Basic Symbols for Register Transfers

Symbol	Description	Examples
Letters (and numerals)	Denotes a register	$AR, R2, DR, IR$
Parentheses	Denotes a part of a register	$R2(1), R2(7{:}0), AR(L)$
Arrow	Denotes transfer of data	$R1 \leftarrow R2$
Comma	Separates simultaneous transfers	$R1 \leftarrow R2,\ R2 \leftarrow R1$
Square brackets	Specifies an address for memory	$DR \leftarrow M[AR]$

Arithmetic Microoperations

Symbolic designation	Description
$R0 \leftarrow R1 + R2$	Contents of $R1$ plus $R2$ transferred to $R0$
$R2 \leftarrow \overline{R2}$	Complement of the contents of $R2$ (1's complement)
$R2 \leftarrow \overline{R2} + 1$	2's complement of the contents of $R2$
$R0 \leftarrow R1 + \overline{R2} + 1$	$R1$ plus 2's complement of $R2$ transferred to $R0$ (subtraction)
$R1 \leftarrow R1 + 1$	Increment the contents of $R1$ (count up)
$R1 \leftarrow R1 - 1$	Decrement the contents of $R1$ (count down)

Logic Microoperations

Symbolic designation	Description
$R0 \leftarrow \overline{R1}$	Logical bitwise NOT (1's complement)
$R0 \leftarrow R1 \wedge R2$	Logical bitwise AND (clears bits)
$R0 \leftarrow R1 \vee R2$	Logical bitwise OR (sets bits)
$R0 \leftarrow R1 \oplus R2$	Logical bitwise XOR (complements bits)

Function Table for Arithmetic Circuit

Select		Input	$G = A + Y + C_{in}$	
S_1	S_0	Y	$C_{in} = 0$	$C_{in} = 1$
0	0	all 0's	$G = A$ (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 1's	$G = A - 1$(decrement)	$G = A$ (transfer)

Function Table for ALU

Operation Select					
S_2	S_1	S_0	C_{in}	Operation	Function
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 1's 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	0	0	X	$G = A \wedge B$	AND
1	0	1	X	$G = A \vee B$	OR
1	1	0	X	$G = A \oplus B$	XOR
1	1	1	X	$G = \overline{A}$	NOT (1's complement)

Function Table for 4-Bit Barrel Shifter

Select		Output				
S_1	S_0	Y_3	Y_2	Y_1	Y_0	Operation
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

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

FS	MF Select	G Select	H Select	Microoperation
00000	0	0000	0	$F = A$
00001	0	0001	1	$F = A + 1$
00010	0	0010	0	$F = A + B$
00011	0	0011	1	$F = A + B + 1$
00100	0	0100	0	$F = A + \overline{B}$
00101	0	0101	1	$F = A + \overline{B} + 1$
00110	0	0110	0	$F = A - 1$
00111	0	0111	1	$F = A$
01000	0	1000	0	$F = A \wedge B$
01010	0	1010	0	$F = A \vee B$
01100	0	1100	0	$F = A \oplus B$
01110	0	1110	0	$F = \overline{A}$
10000	1	0000	0	$F = \text{sr } A$
10001	1	0001	1	$F = \text{sl } A$

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	$F = A$	00000	Function	0	No write	0
$R1$	001	Constant	1	$F = A + 1$	00001	Data In	1	Write	1
$R2$	010			$F = A + B$	00010				
$R3$	011			$F = A + B + 1$	00011				
$R4$	100			$F = A + \overline{B}$	00100				
$R5$	101			$F = A + \overline{B} + 1$	00101				
$R6$	110			$F = A - 1$	00110				
$R7$	111			$F = A$	00111				
				$F = A \wedge B$	01000				
				$F = A \vee B$	01010				
				$F = A \oplus B$	01100				
				$F = \overline{A}$	01110				
				$F = \text{sr } A$	10000				
				$F = \text{sl } A$	10001				

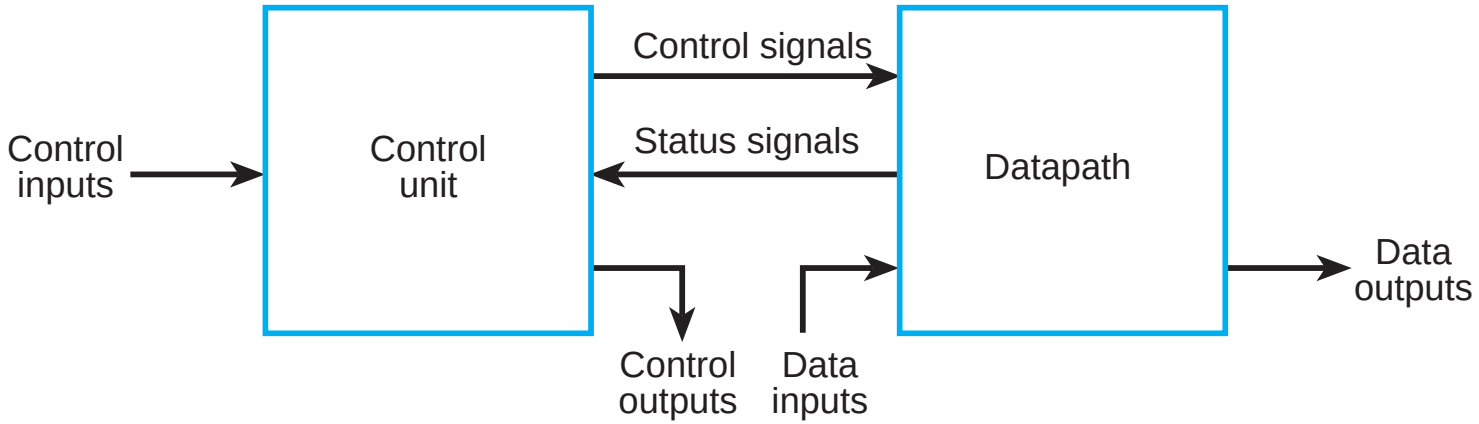
Examples of Microoperations for the Datapath Using Symbolic Notation

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

Examples of Microoperations from Table 7-11, Using Binary Control Words

Micro-operation	DA	AA	BA	MB	FS	MD	RW
$R1 \leftarrow R2 - R3$	001	010	011	0	00101	0	1
$R4 \leftarrow$ sl R6	100	110	000	0	10001	0	1
$R7 \leftarrow R7 + 1$	111	111	000	0	00001	0	1
$R1 \leftarrow R0 + 2$	001	000	000	1	00010	0	1
Data out $\leftarrow R3$	000	000	011	0	00000	0	0
$R4 \leftarrow$ Data in	100	000	000	0	00000	1	1
$R5 \leftarrow 0$	101	000	000	0	01100	0	1

