CISC Instruction Operations

Instruction	Mnemonic	Opcode	Status Effect
Zero-operand Instructions			
No operation	NOP	000000	None
Push registers	PSHR	000001	None
Pop registers	POPR	000010	None
Move string	MVS	000011	None
Return from procedure	RET	000100	None
Return from interrupt	RTI	000101	From stack
Invalid	000110 through 001111		
One-operand Instructions			
Push	PUSH	010000	None
Pop	POP	010001	None
Increment	INC	010010	ZCNV
Decrement	DEC	010011	ZCNV
Negate	NEG	010100	ZCNV
Complement	COM	010101	ZN
Logical shift right	SHR	010110	ZC
Logical shift left	SHL	010111	ZC
Arithmetic shift right	SHRA	011000	ZCNV
Arithmetic shift left	SHLA	011001	ZCNV
Rotate right	ROR	011010	ZC
Rotate left	ROL	011011	ZC
Rotate right with carry	RORC	011100	ZC
Rotate left with carry	ROLC	011101	ZC
Invalid	011110 through 011111		

Instruction	Mnemonic	Opcode	Status Effect
Two-operand Instructions			
Move	MOVE	100000	None
Exchange	XCH	100001	None
Add	ADD	100010	ZCNV
Add with carry	ADDC	101011	ZCNV
Subtract	SUB	100100	ZCNV
Subtract with borrow	SUBB	100101	ZCNV
Multiply	MUL	100110	ZCNV
Divide	DIV	100111	ZCNV
Compare	CMP	101000	ZCNV
AND	AND	101001	ZN
OR	OR	101010	ZN
Exclusive-OR	XOR	101011	ZN
Invalid	101100 through 101111		
Branch Instructions			
Jump	JMP	110000	None
Call procedure	CALL	110001	None
Branch if zero	BZ	111000	None
Branch if no zero	BNZ	111001	None
Branch if carry	BC	111010	None
Branch if no carry	BNC	111011	None
Branch if negative	BN	111100	None
Branch if no negative	BNN	111101	None
Branch if overflow	BV	111110	None
Branch if no overflow	BNV	111111	None
Invalid	110010 through 110111		

Addressing Modes

		Register Transfer Description of Operands		
MODE	**Address Mode**	**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	$M[M[R[DST] + W]]$	$M[M[\mathrm{R}[SRC] + W]]$, $R[DST]$	$M[M[R[DST] + W]]$, $R[SRC]$
110	Relative	$M[PC + W]$	$M[PC + W]$, $R[DST]$	$M[PC + W]$, $R[SRC]$
111	Relative Indirect	$M[M[PC + W]]$	$M[M[PC + W]]$, $R[DST]$	$M[M[PC + W]]$, $R[SRC]$

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	ICST	10			$F = A + B$	00010
$R3$	00011	Register B	MCST	11			$F = A + B + 1$	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						$F = A \oplus B$	01100
$R11$	01011						$F = \overline{A}$	01110
$R12$ (SA)	01100						$F =$ lsl A	10000
$R13$ (SD)	01101						$F =$ lsr A	10001
$R14$ (DA)	01110						$F =$ asl A	10010
$R15$ (DD)	01111						$F =$ asr A	10011
$R[DST]$	10XXX*						$F =$ rol A	10100
$R[SRC]$	11XXX*						$F =$ ror A	10101
							$F =$ rolc A	10110
							$F =$ rorc A	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.

Control Word Information for MC and LS

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

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 *PSR* bit	BZ	0001
			Negative *PSR* bit	BN	0010
			Carry *PSR* bit	BC	0011
			Overflow *PSR* bit	BV	0100
			Enable-interrupt *PS*R bit	EI	0101
			Zero *MSTS* bit	Bz	0110
			Negative *MSTS* bit	Bn	0111
			Carry *MSTS* bit	Bc	1000
			Overflow *MSTS* bit	Bv	1001
			Interrupt signal *INTS*	BI	1010

T-256 Control Word Encoding for Instruction Decoder

MM		MR	
Select	**Code**	**Region**	**Code**
OPCODE(5:4)	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

T-257 Control Information for Miscellaneous Operations for Format A and B Microinstructions

Operations	Symbolic Notation	MO 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	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

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	$OPCODE(5{:}4) \parallel 00_2 = 0000_2$	0EX
IF1	00	000	$OPCODE(5{:}4) \parallel 00_2 = 0100_2$	1OF
IF1	00	000	$OPCODE(5{:}4) \parallel 00_2 = 1000_2$	2OF
IF1	00	000	$OPCODE(5{:}4) \parallel 00_2 = 1100_2$	BAF
In 1OF Microroutine	00	001	$OPCODE(5{:}4) \parallel 00_2 = XXXX$	1EX
In 2OF Microroutine	00	010	$OPCODE(5{:}4) \parallel 00_2 = XXXX$	2EX
In BAF Microroutine	00	011	$OPCODE(5{:}4) \parallel 00_2 = XXXX$	BEX
0EX	01	000	$OPCODE(3{:}0) = O_3O_2O_1O_0$	16 0-Op EX Microinstruction Addresses
1EX	01	001	$OPCODE(3{:}0) = O_3O_2O_1O_0$	16 1-Op EX Microinstruction Addresses
2EX	01	010	$OPCODE(3{:}0) = O_3O_2O_1O_0$	16 2-Op Ex Microinstruction Addresses
BEX	01	011	$OPCODE(3{:}0) = O_3O_2O_1O_0$	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	$MODE \parallel S = M_2M_1M_0X$	8 1-Op OF Microinstruction Addresses
2OF	10	010	$MODE \parallel S = M_2M_1M_0S$	16 2-Op OF Microinstruction Addresses
BAF	10	011	$MODE \parallel S = M_2M_1M_0X$	8 Br-Op Microinstruction Addresses
XCH2	10	100	$MODE \parallel S = XXX1$	XCH3
XCH2	10	100	$MODE \parallel S = 0000$	XCH3
XCH2	10	100	$MODE \parallel S = 1XX0$	XCH4
XCH2	10	100	$MODE \parallel S = X1X0$	XCH4
XCH2	10	100	$MODE \parallel S = XX10$	XCH4
WB0	11	000	$MODE \parallel S = XXX0$	WB1
WB0	11	000	$MODE \parallel S = 0001$	WB1
WB0	11	000	$MODE \parallel S = 1XX1$	WB2
WB0	11	000	$MODE \parallel S = X1X1$	WB2
WB0	11	000	$MODE \parallel S = XX11$	WB2
In WB microroutine	11	001	$MODE \parallel S = XXXX$	INT0
In INT microroutine	11	010	$MODE \parallel S = XXXX$	IF0

Instruction Fetch Microroutine

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

One-operand Fetch Microroutine

Sym Add	Register Transfer Description	MC	MM /LS	MR /PS	DSA /MS	SB	MA	MB	MD	FS /NA	MO
1OF	$CAR \leftarrow PLA[10001_2 \parallel MODE \parallel S]$	2	2	1	00	00	0	0	0	00	0
1RG	$DD \leftarrow R[DST], CAR \leftarrow 1\text{EX(ROM)}$	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	$DD \leftarrow M[DA], CAR \leftarrow 1\text{EX(ROM)}$	2	0	1	0F	0E	3	0	1	00	0
1IM	$DD \leftarrow M[PC], PC \leftarrow PC + 1,$ $CAR \leftarrow \text{EX1(PLA)}$	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	$DD \leftarrow M[DA], CAR \leftarrow 1\text{EX(ROM)}$	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	$DD \leftarrow M[DA], CAR \leftarrow 1\text{EX(ROM)}$	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	$DD \leftarrow M[DA], CAR \leftarrow 1\text{EX(ROM)}$	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	$DD \leftarrow M[DA], CAR \leftarrow 1\text{EX(ROM)}$	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	$DD \leftarrow M[DA], CAR \leftarrow 1\text{EX(ROM)}$	2	0	1	0F	0E	3	0	1	00	0

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	3	0	1	00	0
2DR2	$DD \leftarrow R[DST], CAR \leftarrow 2\mathrm{EX(ROM)}$	2	0	2	0F	10	3	0	0	00	0
2DRS0	$DA \leftarrow M[PC], PC \leftarrow PC + 1$	0	0	0	0E	00	1	0	1	00	0
2DRS1	$DD \leftarrow M[DA]$	0	0	0	0F	0E	3	0	1	00	0
2DRS2	$SD \leftarrow R[SRC], CAR \leftarrow 2\mathrm{EX(ROM)}$	2	0	2	0D	11	3	0	0	00	0

Example of Branch Address Fetch

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

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	$M[SP] \leftarrow R7, CAR \leftarrow \mathrm{INT0(ROM)}$	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	$SP \leftarrow SP + 1, CAR \leftarrow \mathrm{INT0(ROM)}$	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	$SP \leftarrow SP + 1, CAR \leftarrow \mathrm{INT0(ROM)}$	2	1	5	00	00	0	0	0	00	A

One-operand Execution Microroutine Examples

Sym Add	Register Transfer Description	MC	MM /LS	MR /PS	DSA /MS	SB	MA	MB	MD	FS /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	$DD \leftarrow DD + 1, CAR \leftarrow$ 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	$R9 \leftarrow 0$	0	0	0	09	00	3	0	0	00	0
SHR1	$R9 \leftarrow 0 \parallel SHAM$	0	0	0	09	00	0	2	0	02	F
SHR2	$z: CAR \leftarrow$ INT0	3	0	0	06	00	0	0	0	INT0	0
SHR3	$[DD \leftarrow 0 \parallel DD(15{:}1)]$	0	0	0	0F	00	0	0	0	11	0
SHR4	$R9 \leftarrow R9 - 1$	0	0	0	09	00	0	0	0	06	F
SHR5	$\overline{z}: CAR \leftarrow$ SHR3	3	0	1	06	00	0	0	0	SHR3	0
SHR6	$DD \leftarrow DD, (C)AR \leftarrow$ WB0(ROM)	2	1	4	0F	00	0	0	0	00	D

Two-operand Execution Microroutine Examples

Sym Add	Register Transfer Description	MC	MM /LS	MR /PS	DSA /MS	SB	MA	MB	MD	FS /NA	MO
2EX	$CAR \leftarrow ROM[01010_2 \,\|\|\, OPCODE(0{:}3)]$	2	1	2	00	00	0	0	0	00	0
MOVE	$DD \leftarrow SD, CAR \leftarrow$ WB0(ROM)	2	1	4	0F	0D	3	0	0	00	0
XCH0	$R9 \leftarrow SD$	0	0	0	09	0F	3	0	0	00	0
XCH1	$SD \leftarrow DD$	0	0	0	0E	0F	3	0	0	00	0
XCH2	$DD \leftarrow R9, CAR \leftarrow ROM[10100_2 \,\|\|\, MODE \,\|\|S]$	2	2	4	0F	09	3	0	0	00	0
XCH3	$R[SRC] \leftarrow SD, CAR \leftarrow$ WB0(ROM)	2	1	4	11	0D	3	0	0	00	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	0
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

Program Branch Microroutine Examples

Sym Add	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	$\overline{Z}$: $CAR \leftarrow$ BRA	3	0	1	1	—	—	—	—	BRA	0
BNZ1	$CAR \leftarrow$ INT0(ROM)	2	1	5	0E	00	0	0	0	00	0

Write Back Microroutine

Sym Add	Register Transfer Description	MC	MM /LS	MR /PS	DSA /MS	SB	MA	MB	MD	FS /NA	M 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	3	0	0	00	0
WB2	$M[DA] \leftarrow DD, CAR \leftarrow INT0(ROM)$	2	1	5	0E	0F	0	0	0	00	2

Interrupt-Handling Microroutine

Sym Add	Register Transfer Description	MC	MM /LS	MR /PS	DSA /MS	SB	MA	MB	MD	FS /NA	MO
INT0	$\overline{INTS}: CAR \leftarrow$ IF0	3	0	1	A	00	00	00	00	IF1	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

