9
PSHR4	$M[SP] \leftarrow R4, SP \leftarrow SP - 1$	0	0	0	00	04	2	0	0	00	9
PSHR5	$M[SP] \leftarrow R5, SP \leftarrow SP - 1$	0	0	0	00	05	2	0	0	00	9
PSHR6	$M[SP] \leftarrow R6, SP \leftarrow SP - 1$	0	0	0	00	06	2	0	0	00	9
PSHR7	$M[SP] \leftarrow R7, CAR \leftarrow INT0(ROM)$	2	1	5	00	07	2	0	0	00	2
RET0	$PC \leftarrow M[SP]$	0	0	0	00	00	2	0	1	00	3
RET1	$SP \leftarrow SP + 1, CAR \leftarrow INT0(ROM)$	2	1	5	00	00	0	0	0	00	A
RTI0	$PSR \leftarrow M[SP]$	0	0	0	00	00	2	0	1	00	6
RTI1	$SP \leftarrow SP + 1$	0	0	0	00	00	0	0	0	00	A
RTI3	$PC \leftarrow M[SP],$	0	0	0	00	00	2	0	1	00	3
RTI4	$SP \leftarrow SP + 1, CAR \leftarrow INT0(ROM)$	2	1	5	00	00	0	0	0	00	A

Table 10-13 Zero-operand Execution Microroutine Examples

TABLE 10-14
One-operand Execution Microroutine Examples

Sym			MM	MR	DSA					FS	
Add	Register Transfer Description	MC	/LS	/PS	/MS	SB	MA	MB	MD	/NA	MO
1EX	$CAR \leftarrow ROM[01001_2 \parallel OPCODE(3:0)]$	2	1	1	00	00	0	0	0	00	0
INC	$DD \leftarrow DD + 1, CAR \leftarrow WB0(ROM)$	2	1	4	0F	00	0	0	0	01	C
DEC	$DD \leftarrow DD - 1, CAR \leftarrow WB0(ROM)$	2	1	4	0F	00	0	0	0	06	C
NEG0	$DD \leftarrow \overline{DD}$	0	0	0	0F	00	0	0	0	0E	0
NEG	$DD \leftarrow \overline{DD} + 1, CAR \leftarrow WB0(ROM)$	2	1	4	0F	00	0	0	0	01	C
COM	$DD \leftarrow \overline{DD}, CAR \leftarrow WB0(ROM)$	2	1	4	0F	00	0	0	0	0E	E
SHR0	$R9 \leftarrow SD$	0	0	0	09	0D	0	0	0	10	0
SHR1	$R9 \leftarrow R9$ (Set MSTs)	0	0	0	09	0	0	0	0	00	F
SHR2	$z: CAR \leftarrow SHR6$	3	0	0	6	00	0	0	0	SHR6	0
SHR3	$DD \leftarrow 0 \parallel DD(15:1)$	0	0	0	0F	0F	0	0	0	11	0
SHR4	$R9 \leftarrow R9 - 1$	0	0	0	09	00	0	0	0	06	F
SHR5	$\bar{z}: CAR \leftarrow SHR3$	3	0	1	6	00	0	0	0	SHR3	0
SHR6	$DD \leftarrow DD, CAR \leftarrow WB0(ROM)$	2	1	4	0F	00	0	0	0	00	D

Table 10-14 One-operand Execution Microroutine Examples

TABLE 10-15
Two-operand Execution Microroutine Examples

Sym Add	Register Transfer Description	MM MR DSA				FS					
		MC	/LS	/PS	/MS	SB	MA	MB	MD	/NA	MO
2EX	$CAR \leftarrow ROM[01010_2 \parallel OPCODE(0:3)]$	2	1	2	00	00	0	0	0	00	0
MOVE	$DD \leftarrow SD, CAR \leftarrow WB0(ROM)$	2	1	4	0F	0D	0	0	0	10	0
XCH0	$R9 \leftarrow SD$	0	0	0	09	0F	0	0	0	10	0
XCH1	$SD \leftarrow DD$	0	0	0	0E	0F	0	0	0	10	0
XCH2	$DD \leftarrow R9, CAR \leftarrow ROM[10100_2 \parallel MODE \parallel S]$	2	2	4	0F	09	0	0	0	10	0
XCH3	$R[SA] \leftarrow SD, CAR \leftarrow WB0(ROM)$	2	1	4	11	0D	0	0	0	10	0
XCH4	$M[SA] \leftarrow SD, CAR \leftarrow WB0(ROM)$	2	1	4	0C	0D	0	0	0	00	2
ADD	$DD \leftarrow DD + SD, CAR \leftarrow WB0(ROM)$	2	1	4	0F	0D	0	0	0	02	C
ADDC	$DD \leftarrow DD + SD + C, CAR \leftarrow WB0(ROM)$	2	1	4	0F	0D	0	0	0	03	B
CMP	$DD \leftarrow DD - SD, CAR \leftarrow INT0(ROM)$	2	1	5	0F	0D	0	0	0	05	C

Table 10-15 Two-operand Execution Microroutine Examples

TABLE 10-16
Program Branch Microroutine Examples

Sym	Register Transfer Description	MC	MM /LS	MR /PS	DSA /MS	SB	MA	MB	MD	FS /NA	MO
BEX	$CAR \leftarrow ROM[01011_2 \parallel OPCODE(3:0)]$	2	1	3	00	00	0	0	0	00	0
JMP	$PC \leftarrow DA, CAR \leftarrow INT0(ROM)$	2	1	5	0E	00	0	0	0	00	3
CALL0	$R8 \leftarrow PC$	0	0	0	08	00	1	0	0	00	0
CALL1	$SP \leftarrow SP - 1$	0	0	0	0	00	0	0	0	00	8
CALL2	$M[SP] \leftarrow R8$	0	0	0	0	08	2	0	0	00	2
CALL3	$PC \leftarrow DA, CAR \leftarrow INT0(ROM)$	2	1	5	0E	00	0	0	0	00	3
BZ0	$Z: CAR \leftarrow BRA$	3	0	0	1	—	—	—	—	BRA	0
BZ1	$CAR \leftarrow INT0(ROM)$	2	1	5	00	00	0	0	0	00	0
BRA	$PC \leftarrow DA, CAR \leftarrow INT0(ROM)$	2	1	5	0E	00	0	0	0	00	3
BNZ0	$\bar{Z}: CAR \leftarrow BRA$	3	0	1	1	—	—	—	—	BRA	0
BNZ1	$CAR \leftarrow INT0(ROM)$	2	1	5	00	00	0	0	0	00	0

Table 10-16 Program Branch Microroutine Examples

TABLE 10-17
 Write Back Microroutine

Sym			MM	MR	DSA					FS	M
Add	Register Transfer Description	MC	/LS	/PS	/MS	SB	MA	MB	MD	/NA	O
WB0	$CAR \leftarrow ROM[11000_2 \parallel MODE \parallel S]$	2	3	0	00	00	0	0	0	00	0
WB1	$R[DST] \leftarrow DD, CAR \leftarrow INT0(ROM)$	2	1	5	10	0F	0	0	0	10	0
WB2	$M[DA] \leftarrow DD, CAR \leftarrow INT0(ROM)$	2	1	5	0E	0F	0	0	0	00	2