Interaction between Datapath and Control Unit

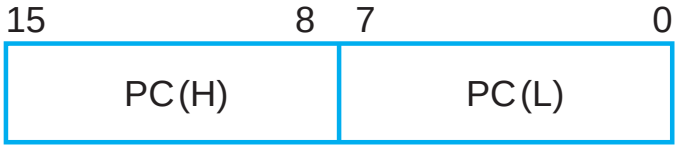


Block Diagrams of Registers

(a) Register R

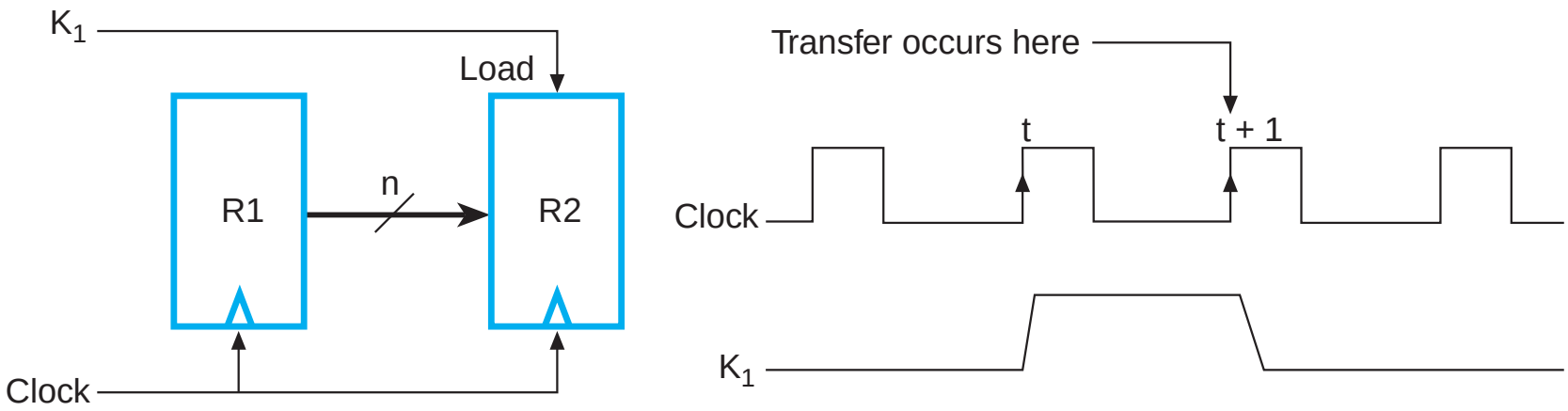
(b) Individual bits of 8-bit register

(c) Numbering of 16-bit register