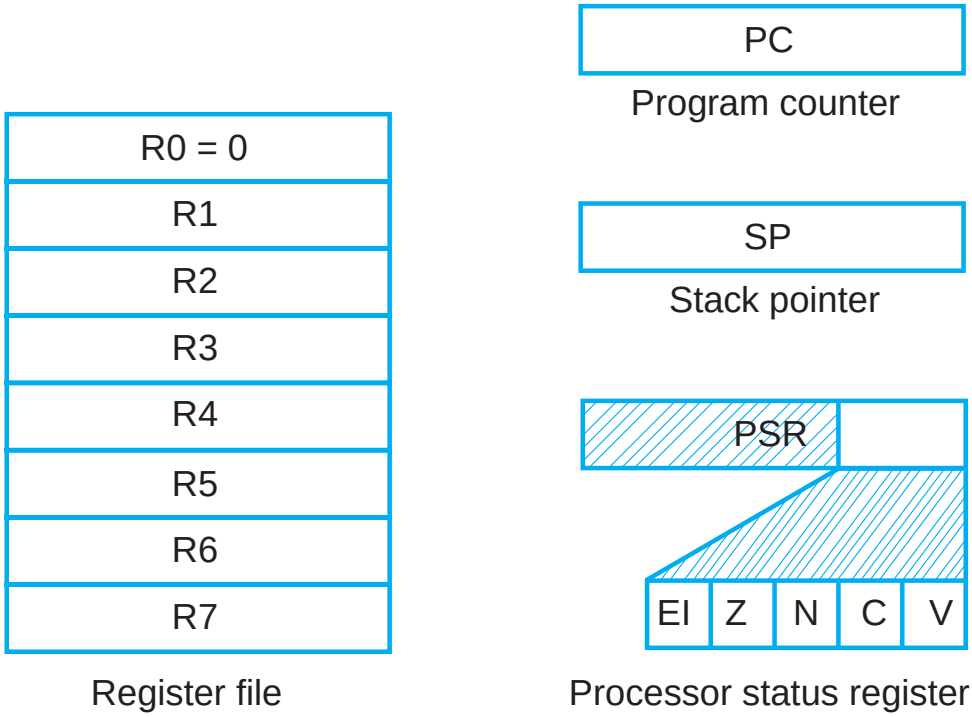
RISC Instruction Operations

Operation	Symbolic Notation	Op Code	Action
No Operation	NOP	0000000	None
Add	ADD	0000010	$R[DR] \leftarrow R[SA] + R[SB]$
Subtract	SUB	0000101	$R[DR] \leftarrow R[SA] + R[SB] + 1$
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	$R[DR] \leftarrow R[SA] + \overline{(\text{se } IM)} + 1$
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 $PC \leftarrow PC + \text{se } IM$
Branch on Nonzero	BNZ	1100000	If $R[SA] \neq 0$, then $PC \leftarrow PC + \text{se } IM$
Jump	JMP	1000100	$PC \leftarrow PC + \text{se } IM$
Jump and Link	JML	1000000	$PC \leftarrow PC + \text{se } IM, R[DR] \leftarrow PC + 1$

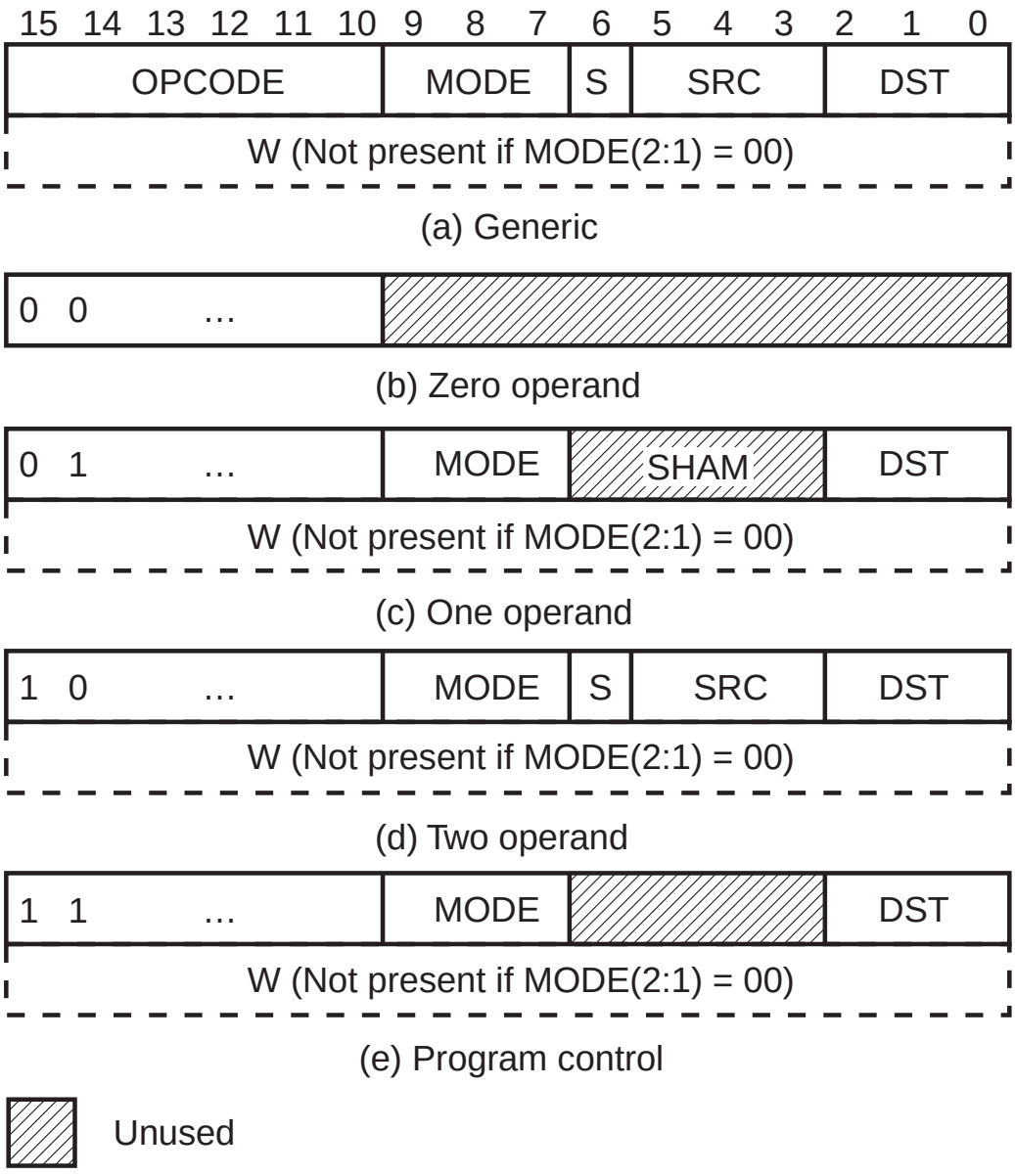
Control Words for Instructions

Symbolic 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	$R[DR] \leftarrow \mathrm{R}[SA] \vee \mathrm{R}[SB]$	0001010	1	00	00	—	0	01010	0	0	—
XOR	$R[DR] \leftarrow \mathrm{R}[SA] \oplus \mathrm{R}[SB]$	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	$R[DR] \leftarrow R[SA] + \mathrm{se}\, IM$	0100010	1	00	00	—	0	00010	1	0	1
SBI	$R[DR] \leftarrow R[SA] + \overline{(\mathrm{se}\, IM)} + 1$	0100101	1	00	00	—	0	00101	1	0	1
NOT	$R[DR] \leftarrow \overline{R[SA]}$	0101110	1	00	00	—	0	01110	—	0	—
ANI	$R[DR] \leftarrow R[SA] \wedge (0 \,\|\| IM)$	0101000	1	00	00	—	0	01000	1	0	0
ORI	$R[DR] \leftarrow R[SA] \vee (0 \,\|\| IM)$	0101010	1	00	00	—	0	01010	1	0	0
XRI	$R[DR] \leftarrow R[SA] \oplus (0 \,\|\| IM)$	0101100	1	00	00	—	0	01100	1	0	0
AIU	$R[DR] \leftarrow R[SA] + (0 \,\|\| IM)$	1100010	1	00	00	—	0	00010	1	0	0
SIU	$R[DR] \leftarrow R[SA] + \overline{(0 \,\|\| IM)} + 1$	1100101	1	00	00	—	0	00101	1	0	0
MOV	$R[DR] \leftarrow R[SA]$	1000000	1	00	00	—	0	00000	—	0	—
LSL	$R[DR] \leftarrow \mathrm{lsl}\, R[SA]$ by SH	0110000	1	00	00	—	0	10100	—	0	—
LSR	$R[DR] \leftarrow \mathrm{lsr}\, R[SA]$ 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 $PC \leftarrow PC + 1 + \mathrm{se}\, IM$	0100000	0	—	01	0	0	00000	1	0	1
BNZ	If $R[SA] \neq 0$, then $PC \leftarrow PC + 1 + \mathrm{se}\, IM$	1100000	0	—	01	1	0	00000	1	0	1
JMP	$PC \leftarrow PC + 1 + \mathrm{se}\, IM$	1000100	0	—	11	—	0	—	1	—	1
JML	$PC \leftarrow PC + 1 + \mathrm{se}\, IM, R[DR] \leftarrow PC +$	10000111	1	00	11	—	0	00111	1	1	1