Table 10-17 Write Back Microroutine

□ **TABLE 10-18**
Interrupt-Handling Microroutine

Sym			MM	MR	DSA					FS	
Add	Register Transfer Description	MC	/LS	/PS	/MS	SB	MA	MB	MD	/NA	MO
INT0	$\overline{INTS}: CAR \leftarrow IF0$	3	0	1	A	00	00	00	00	IF0	0
INT1	$R8 \leftarrow PC$	0	0	0	08	00	1	0	0	00	0
INT2	$SP \leftarrow SP - 1$	0	0	0	00	00	0	0	0	00	8
INT3	$M[SP] \leftarrow R8, SP \leftarrow SP - 1$	0	0	0	00	08	2	0	0	00	9
INT4	$M[SP] \leftarrow PSR$	0	0	0	00	00	2	1	0	00	9
INT5	$PSR \leftarrow 0$	0	0	0	00	00	0	0	0	00	7
INT6	$INACK \leftarrow 1$	0	0	0	00	00	0	0	0	00	1
INT7	$PC \leftarrow IVAD, CAR \leftarrow IF0(ROM)$	2	3	2	00	00	0	0	1	00	3

Table 10-18 Interrupt-Handling Microroutine

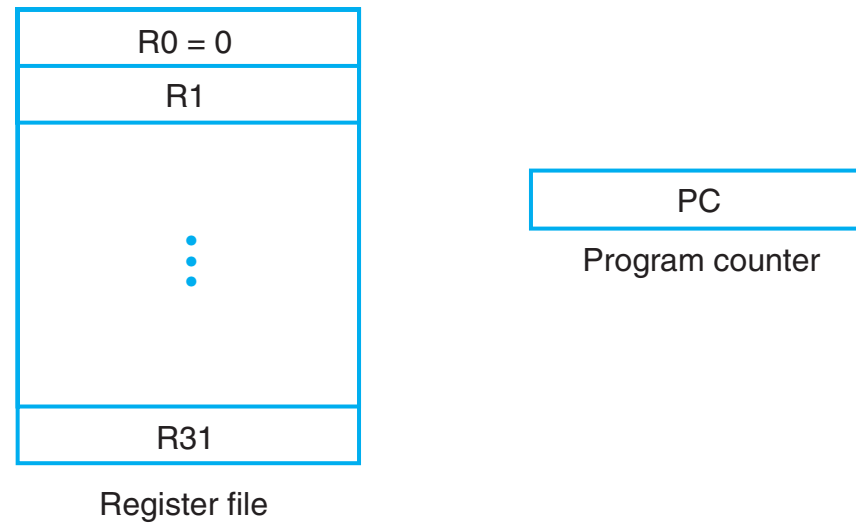


Fig. 10-12 CPU Register Set Diagram for RISC

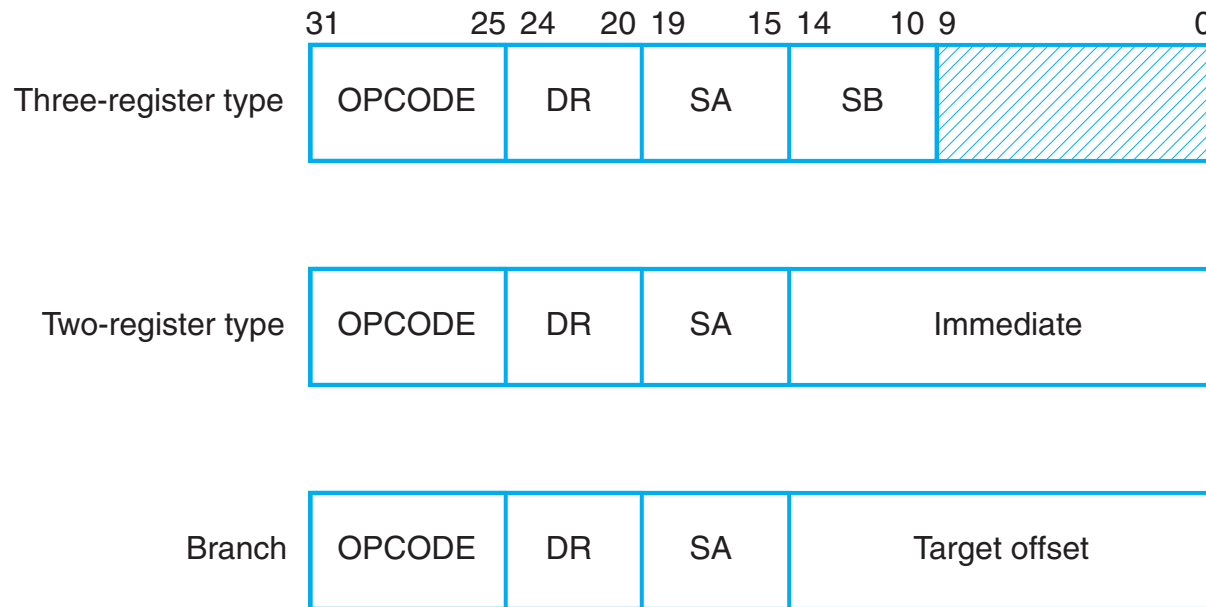


Fig. 10-13 RISC CPU Instruction Formats

□ **TABLE 10-19**
RISC Instruction Operations

Operation	Symbolic Notation	Opcode	Action
No Operation	NOP	0000000	None
Add	ADD	0000010	$R[DR] \leftarrow R[SA] + R[SB]$
Subtract	SUB	0000101	$R[DR] \leftarrow R[SA] + \overline{R[SB]} + 1$
Set if Less Than	SLT	1100101	If $R[SA] < R[SB]$ then $R[DR] = 1$
AND	AND	0001000	$R[DR] \leftarrow R[SA] \wedge R[SB]$
OR	OR	0001010	$R[DR] \leftarrow R[SA] \vee R[SB]$
Exclusive-OR	XOR	0001100	$R[DR] \leftarrow R[SA] \oplus R[SB]$
Store	ST	0000001	$M[R[SA]] \leftarrow R[SB]$
Load	LD	0100001	$R[DR] \leftarrow M[R[SA]]$
Add Immediate	ADI	0100010	$R[DR] \leftarrow R[SA] + \text{se } IM$
Subtract Immediate	SBI	0100101	$R[DR] \leftarrow R[SA] + (\text{se } IM) + 1$
Complement	NOT	0101110	$R[DR] \leftarrow \overline{R[SA]}$
AND Immediate	ANI	0101000	$R[DR] \leftarrow R[SA] \wedge (0 \parallel IM)$
OR Immediate	ORI	0101010	$R[DR] \leftarrow R[SA] \vee (0 \parallel IM)$
Exclusive-OR Immediate	XRI	0101100	$R[DR] \leftarrow R[SA] \oplus (0 \parallel IM)$
Add Immediate Unsigned	AIU	1100010	$R[DR] \leftarrow R[SA] + (0 \parallel IM)$
Subtract Immediate Unsigned	SIU	1100101	$R[DR] \leftarrow R[SA] + (\overline{0 \parallel IM}) + 1$
Move	MOV	1000010	$R[DR] \leftarrow R[SA]$
Logical Left Shift by SH Bits	LSL	0110000	$R[DR] \leftarrow \text{lsl } R[SA] \text{ by } SH$
Logical Right Shift by SH Bits	LSR	0110001	$R[DR] \leftarrow \text{lsr } R[SA] \text{ by } SH$
Jump Register	JMR	1100001	$PC \leftarrow R[SA]$
Branch on Zero	BZ	0100000	If $R[SA] = 0$, then $PC \leftarrow PC + 1 + \text{se } IM$
Branch on Nonzero	BNZ	1100000	If $R[SA] \neq 0$, then $PC \leftarrow PC + 1 + \text{se } IM$
Jump	JMP	1000100	$PC \leftarrow PC + 1 + \text{se } IM$
Jump and Link	JML	1000000	$PC \leftarrow PC + 1 + \text{se } IM, R[DR] \leftarrow PC + 1$