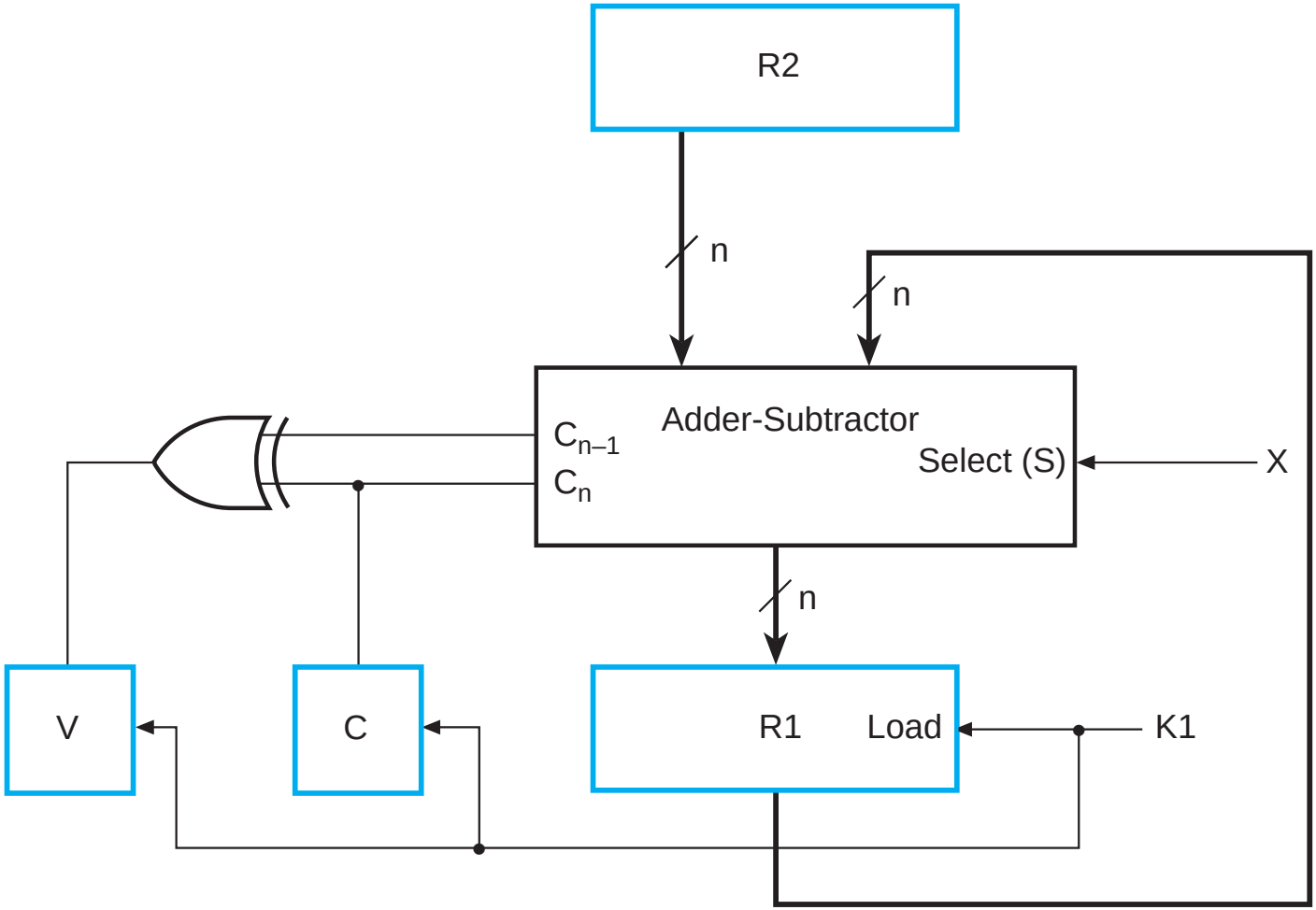


(d) Two-part 16-bit register

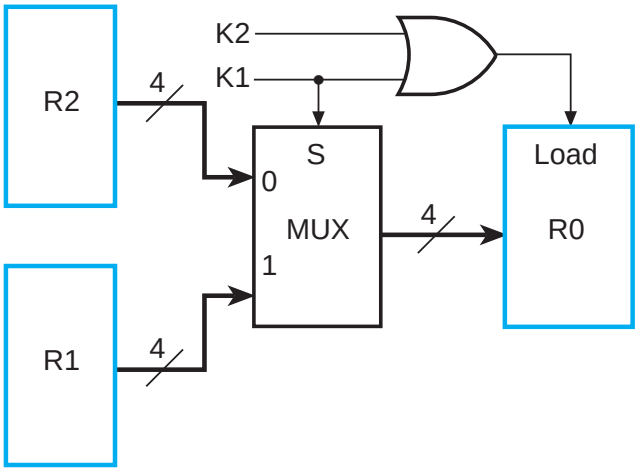
Transfer from R1 to R2 when $K_1 = 1$



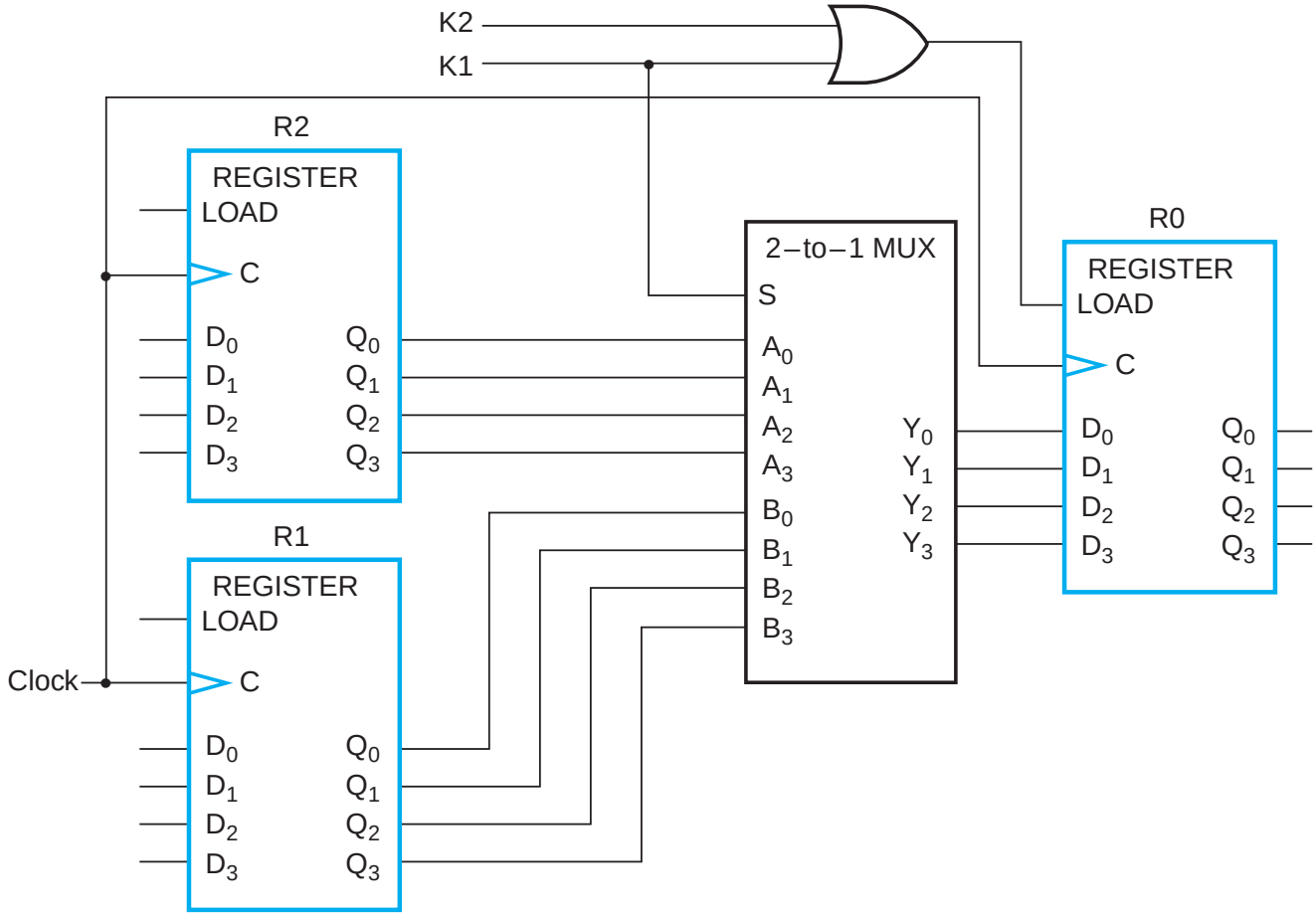
Implementation of Add and Subtract Microoperations