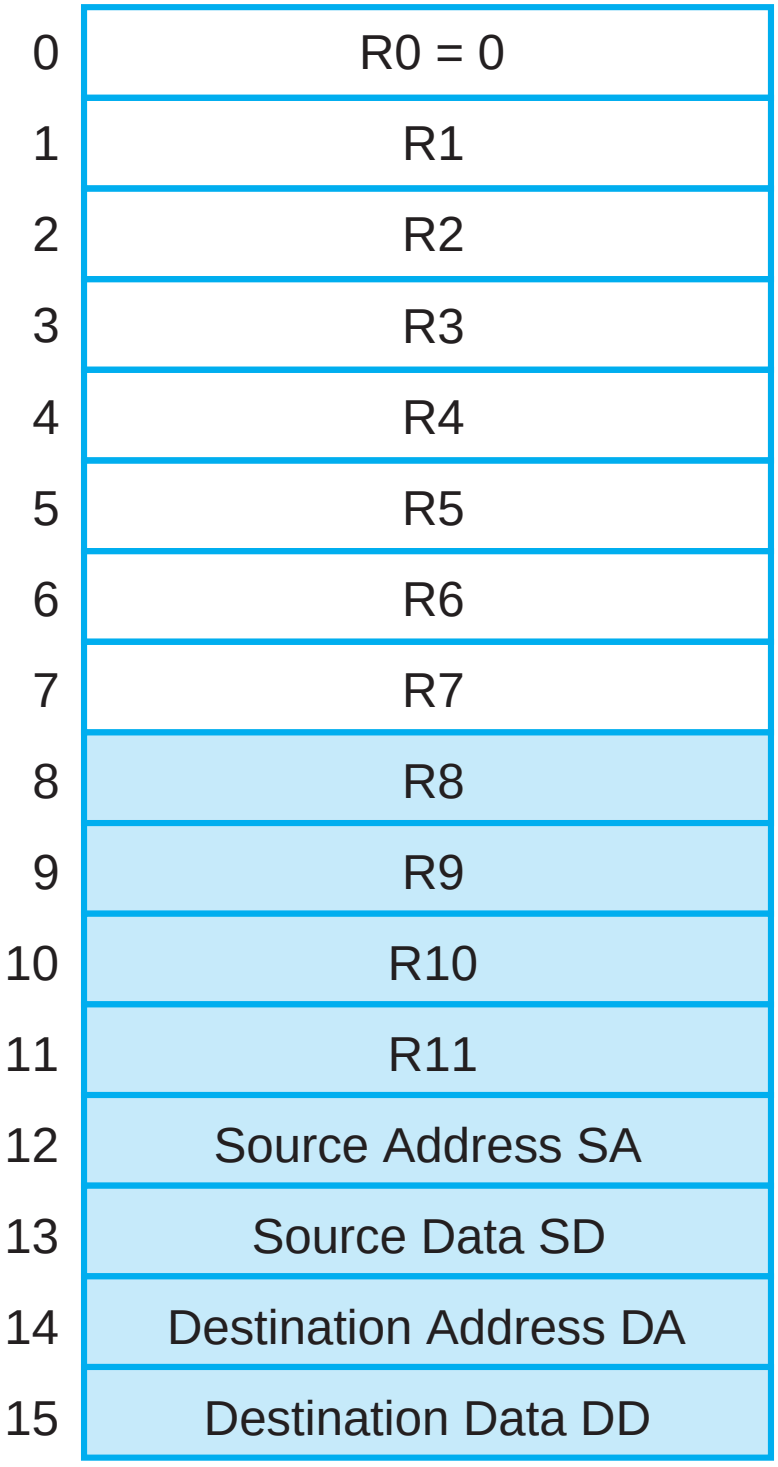
CPU Register Set Diagram for CISC



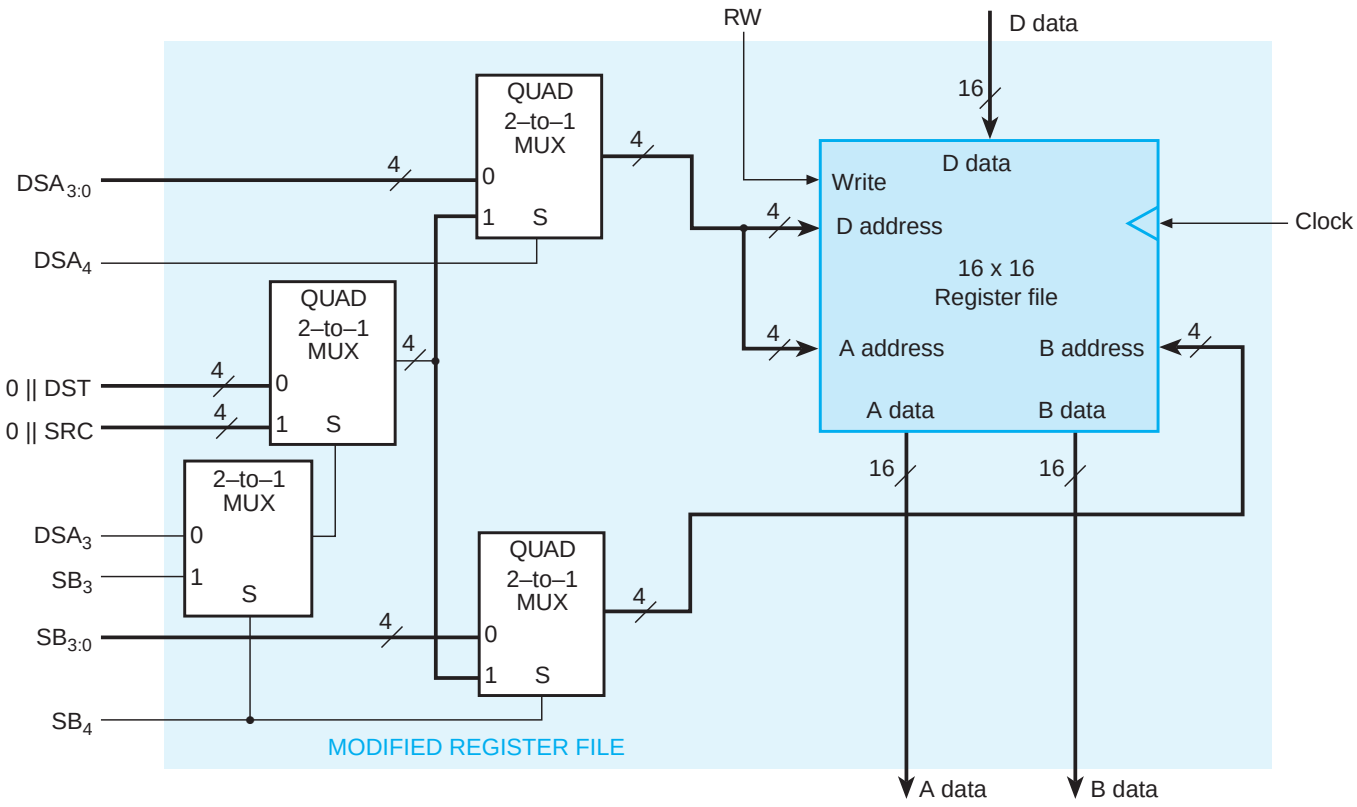
Instruction Formats



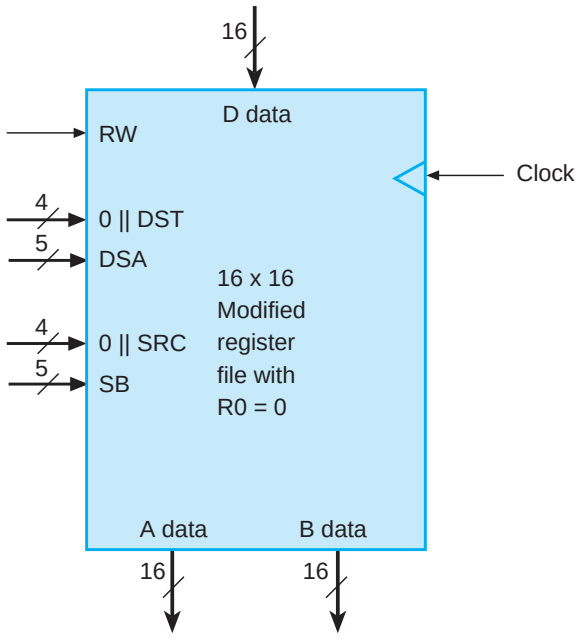
Register File Map



Modifications to Register File

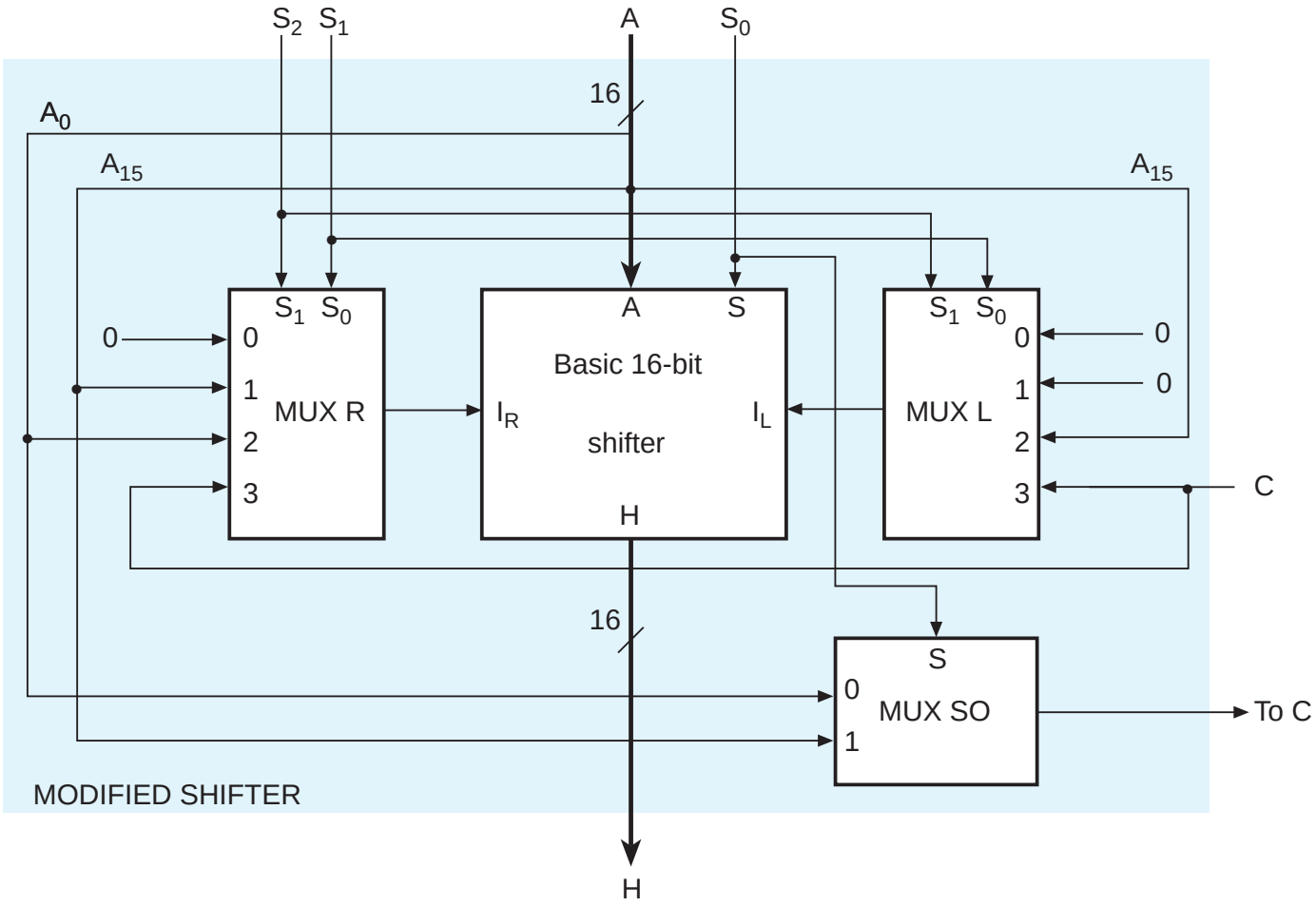


(a) Logic diagram

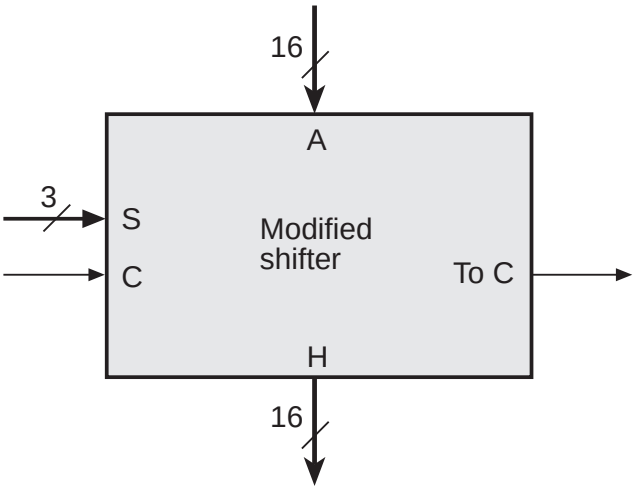


(b) Symbol

16-Bit Shifter and Symbol