Table 10-19 RISC Instruction Operations

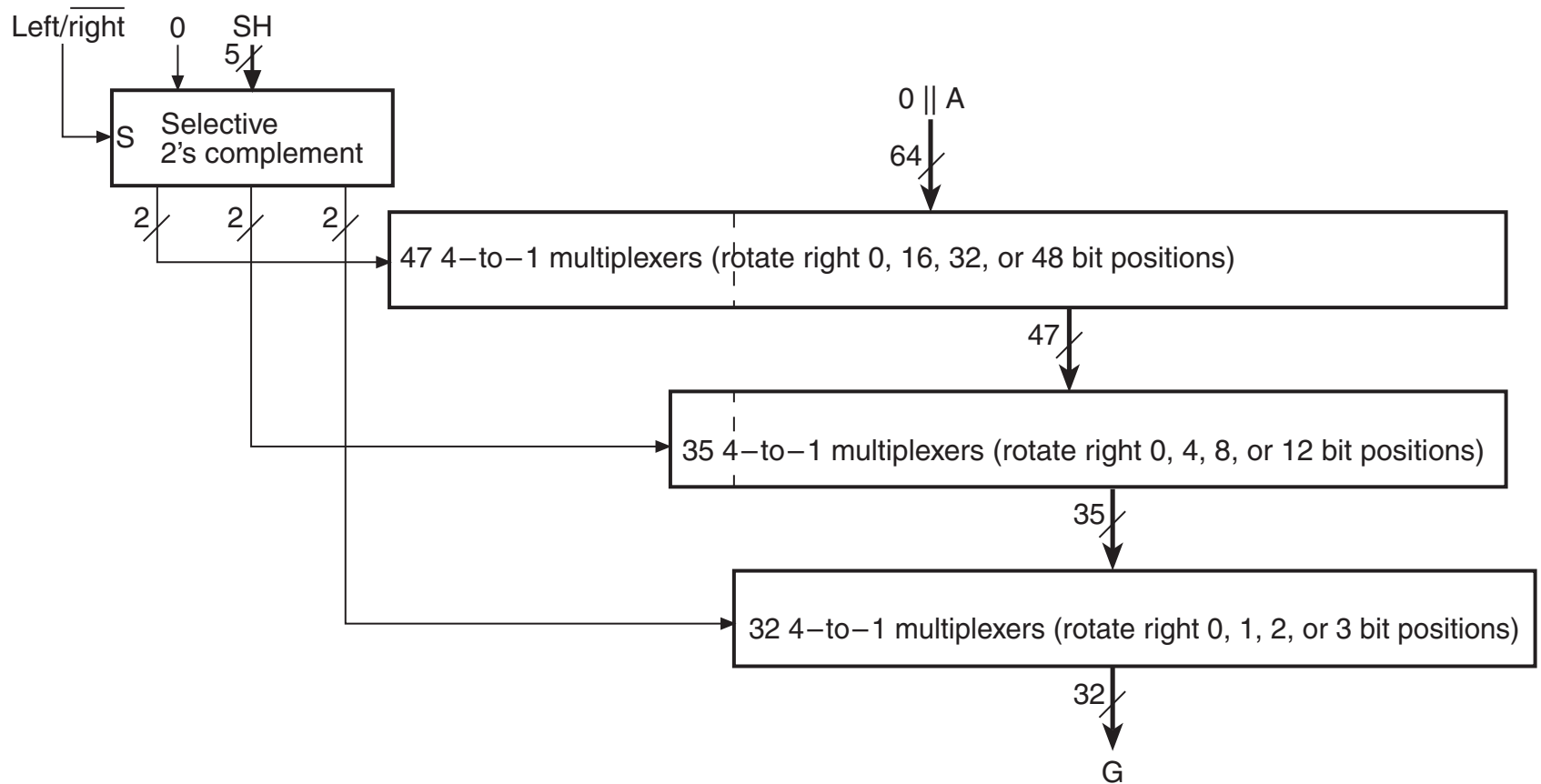


Fig. 10-14 32-bit Barrel Shifter

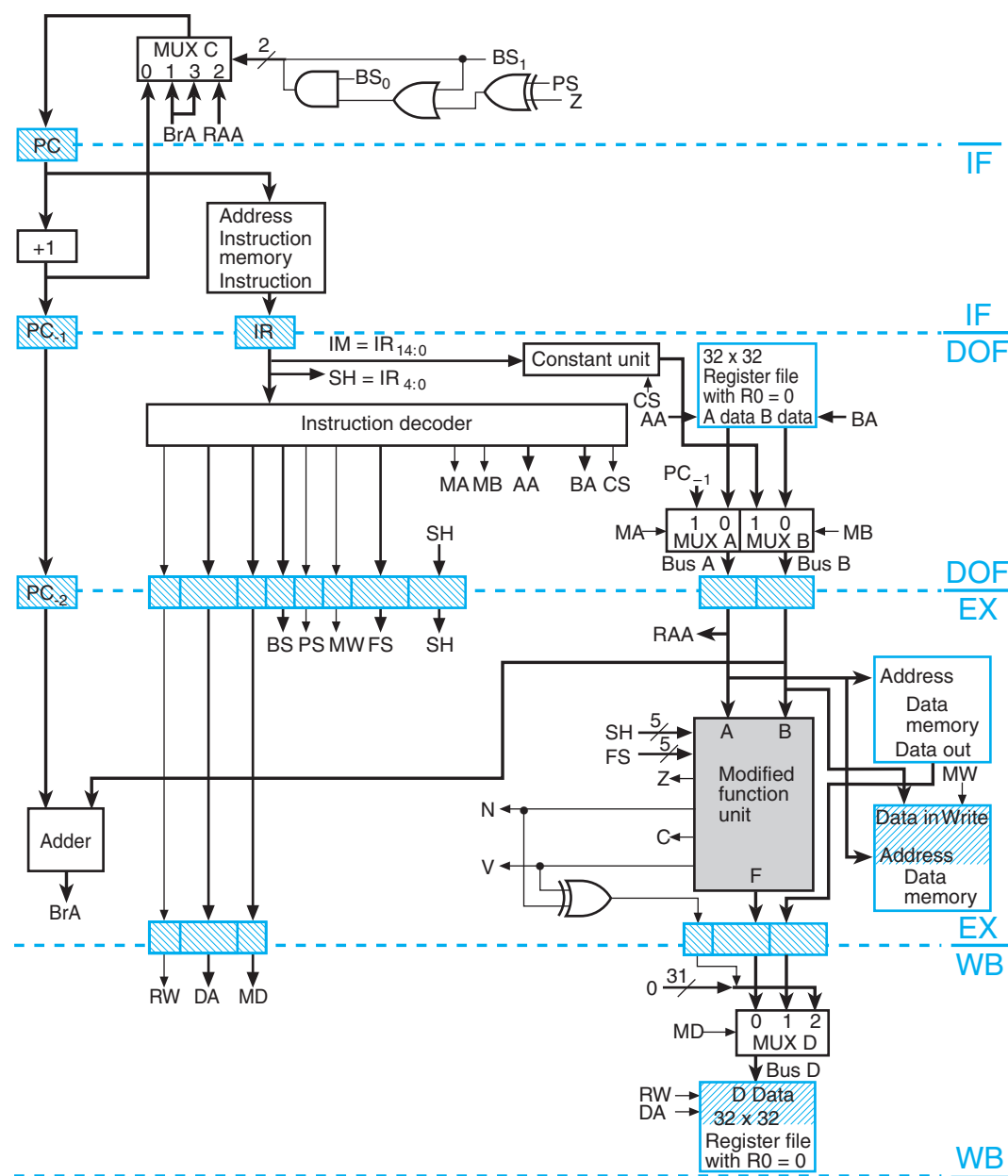


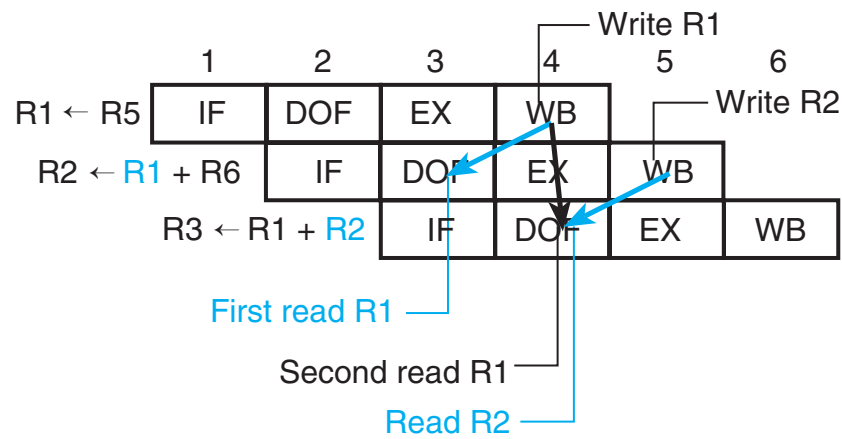
Fig. 10-15 Pipelined RISC CPU

TABLE 10-20
Control Words for Instructions

Sym- bolic Notation	Action	Op Code	Control Word Values								
			RW	MD	BS	PS	MW	FS	MB	MA	CS
NOP	None	0000000	0	—	00	—	0	—	—	—	—
ADD	$R[DR] \leftarrow R[SA] + R[SB]$	0000010	1	00	00	—	0	00010	0	0	—
SUB	$R[DR] \leftarrow R[SA] + \overline{R[SB]} + 1$	0000101	1	00	00	—	0	00101	0	0	—
SLT	If $R[SA] < R[SB]$ then $R[DR] = 1$	1100101	1	10	00	—	0	00101	0	0	—
AND	$R[DR] \leftarrow R[SA] \wedge R[SB]$	0001000	1	00	00	—	0	01000	0	0	—
OR	$R[DR] \leftarrow R[SA] \vee R[SB]$	0001010	1	00	00	—	0	01010	0	0	—
XOR	$R[DR] \leftarrow R[SA] \oplus R[SB]$	0001100	1	00	00	—	0	01100	0	0	—
ST	$M[R[SA]] \leftarrow R[SB]$	0000001	0	00	00	—	1	—	0	0	—
LD	$R[DR] \leftarrow M[R[SA]]$	0100001	1	01	00	—	0	—	—	0	—
ADI	$R[DR] \leftarrow R[SA] + \text{se } IM$	0100010	1	00	00	—	0	00010	1	0	1
SBI	$R[DR] \leftarrow R[SA] + (\text{se } IM) + 1$	0100101	1	00	00	—	0	00101	1	0	1
NOT	$R[DR] \leftarrow \overline{R[SA]}$	0101110	1	00	00	—	0	01110	—	0	—