T-172 Use of Multiplexers to Select between Two Registers

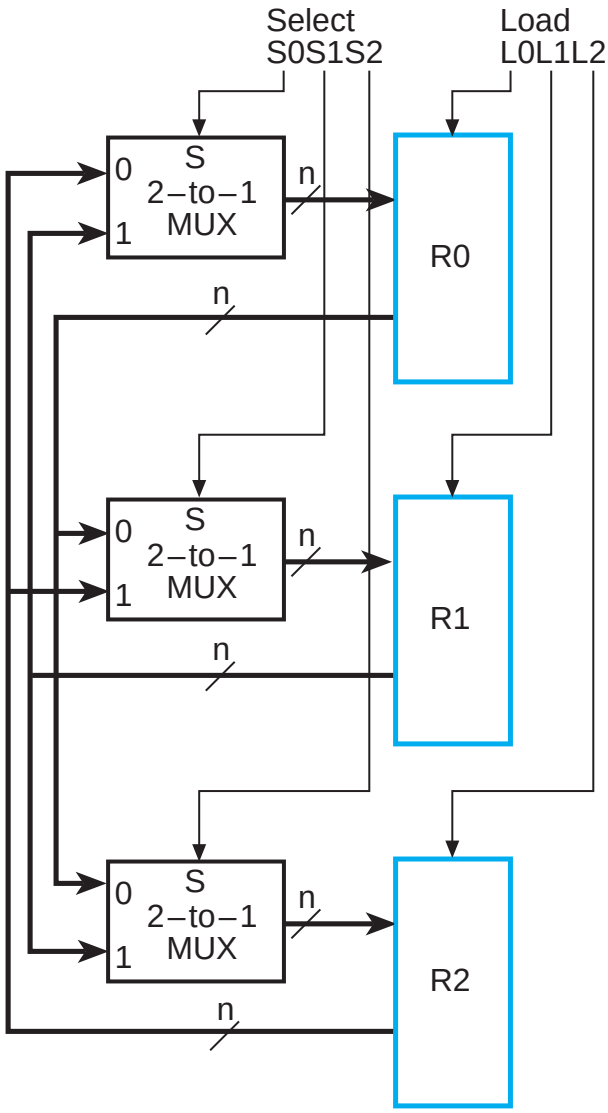


(a) Block diagram

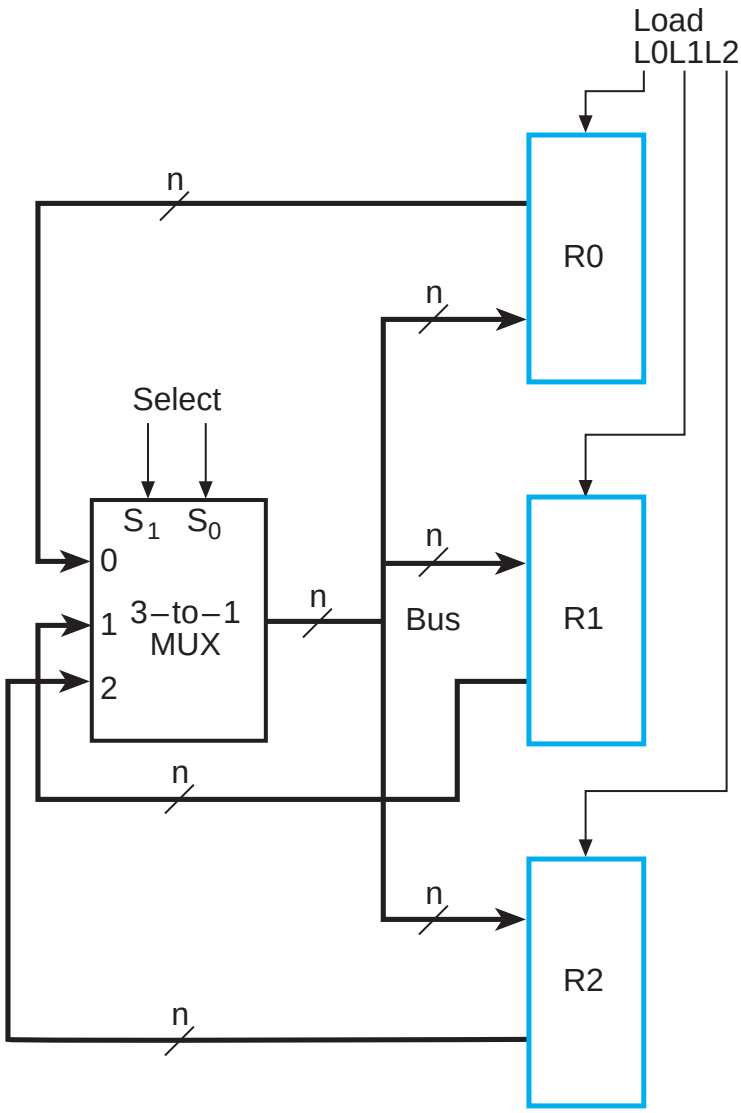


(b) Detailed logic

Single Bus versus Dedicated Multiplexers