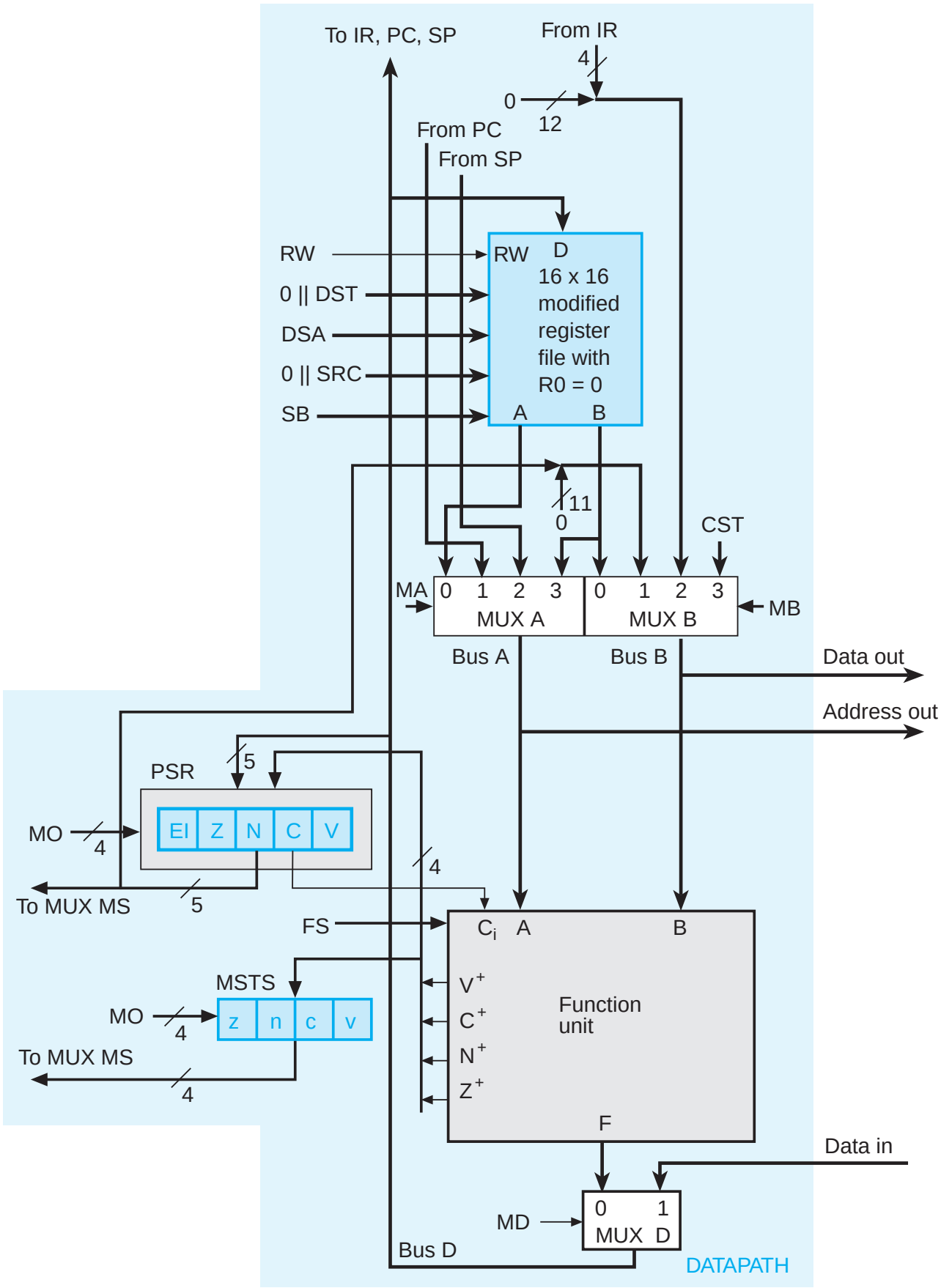


(a) Logic Diagram

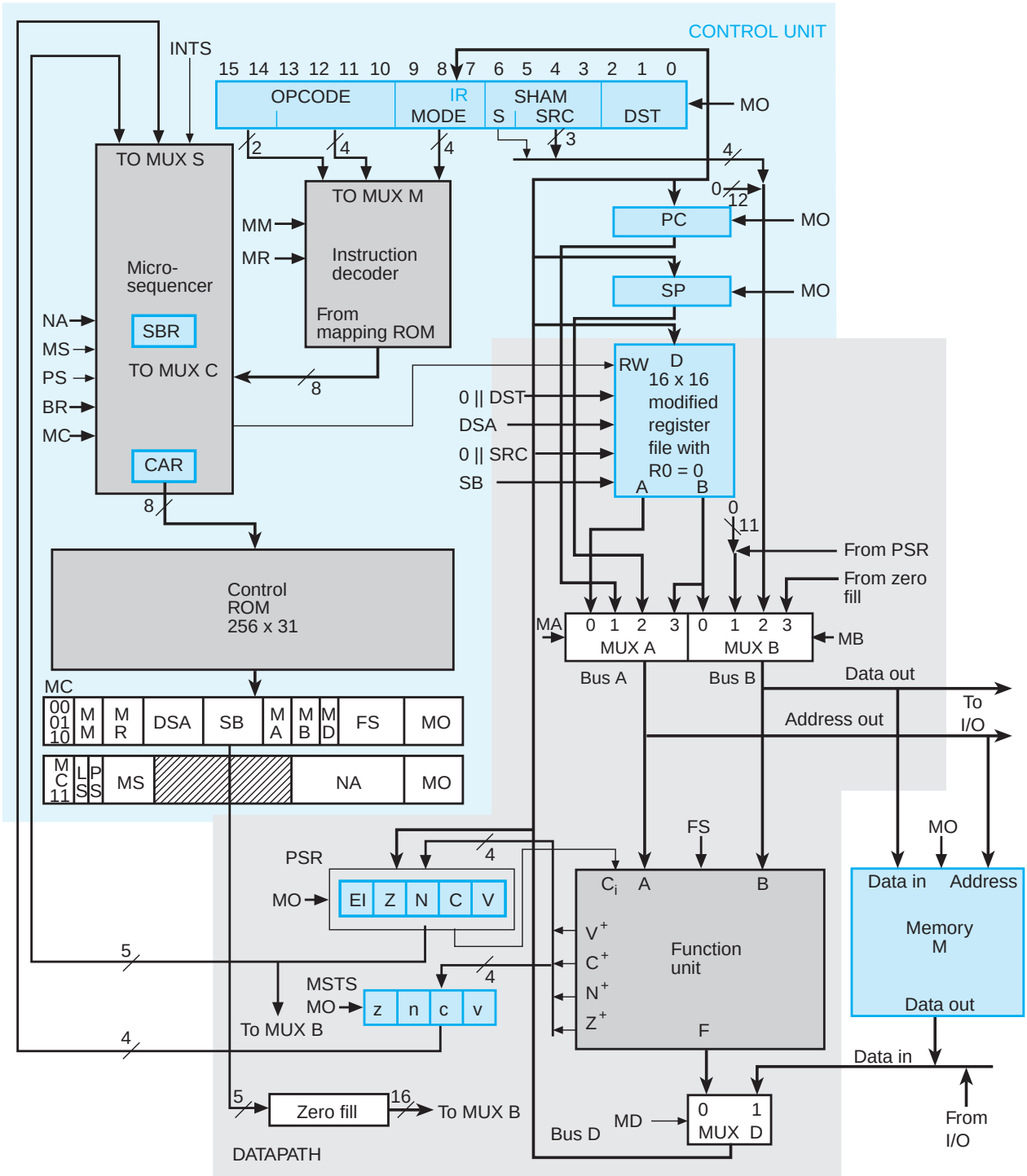


(b) Symbol

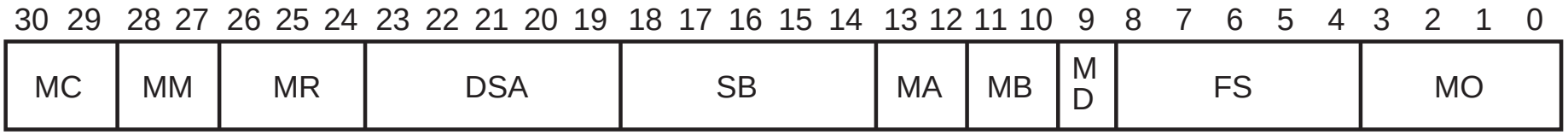
CISC Datapath



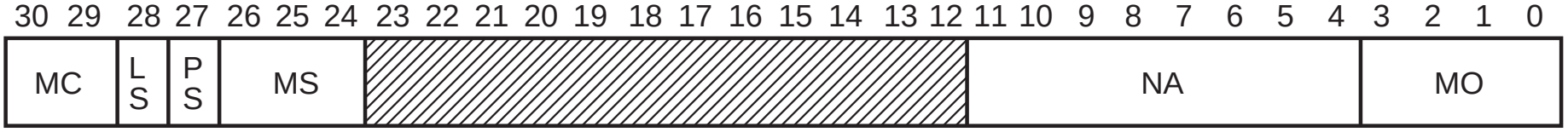
CISC CPU



Microinstruction Formats

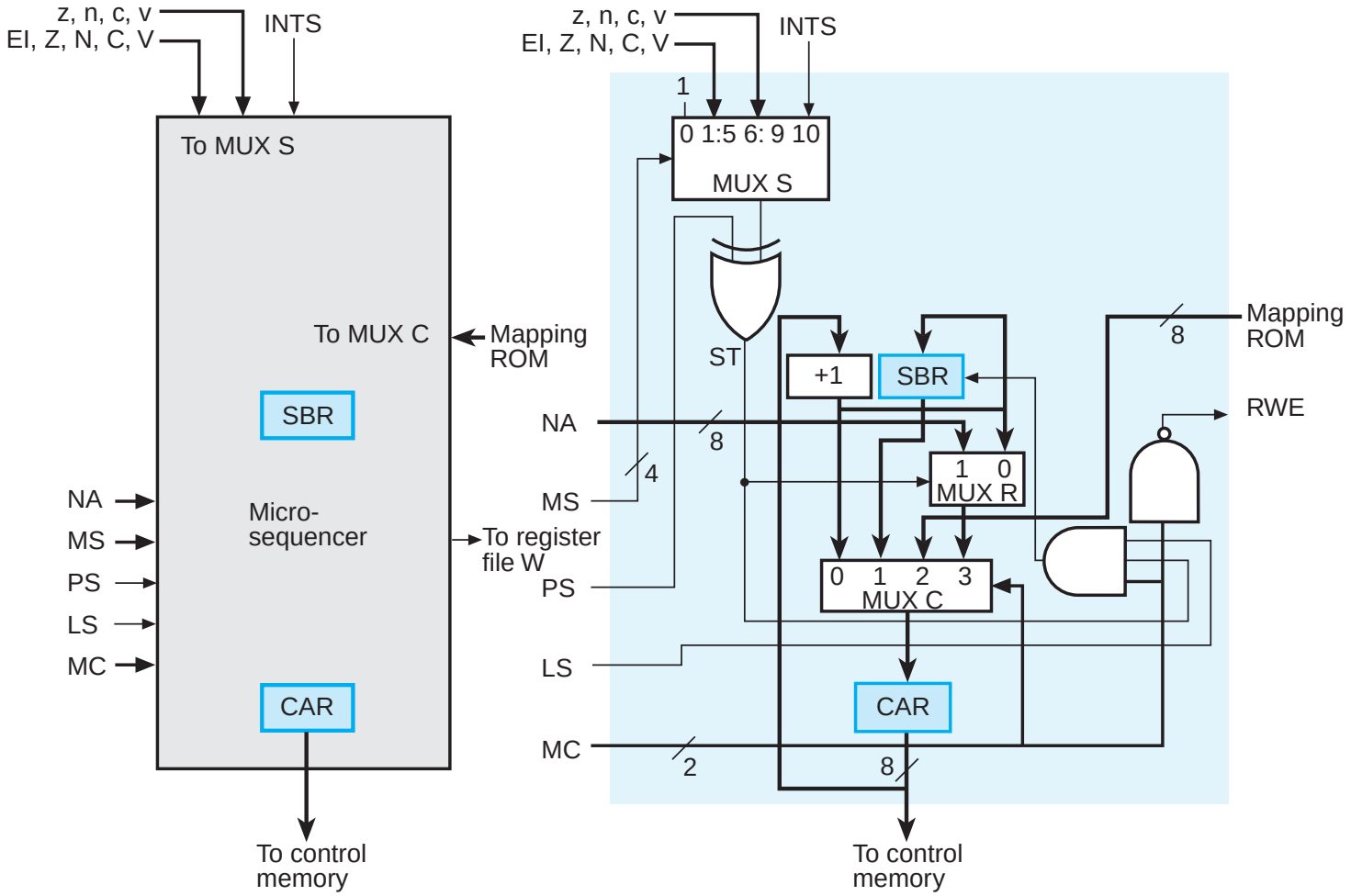


(a) Format A (MC = 00, 01, 11)

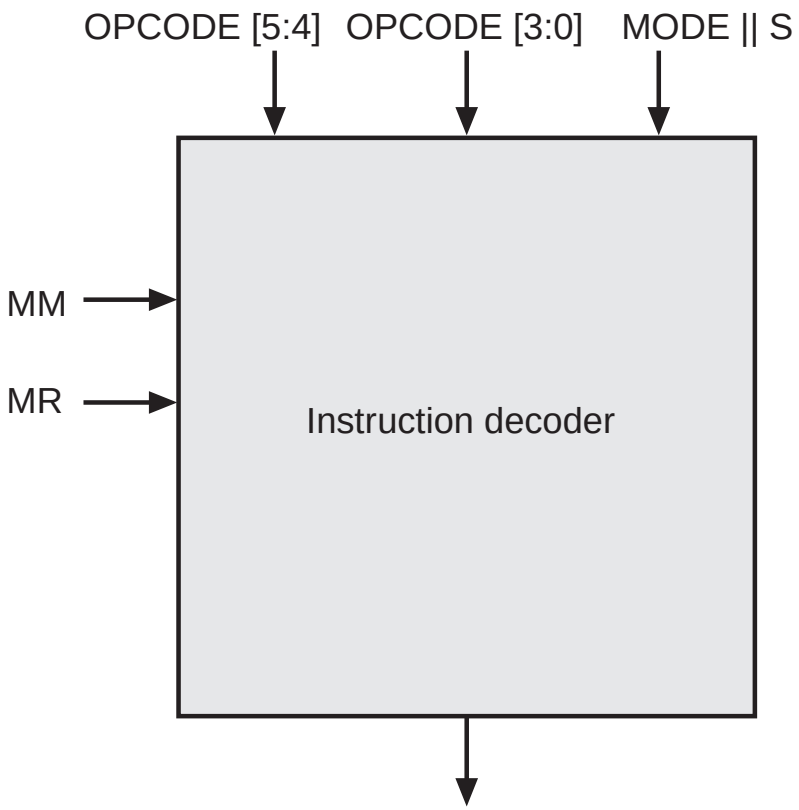


(b) Format B (MC = 11)