ANI	$R[DR] \leftarrow R[SA] \wedge (0 \parallel IM)$	0101000	1	00	00	—	0	01000	1	0	0
ORI	$R[DR] \leftarrow R[SA] \vee (0 \parallel IM)$	0101010	1	00	00	—	0	01010	1	0	0
XRI	$R[DR] \leftarrow R[SA] \oplus (0 \parallel IM)$	0101100	1	00	00	—	0	01100	1	0	0
AIU	$R[DR] \leftarrow R[SA] + (0 \parallel IM)$	1100010	1	00	00	—	0	00010	1	0	0
SIU	$R[DR] \leftarrow R[SA] + (0 \parallel IM) + 1$	1100101	1	00	00	—	0	00101	1	0	0
MOV	$R[DR] \leftarrow R[SA]$	1000000	1	00	00	—	0	00000	—	0	—
LSL	$R[DR] \leftarrow \text{lsl } R[SA] \text{ by } SH$	0110000	1	00	00	—	0	10100	—	0	—
LSR	$R[DR] \leftarrow \text{lsr } R[SA] \text{ by } SH$	0110001	1	00	00	—	0	11000	—	0	—
JMR	$PC \leftarrow R[SA]$	1100001	0	—	10	—	0	00000	—	—	—
BZ	If $R[SA] = 0$, then $PC \leftarrow PC + 1 + \text{se } IM$	0100000	0	—	01	0	0	00000	1	0	1
BNZ	If $R[SA] \neq 0$, then $PC \leftarrow PC + 1 + \text{se } IM$	1100000	0	—	01	1	0	00000	1	0	1
JMP	$PC \leftarrow PC + 1 + \text{se } IM$	1000100	0	—	11	—	0	—	1	—	1
JML	$PC \leftarrow PC + 1 + \text{se } IM, R[DR] \leftarrow PC +$	0000111	1	00	11	—	0	00111	1	1	1

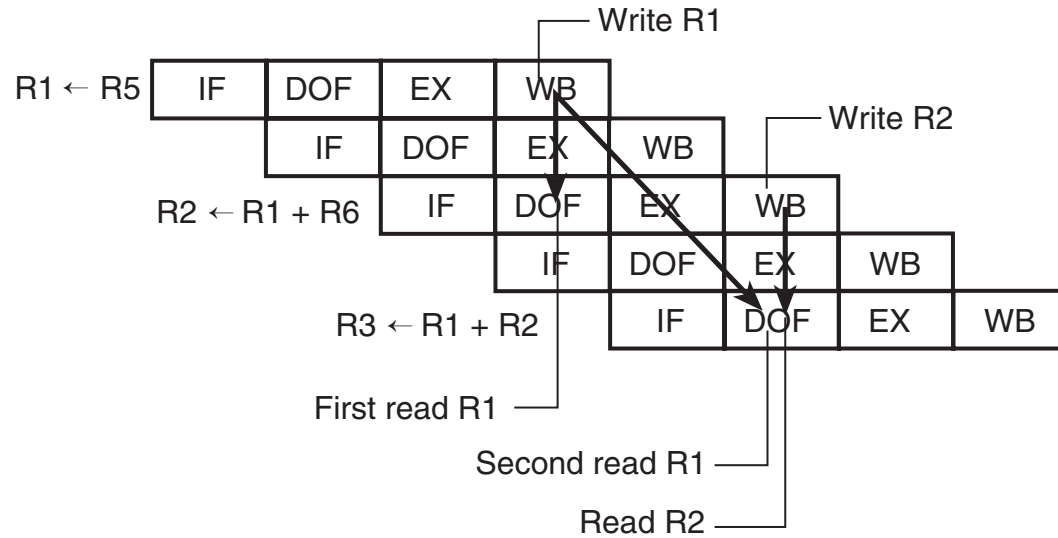
Table 10-20 Control Words for Instructions