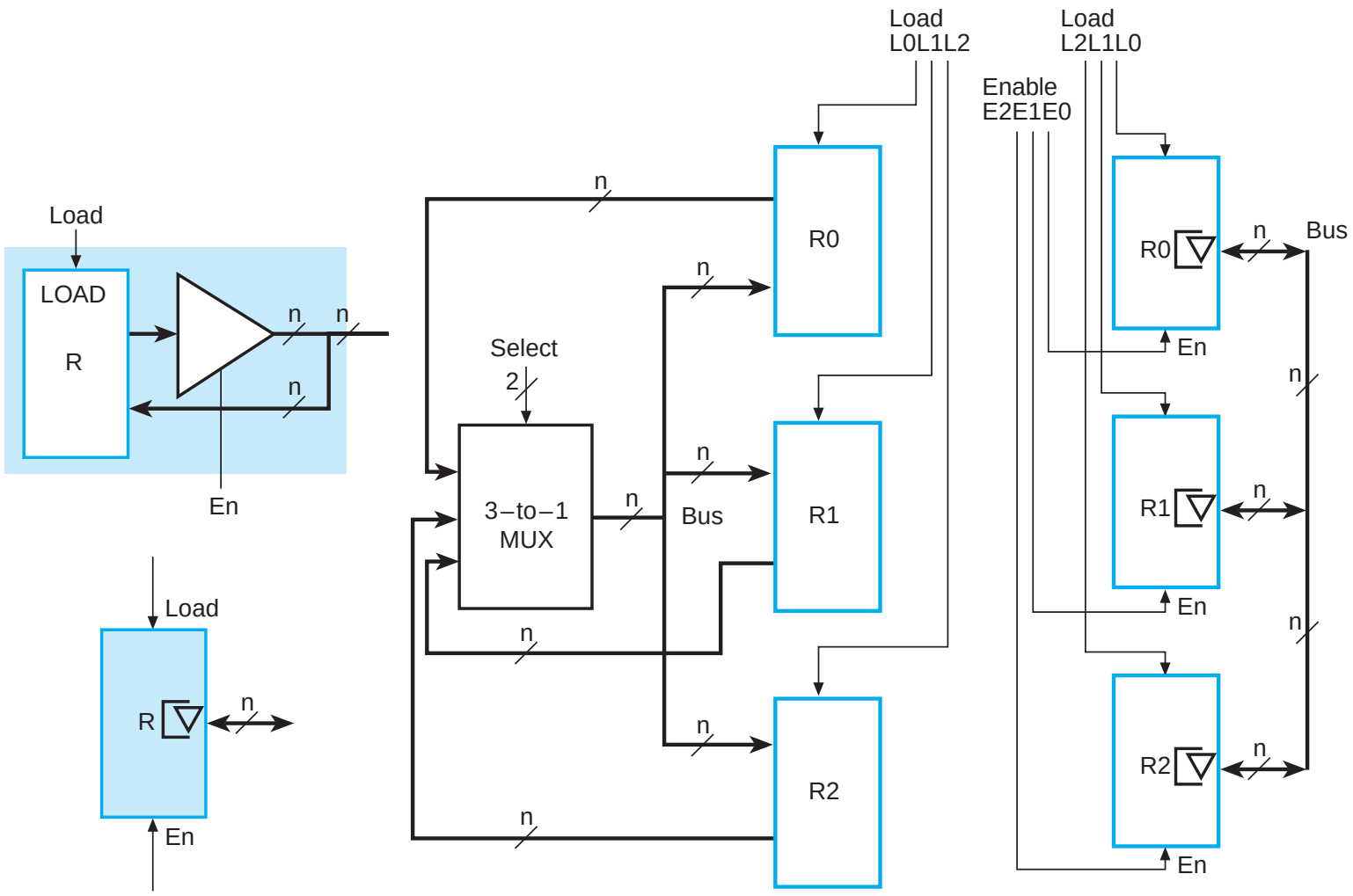


(a) Dedicated multiplexers



(b) Single Bus

Three-State Bus versus Multiplexer Bus

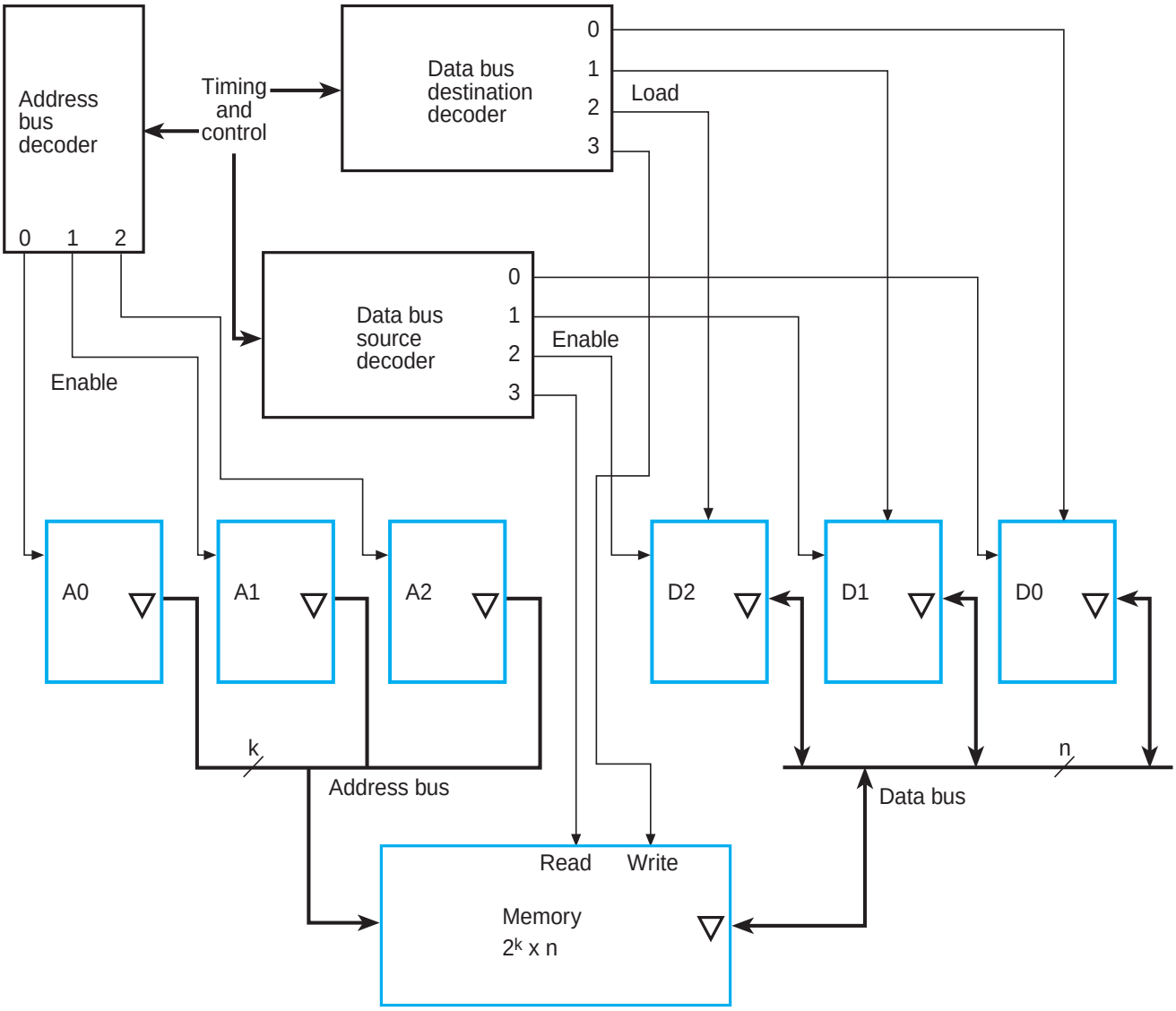


(a) Register with bidirectional input-output lines and symbol