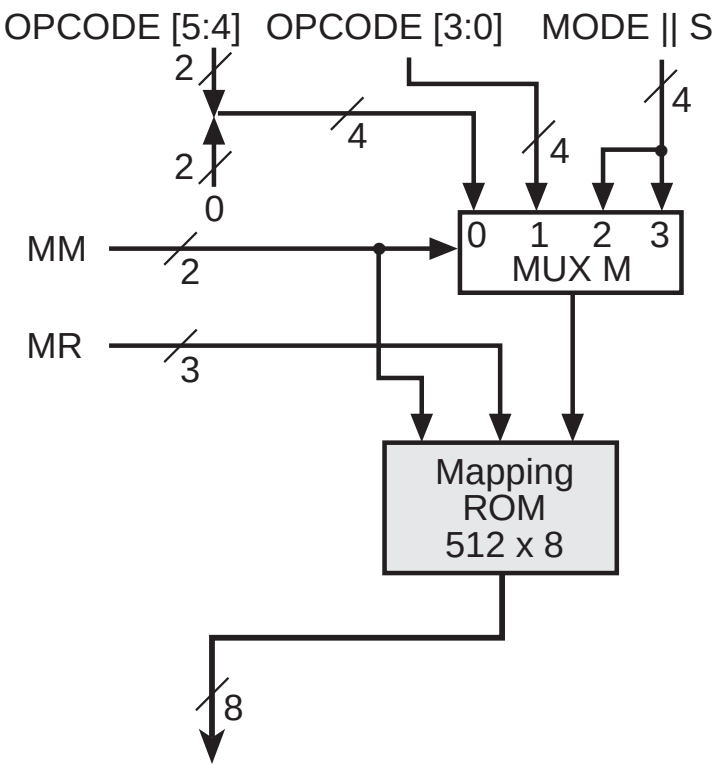
The Microsequencer



Instruction Decoder

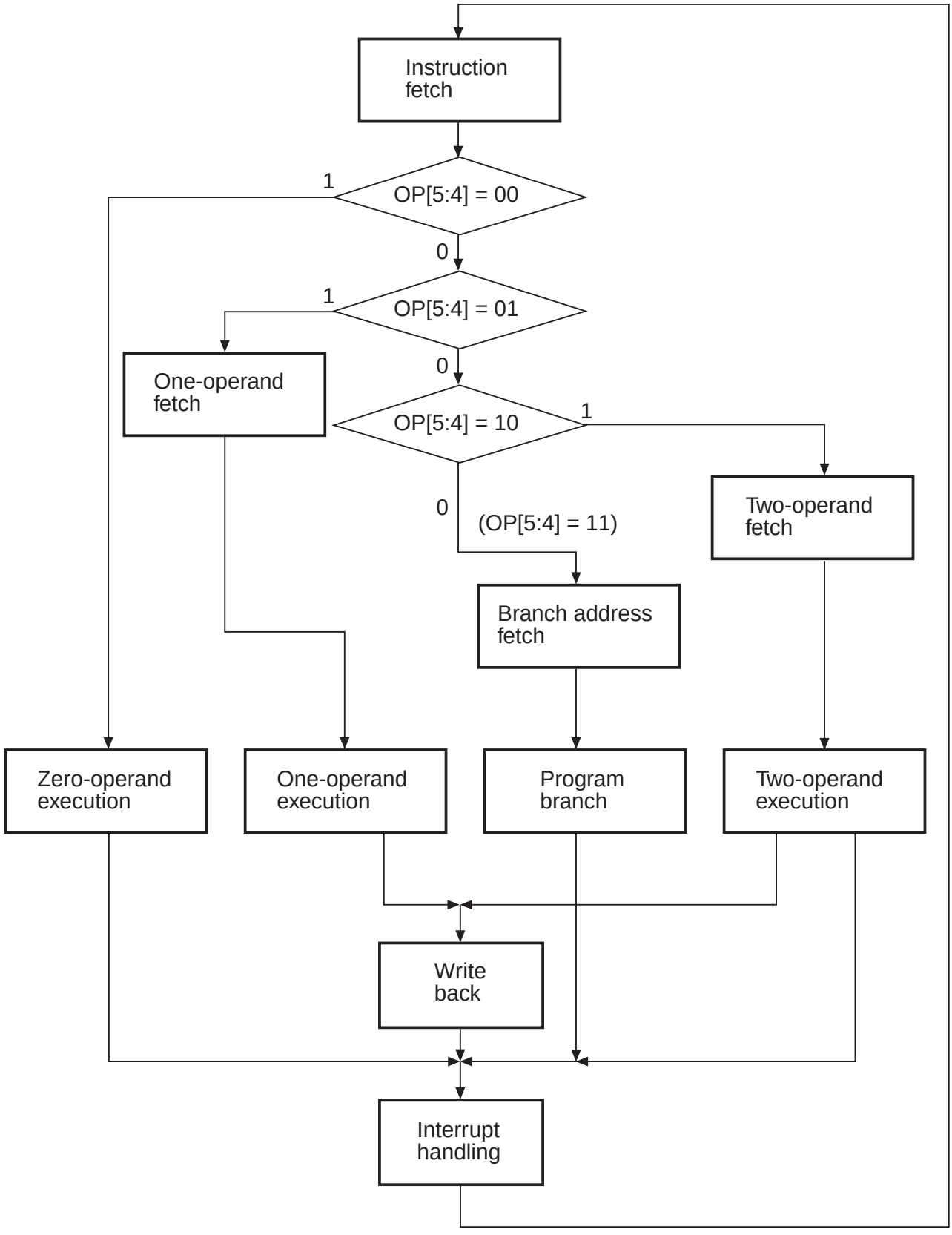


(a) Symbol

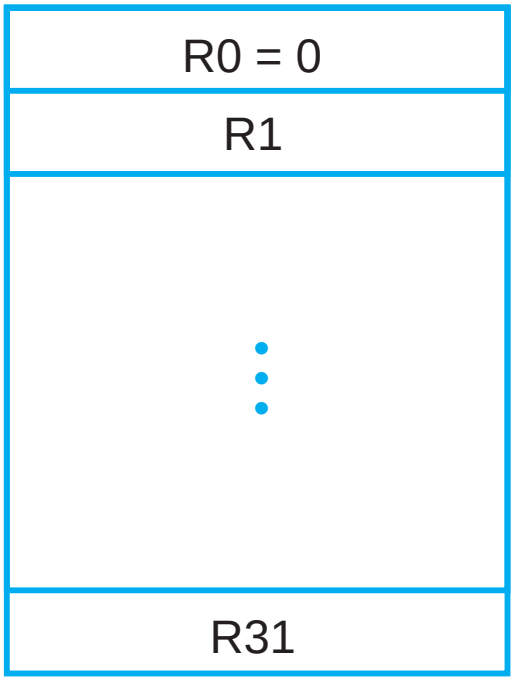


(b) Logic diagram

ASM-like Chart of Microroutines



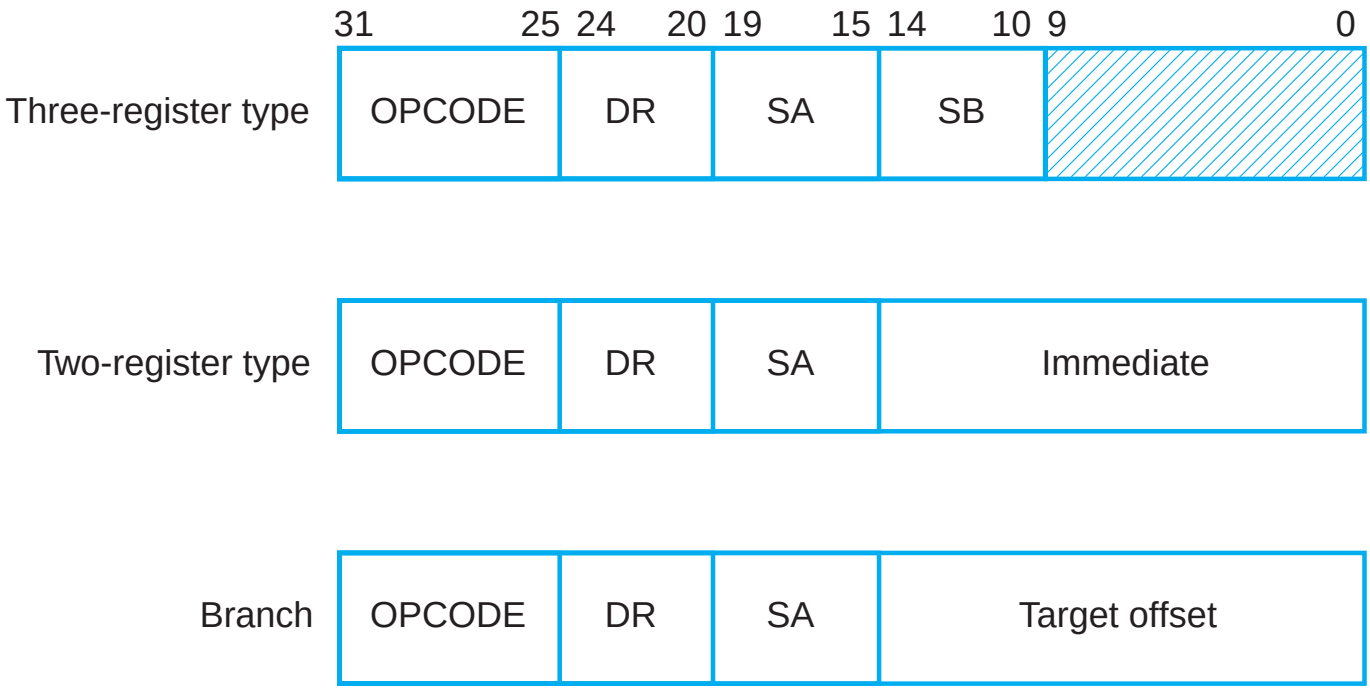
CPU Register Set Diagram for RISC



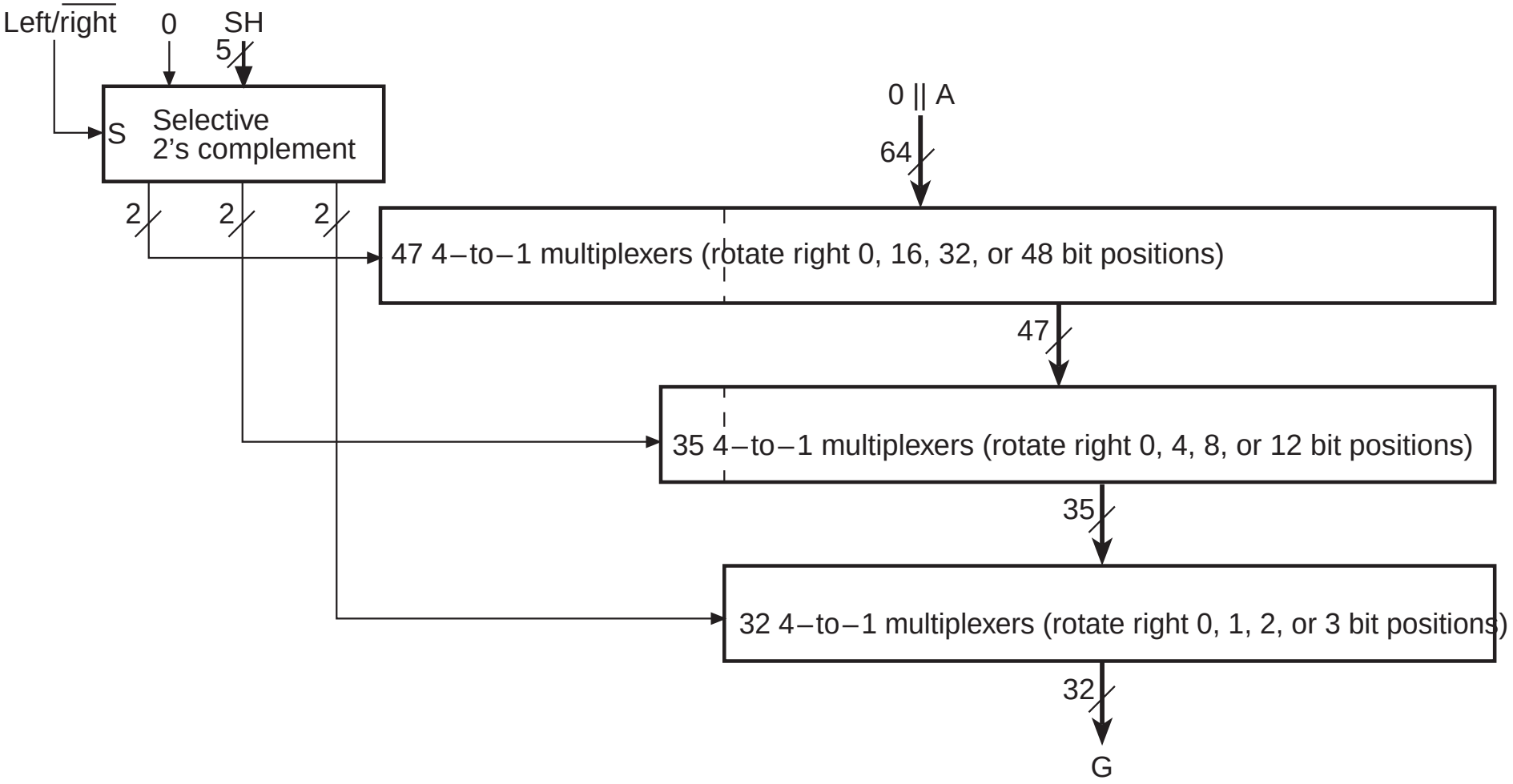
Register file