MOV R1, R5
 ADD R2, R1, R6
 ADD R3, R1, R2



(a) The data hazard problem

MOV R1, R5
 NOP
 ADD R2, R1, R6
 NOP
 ADD R3, R1, R2



(b) A program-based solution

Fig. 10-16 Example of Data Hazard

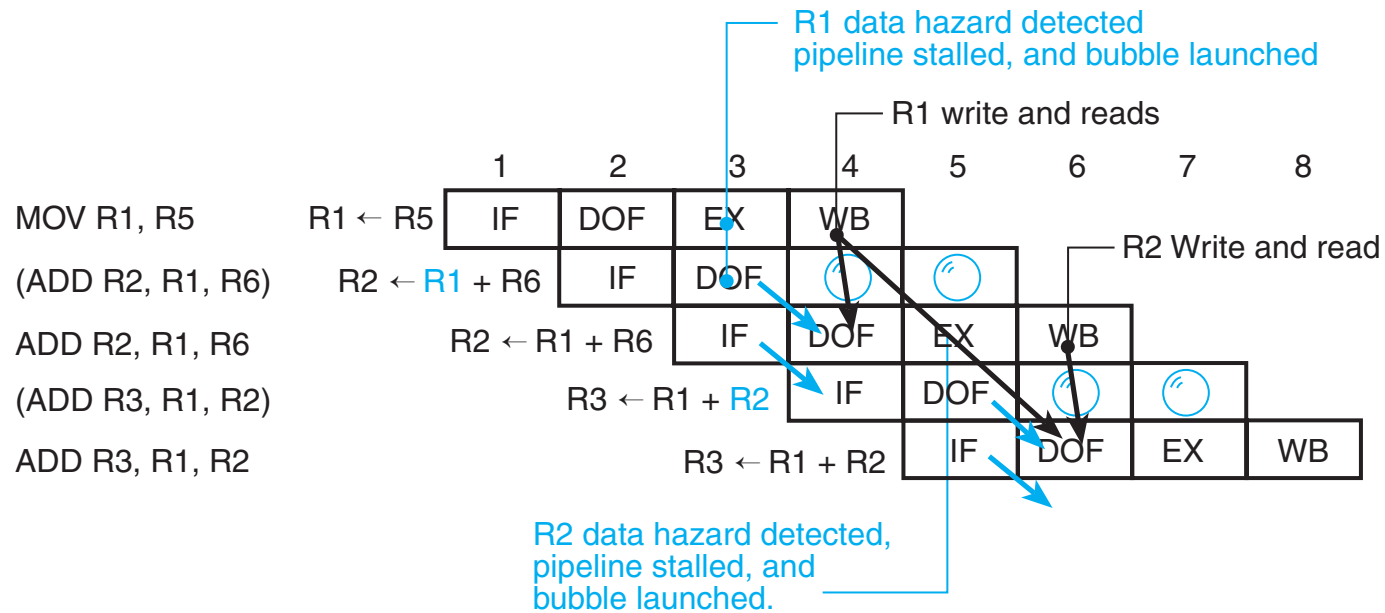
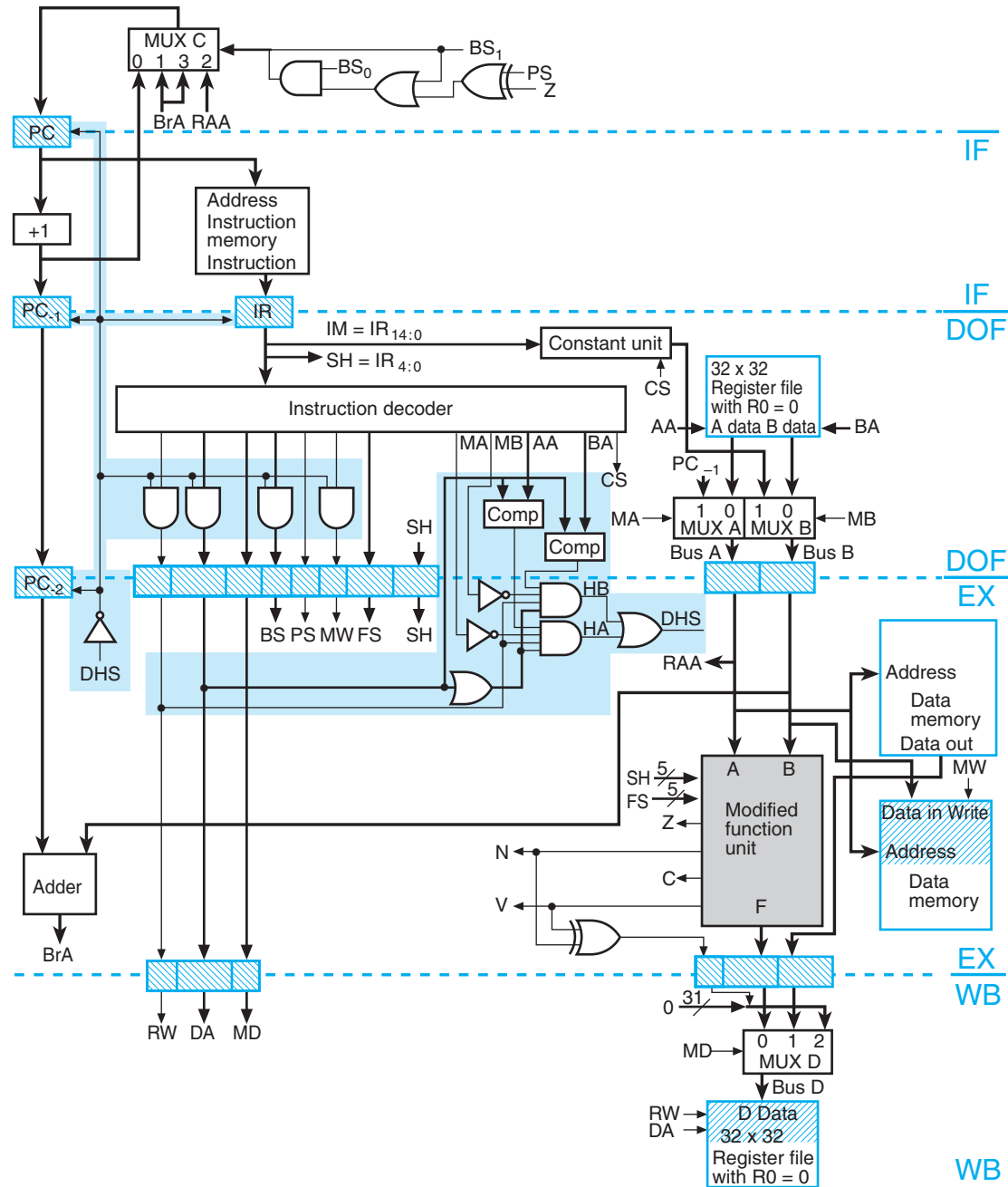
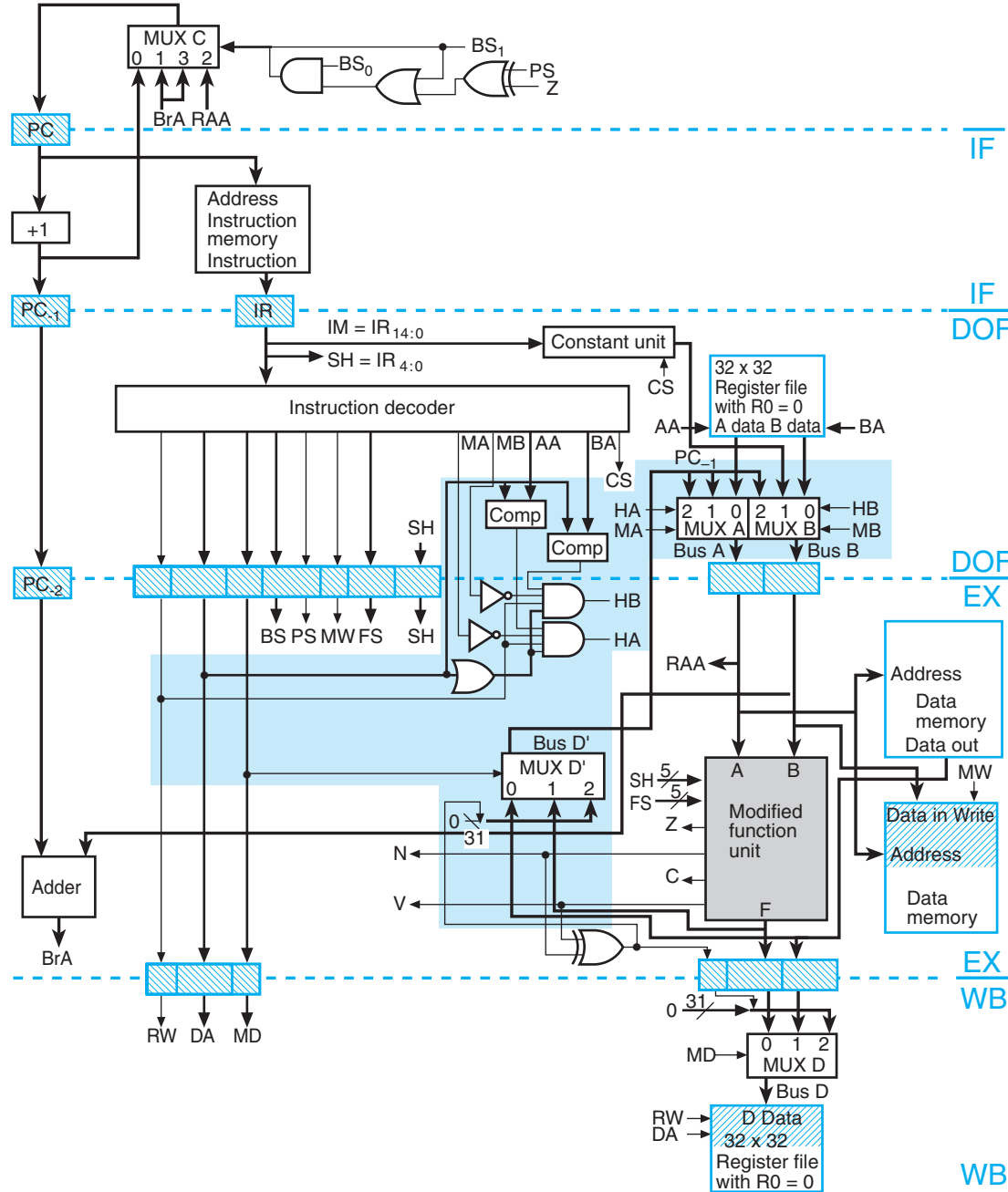