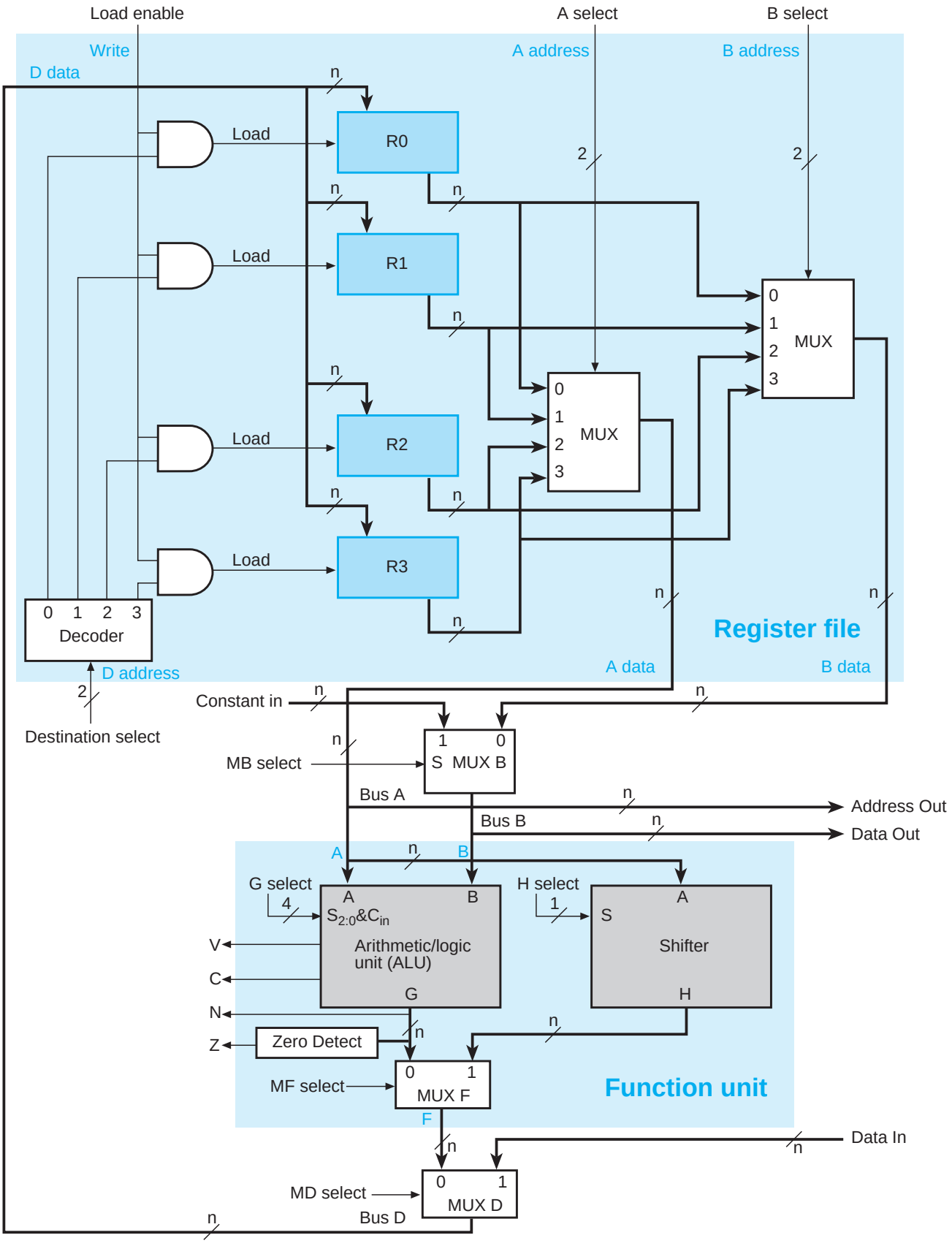
(b) Multiplexer bus

(c) Three-state bus using registers with bidirectional lines

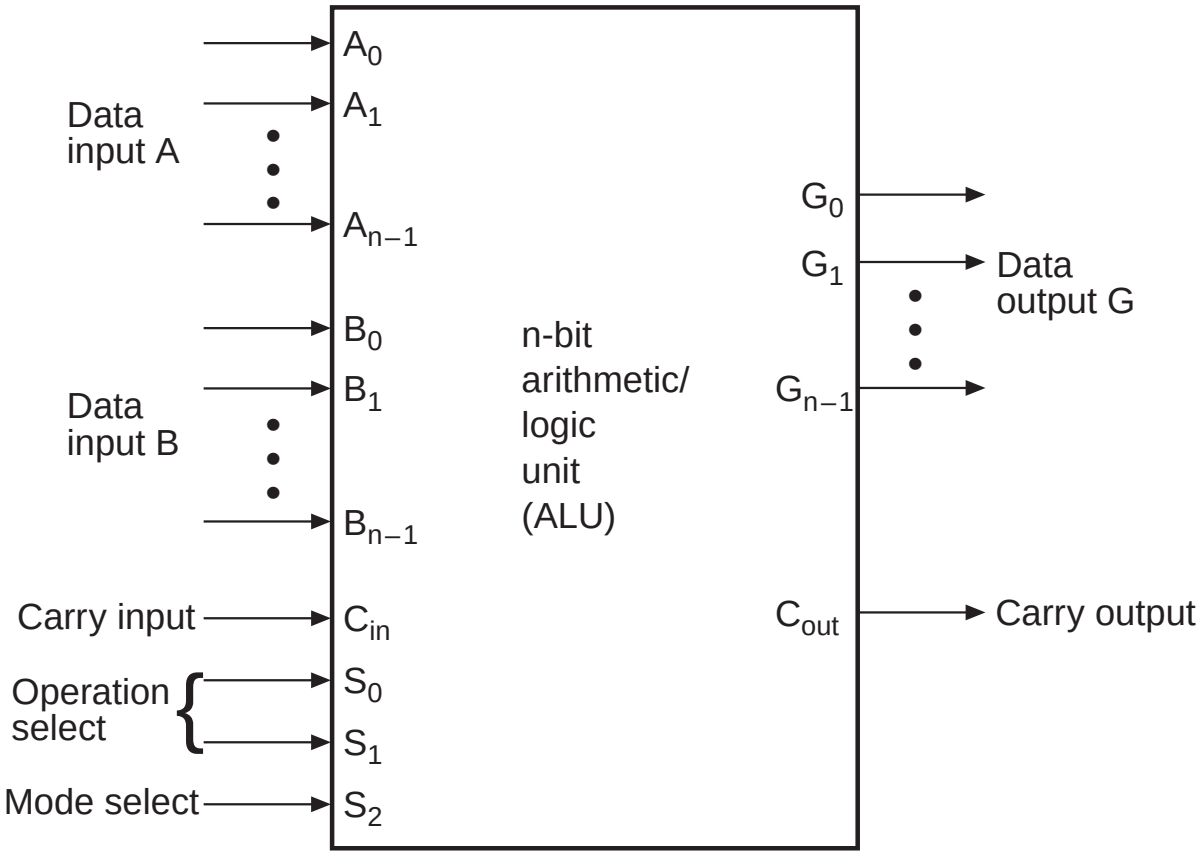
T-175 Memory Unit Connected to Address and Data Buses