Program counter

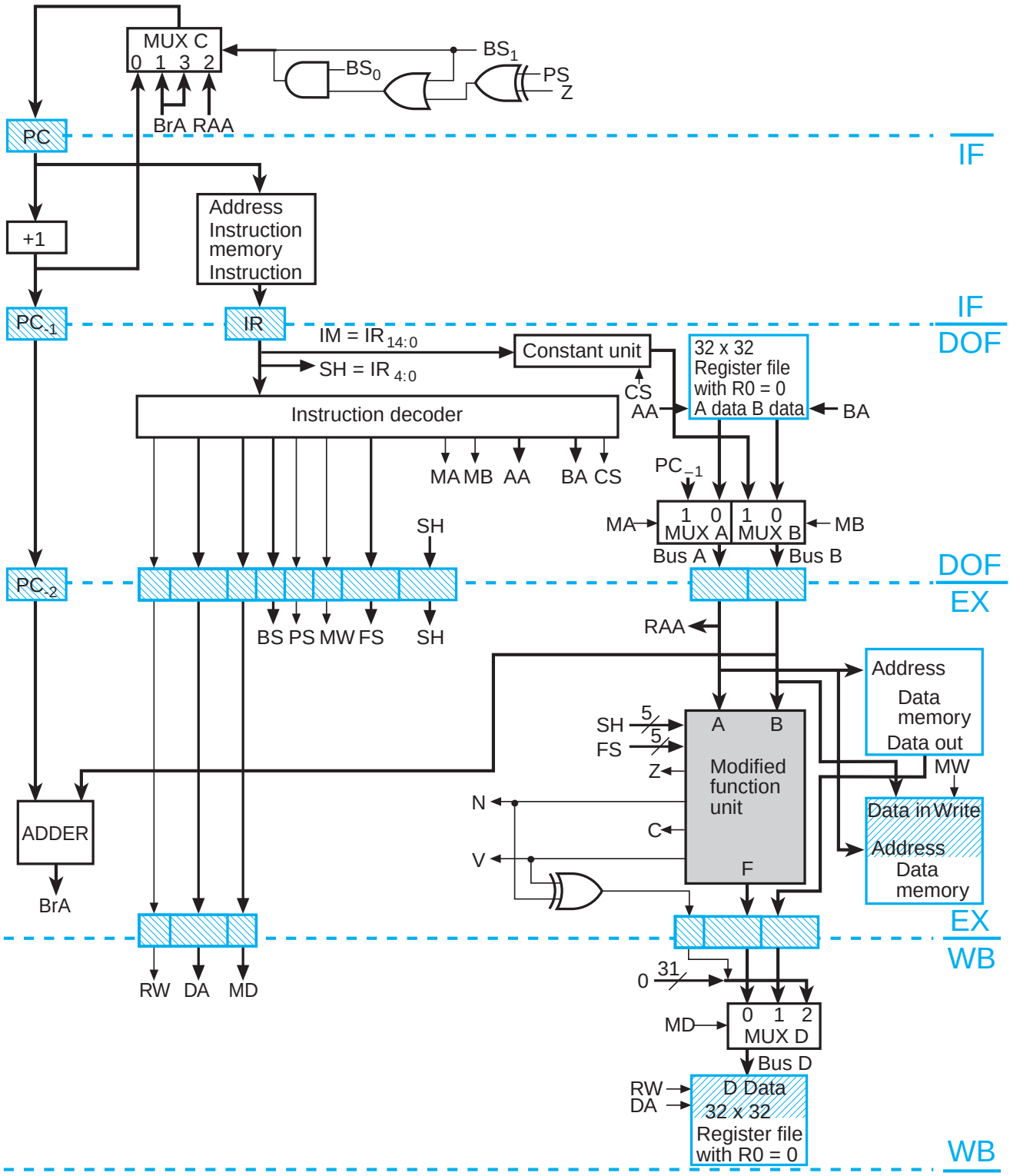
RISC CPU Instruction Formats



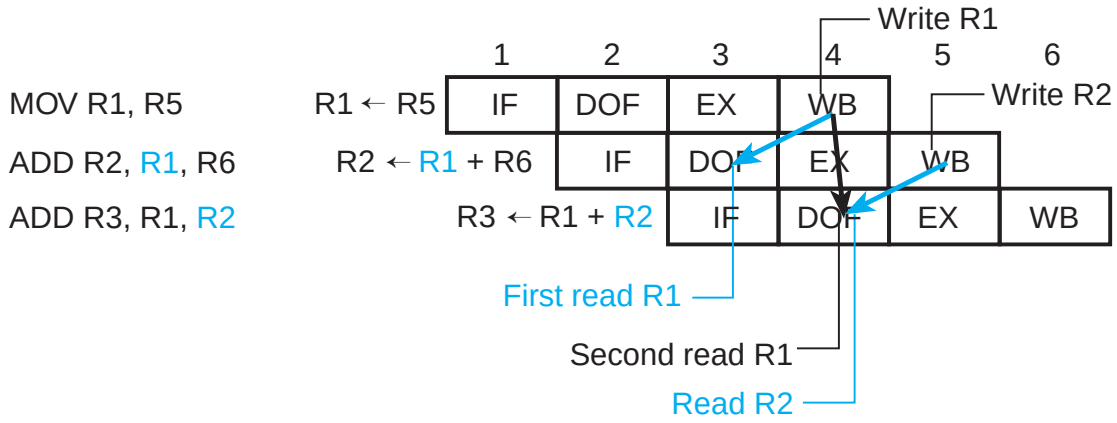
32-bit Barrel Shifter



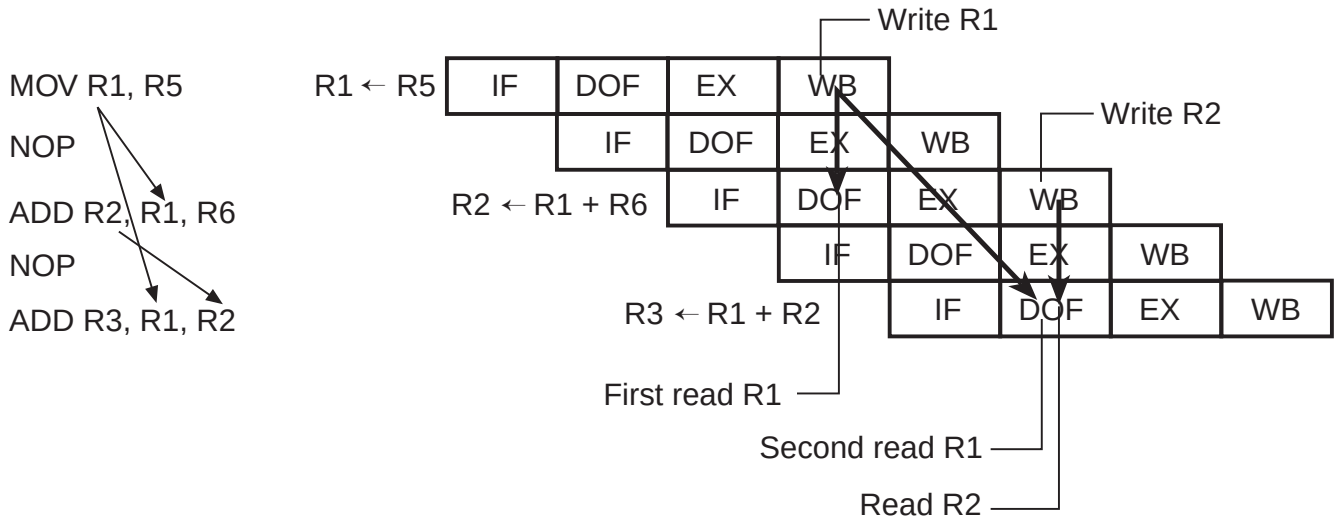
Pipelined RISC CPU



Example of Data Hazard

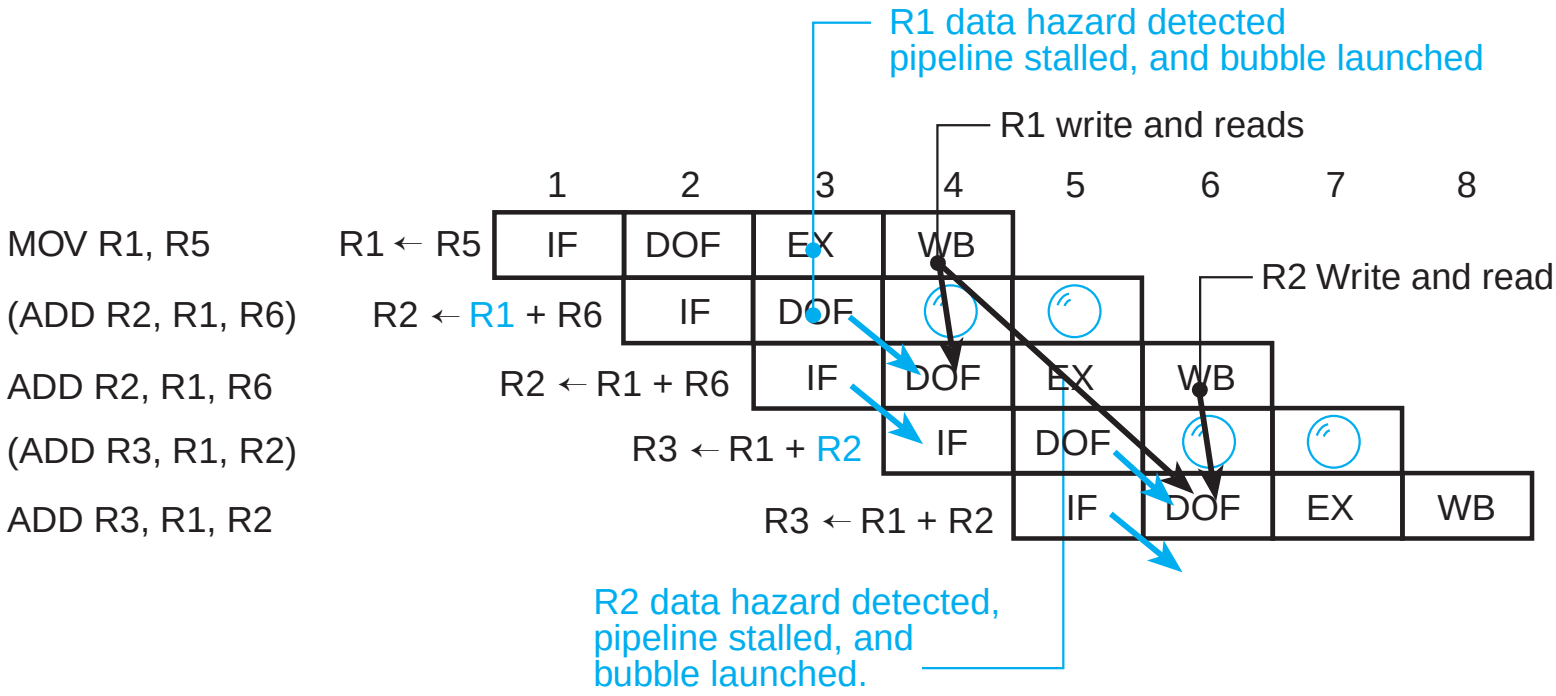


(a) The data hazard problem

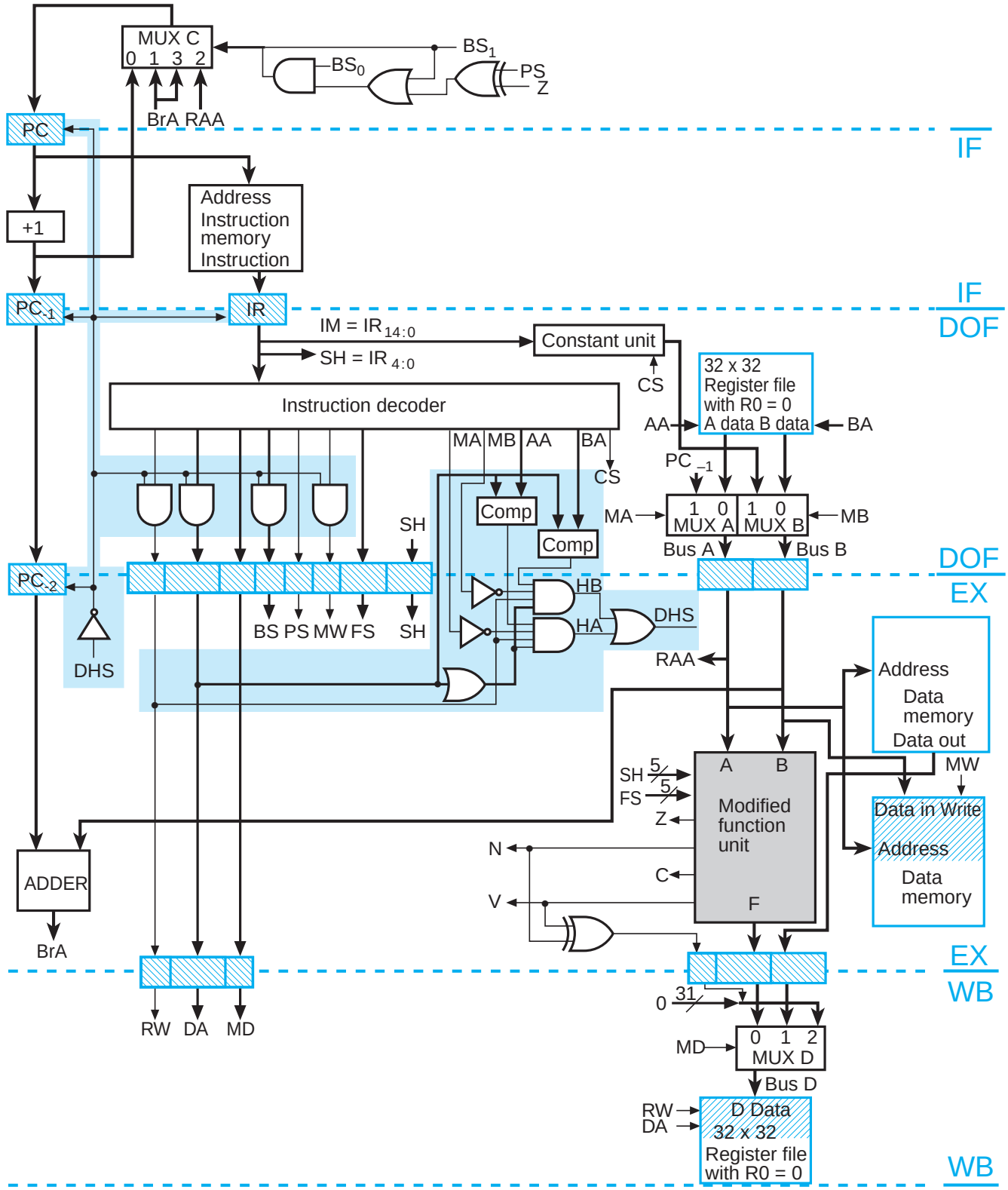


(b) A program-based solution