Fig. 10-17 Example of Data Hazard Stall





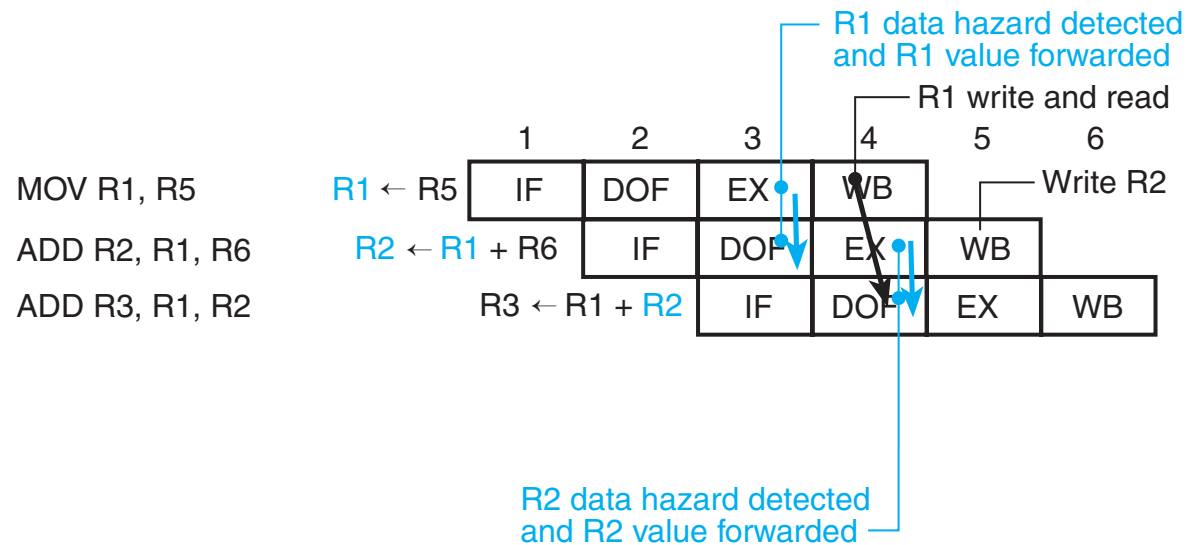
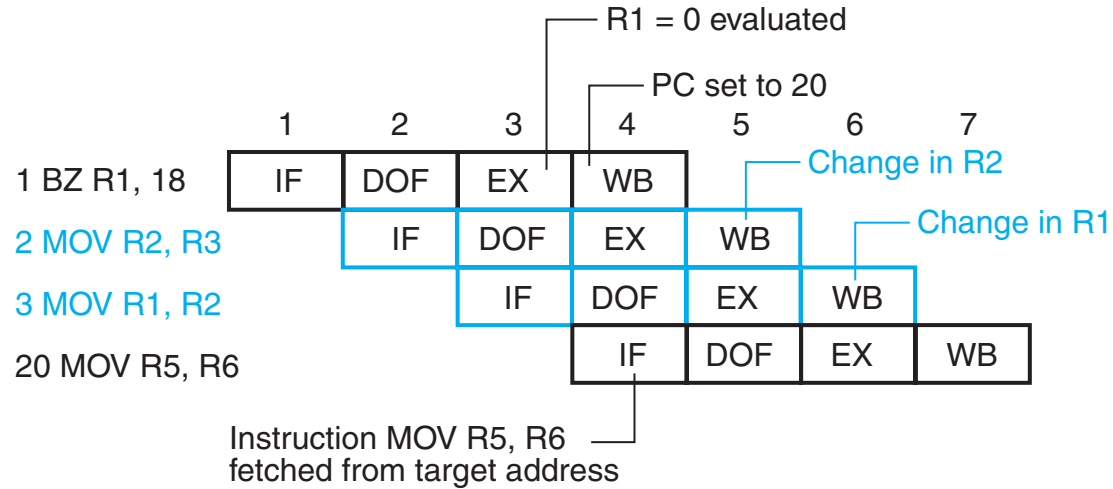
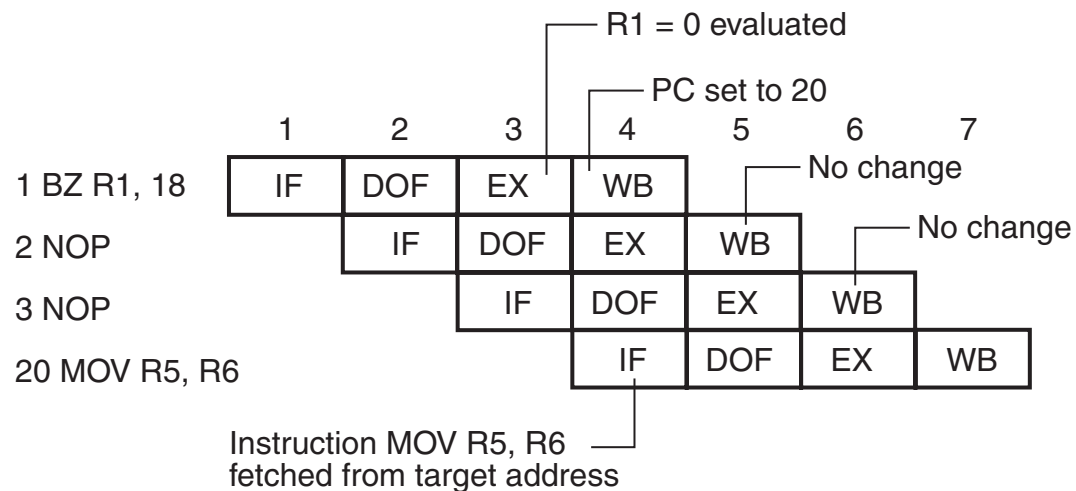