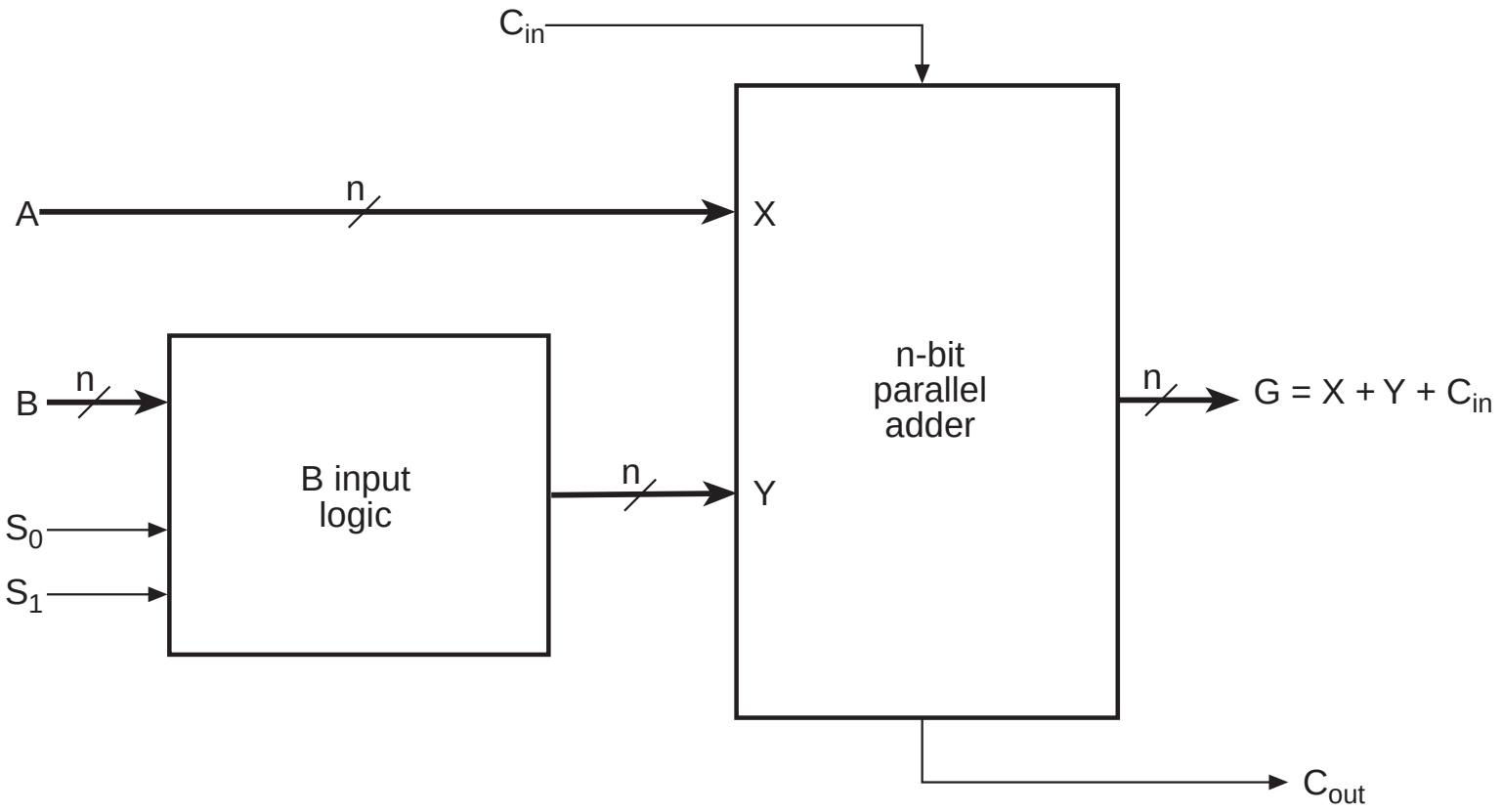
Block Diagram of a Datapath



Block Diagram of *n*-Bit ALU



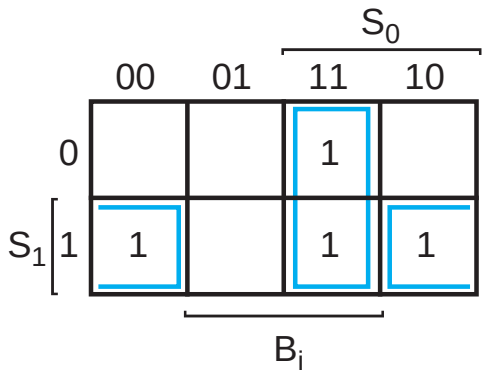
Block Diagram of an Arithmetic Circuit



B Input Logic for One Stage of Arithmetic Circuit

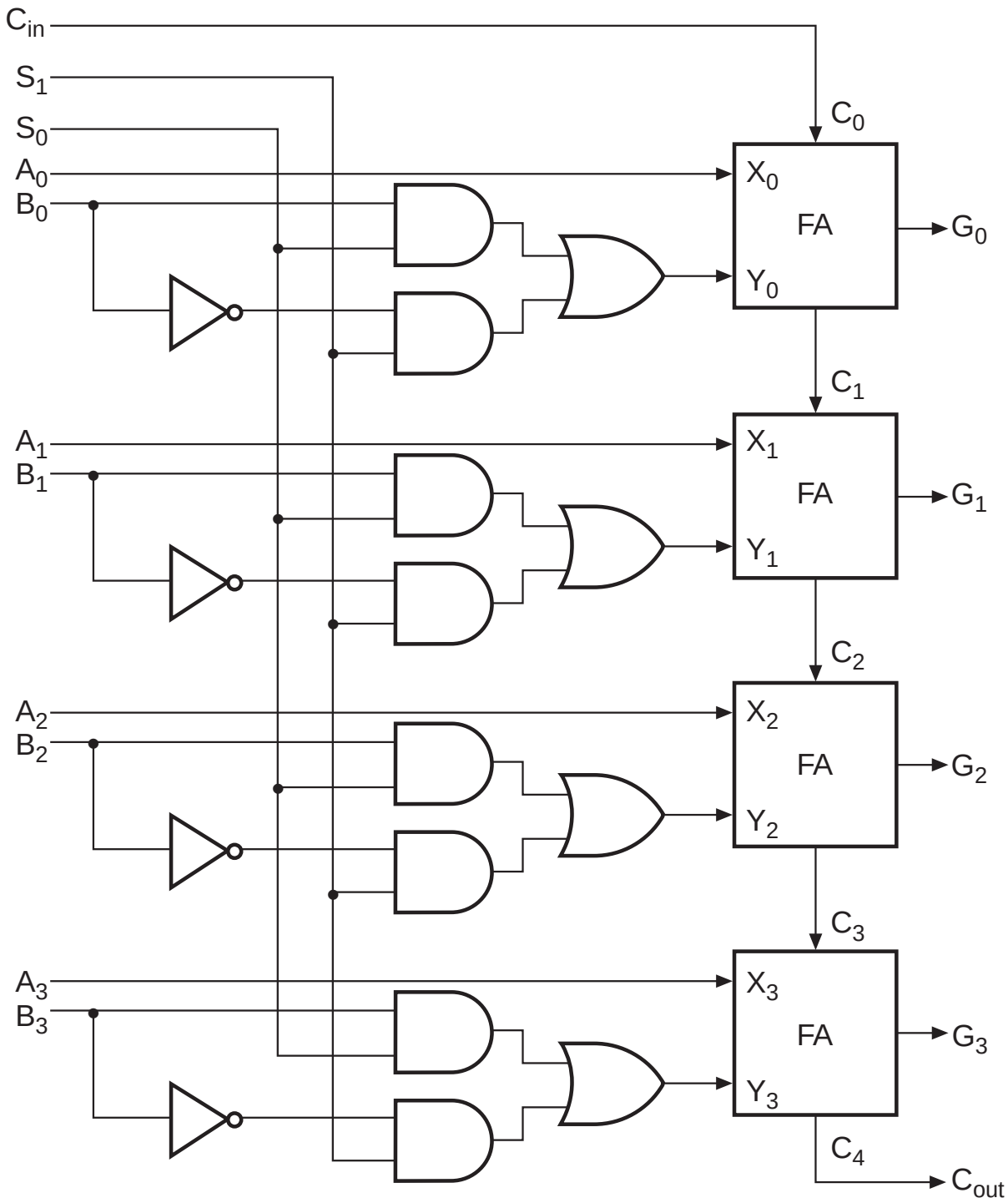
Inputs			Output	
S_1	S_0	B_i	Y_i	
0	0	0	0	$Y_i = 0$
0	0	1	0	
0	1	0	0	$Y_i = B_i$
0	1	1	1	
1	0	0	1	$Y_i = \bar{B}_i$
1	0	1	0	
1	1	0	1	$Y_i = 1$
1	1	1	1	