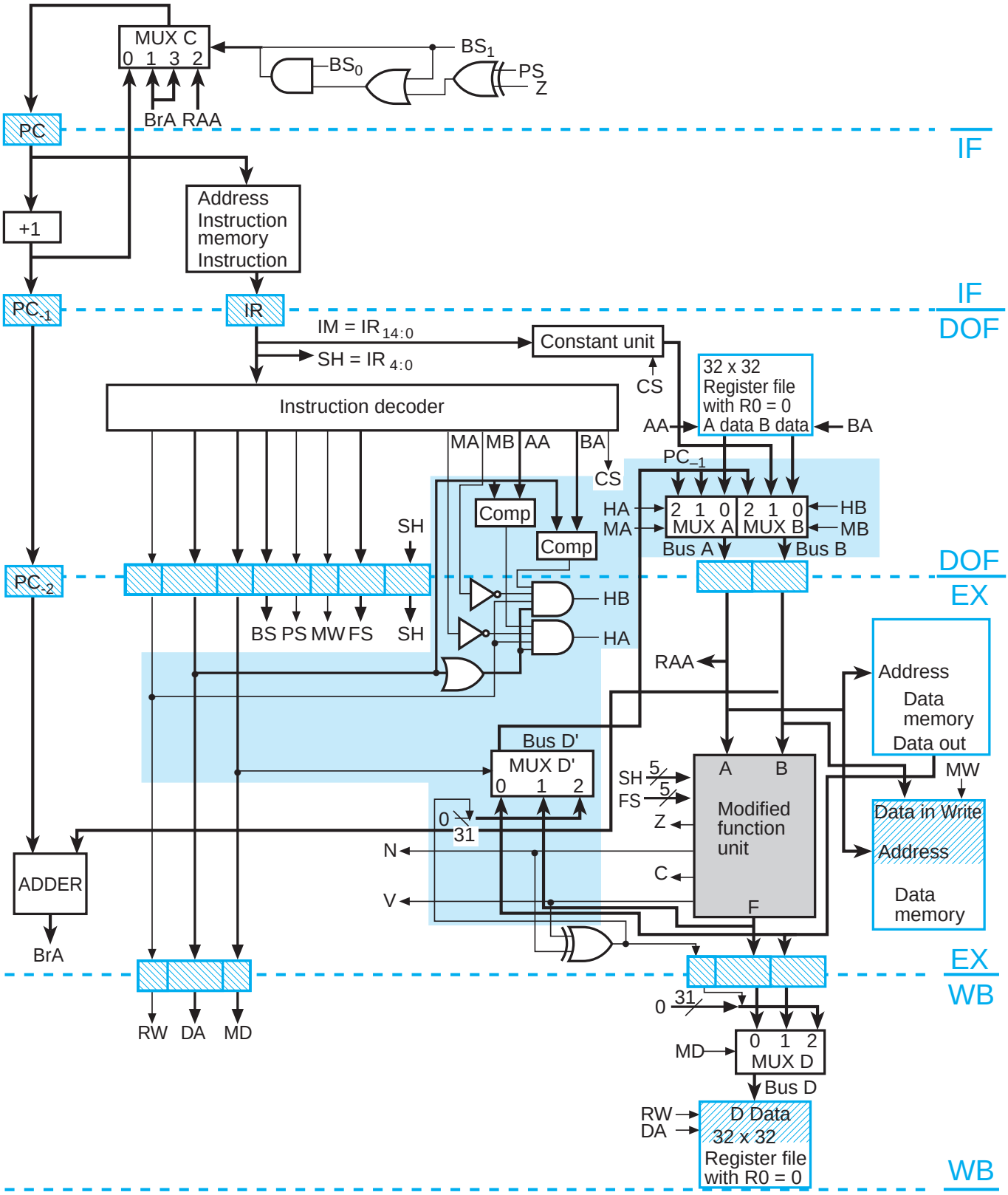
Example of Data Hazard Stall



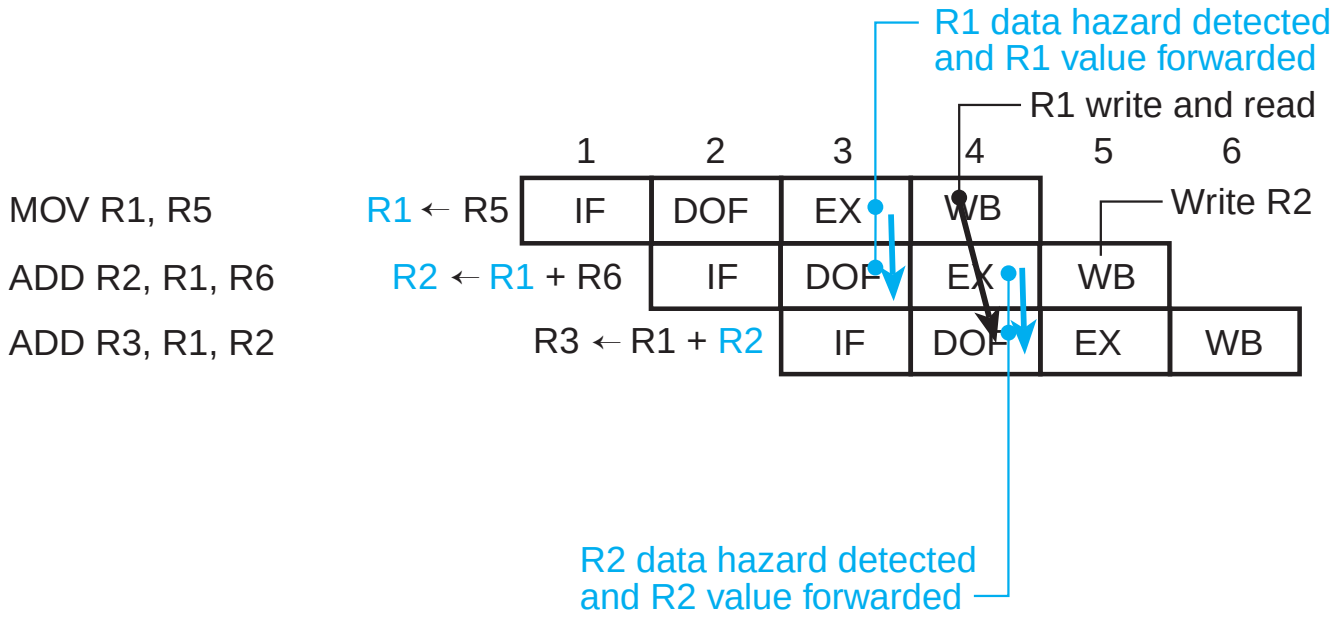
Pipelined RISC: Data Hazard Stall



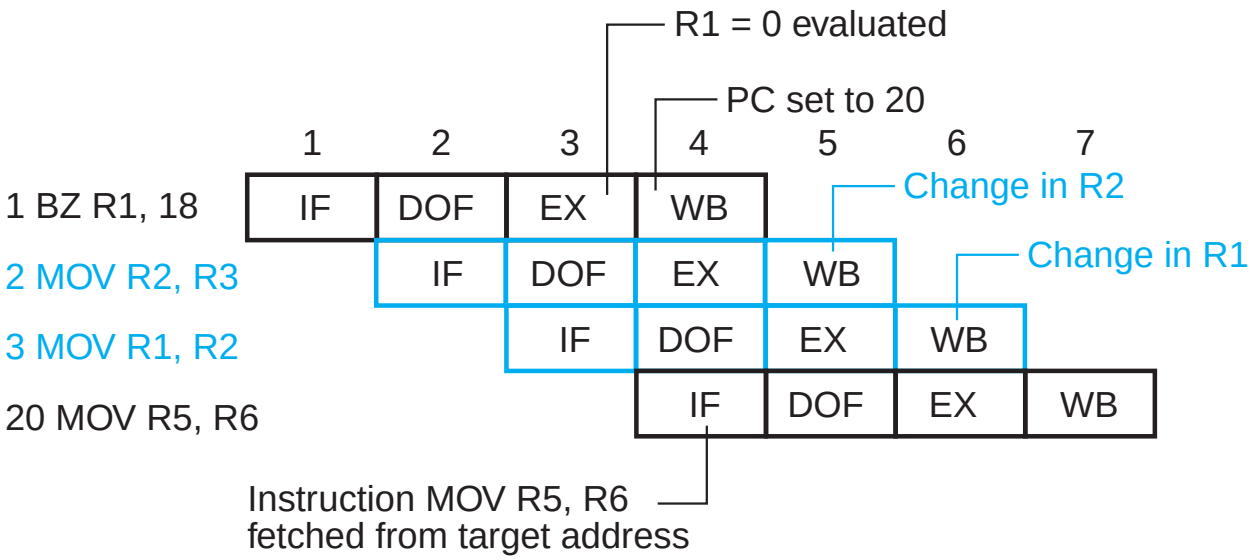
Pipeline RISC: Data Forwarding



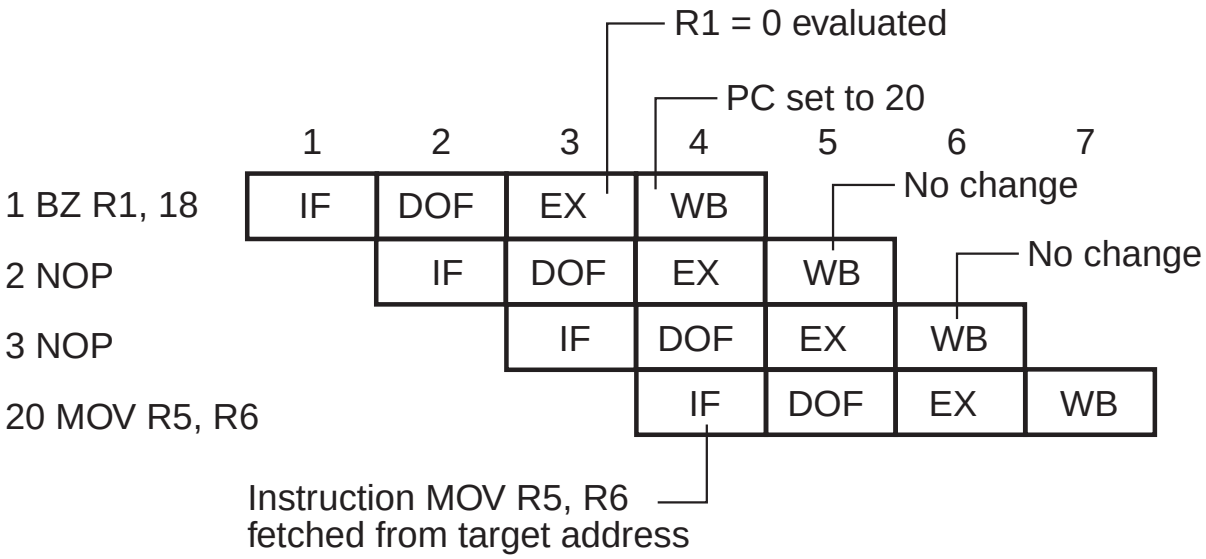
Example of Data Forwarding



Example of Control Hazard

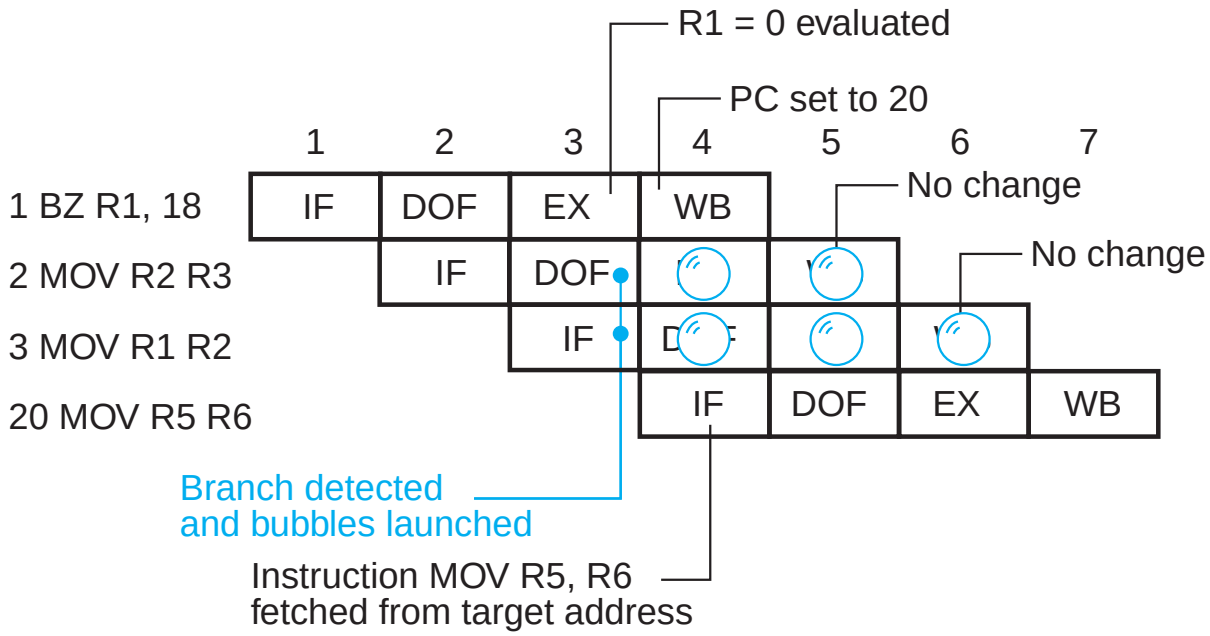