Fig. 10-20 Example of Data Forwarding



(a) Branch Hazard Problem



(b) Program-based Solution

Fig. 10-21 Example of Control Hazard

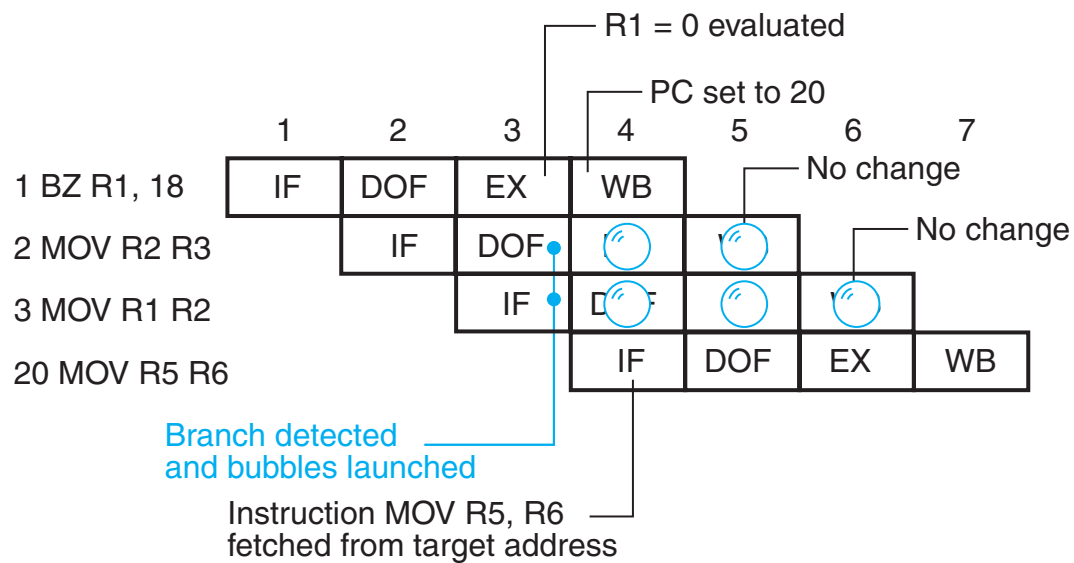
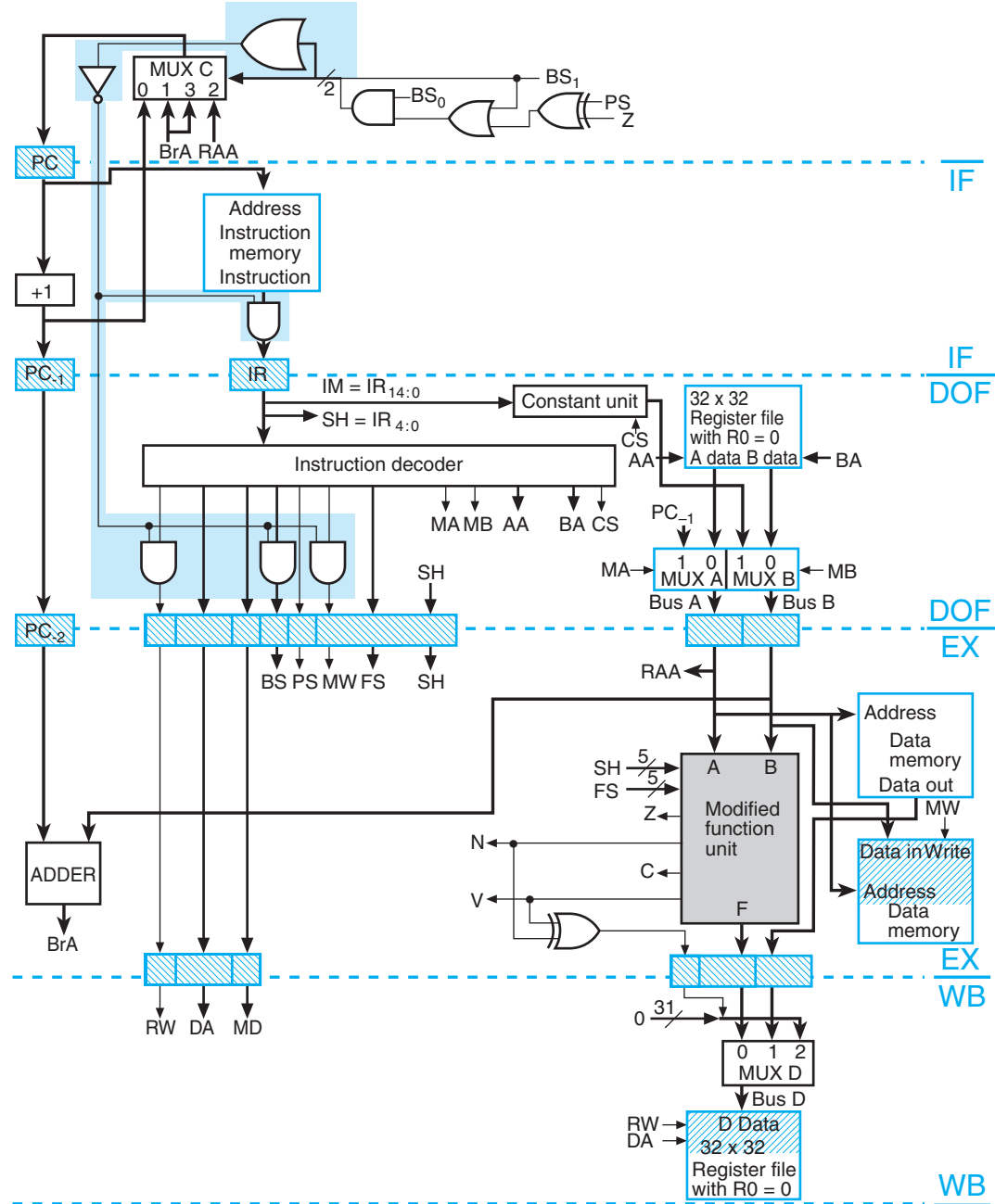
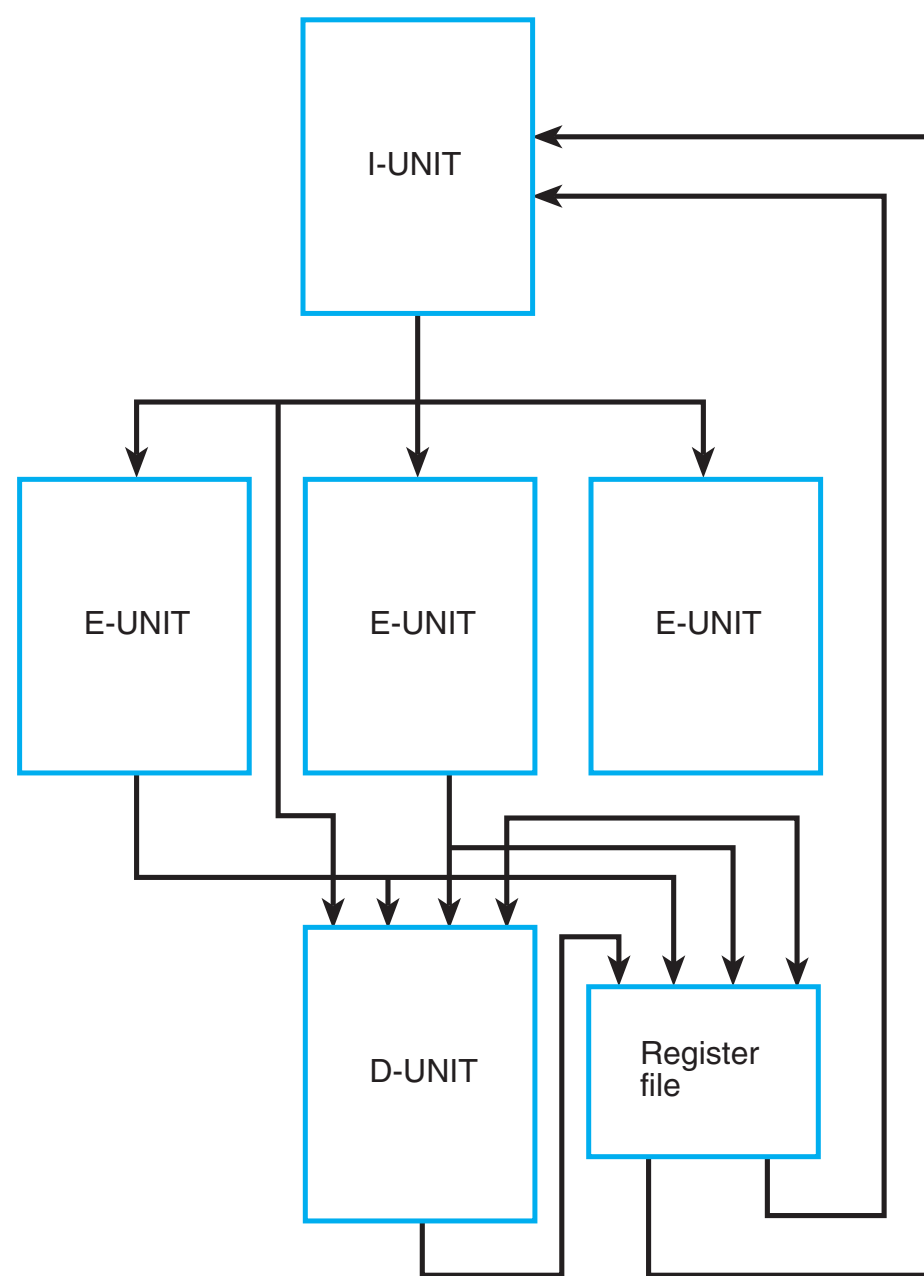