(a) Branch Hazard Problem

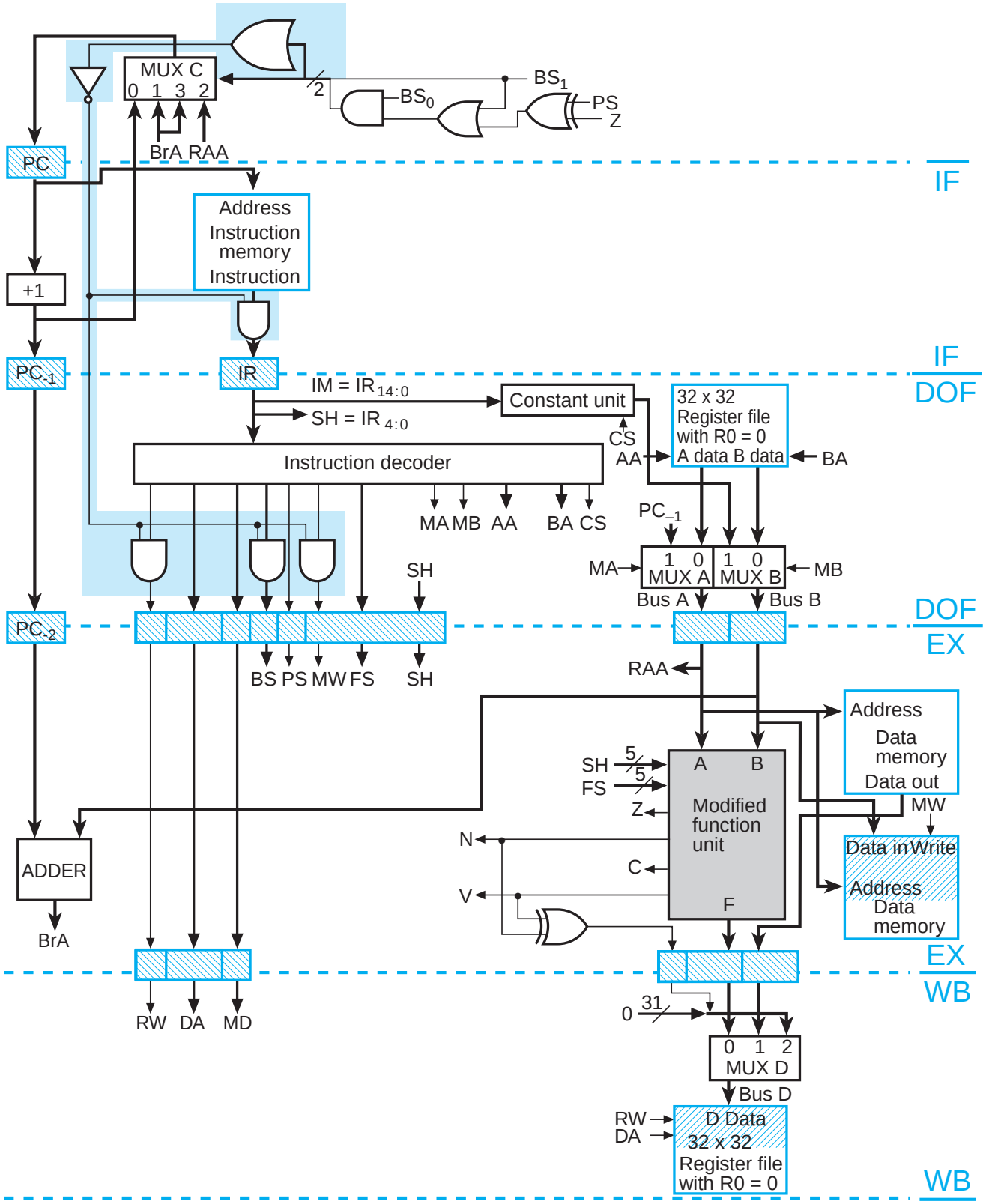


(b) Program-based Solution

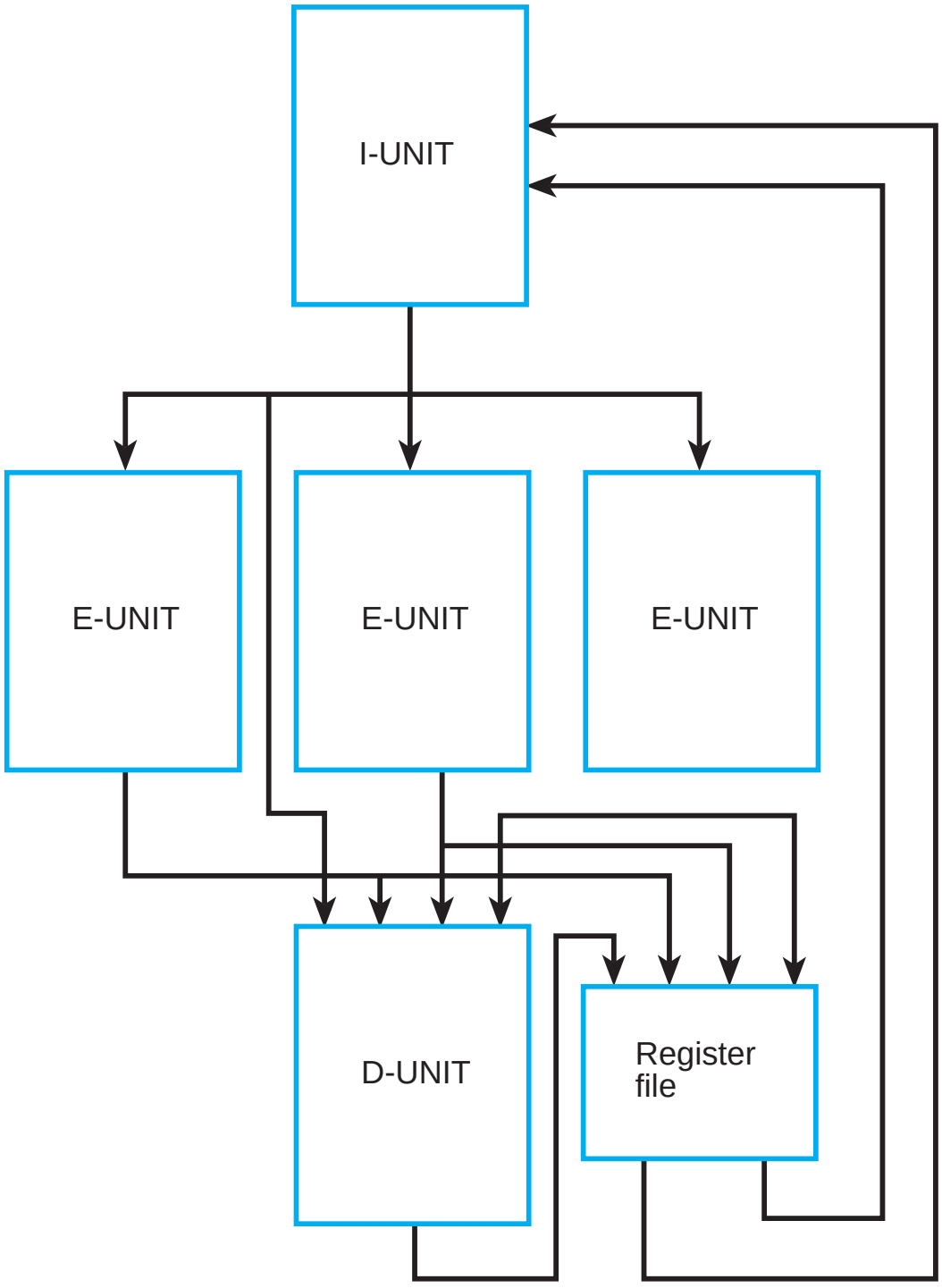
Example of Branch Prediction with Branch Taken



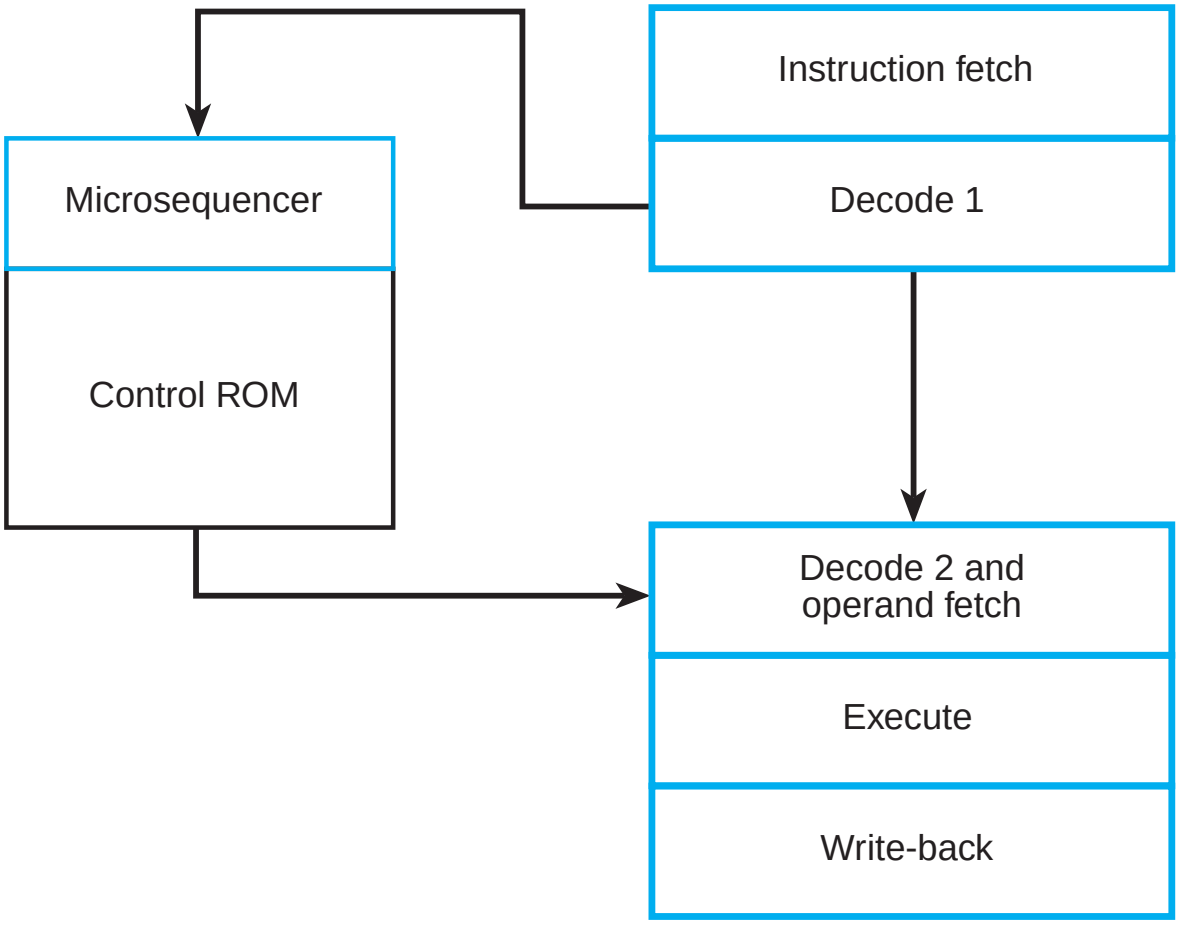
Pipelined RISC: Branch Prediction



Multiple Execution Unit Organization



Combined CISC-RISC Oganization



Superscalar Organization