Fig. 10-22 Example of Branch Prediction with Branch Taken





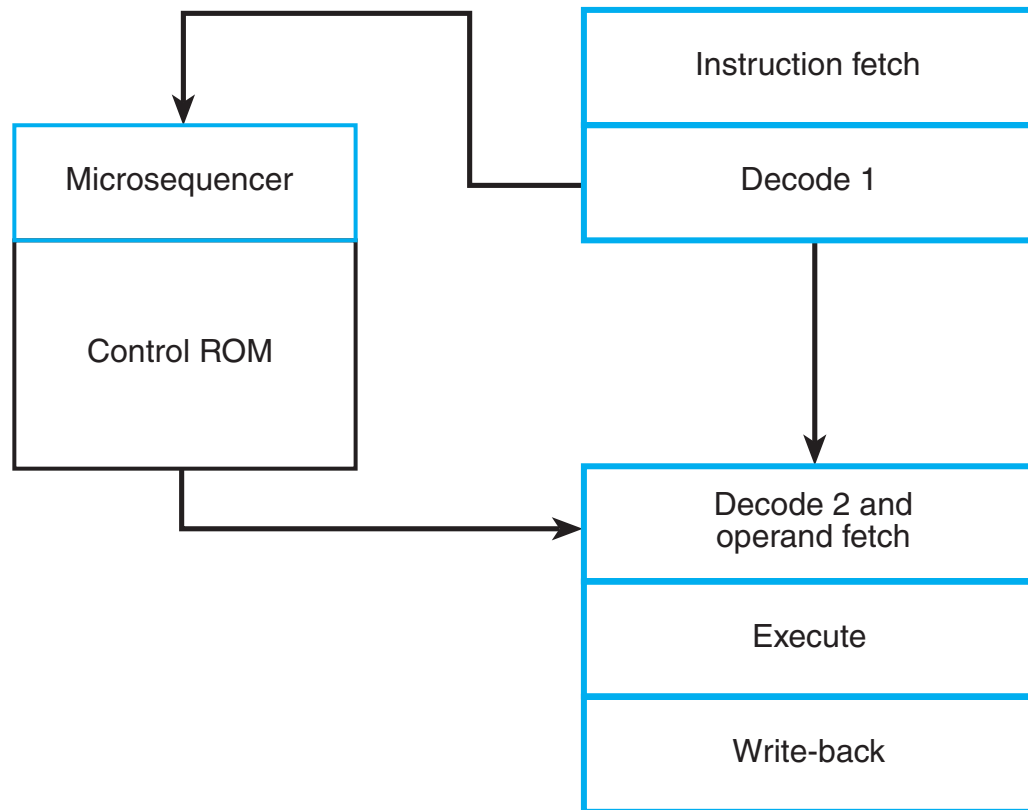


Fig. 10-25 Combined CISC-RISC Organization

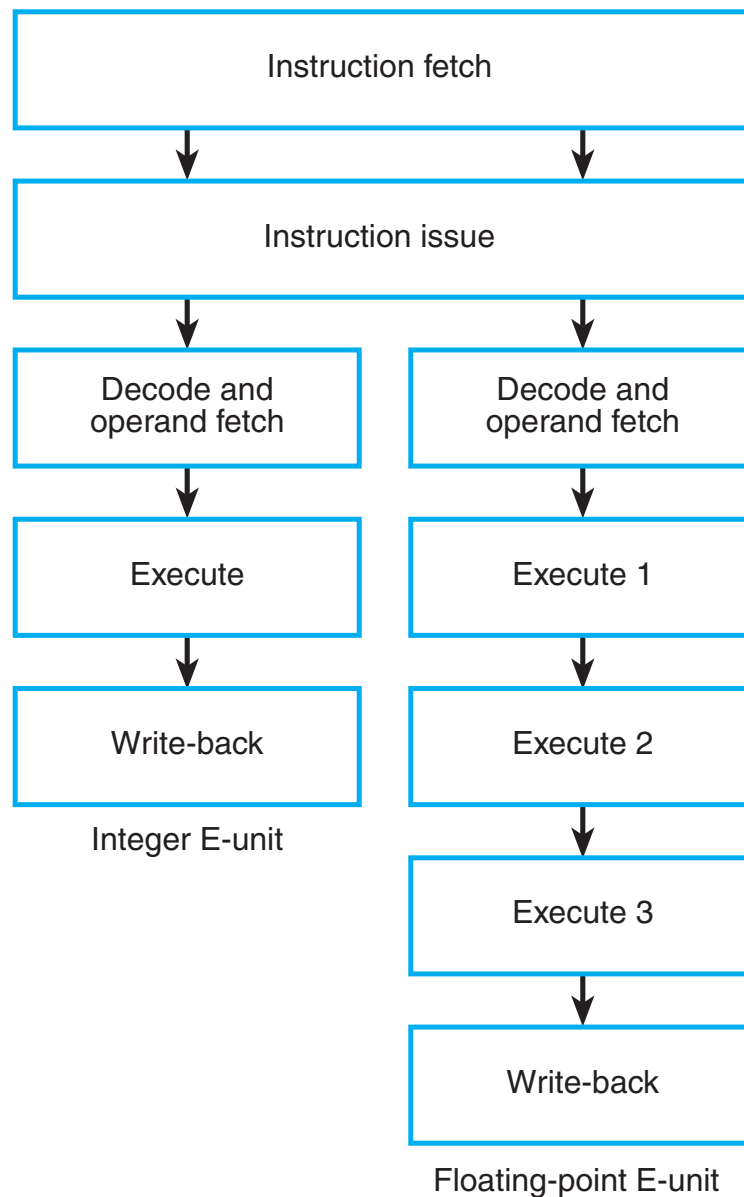


Fig. 10-26 Superscalar Organization