(a) Truth table

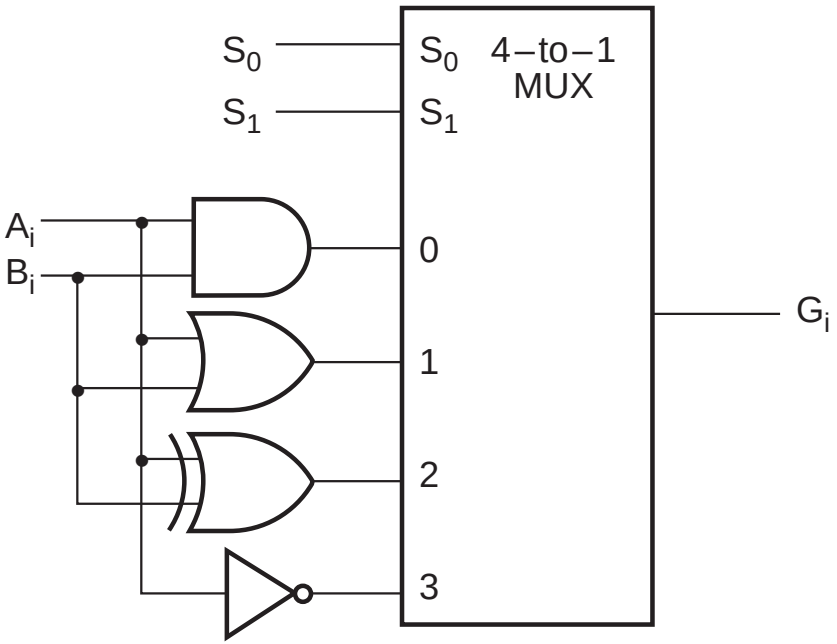


(b) Map Simplification:
$Y_i = B_i S_0 + \bar{B}_i S_1$

Logic Diagram of a 4-bit Arithmetic Circuit



One Stage of Logic Circuit

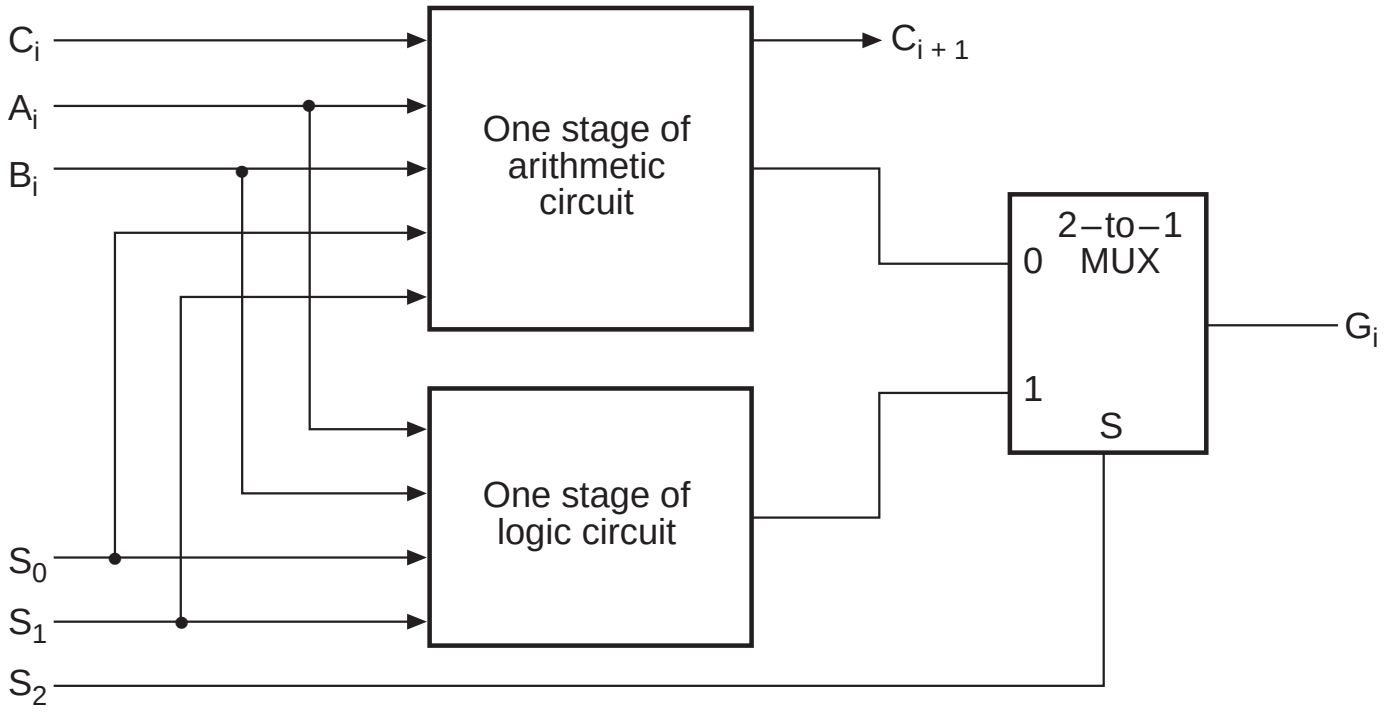


(a) Logic Diagram

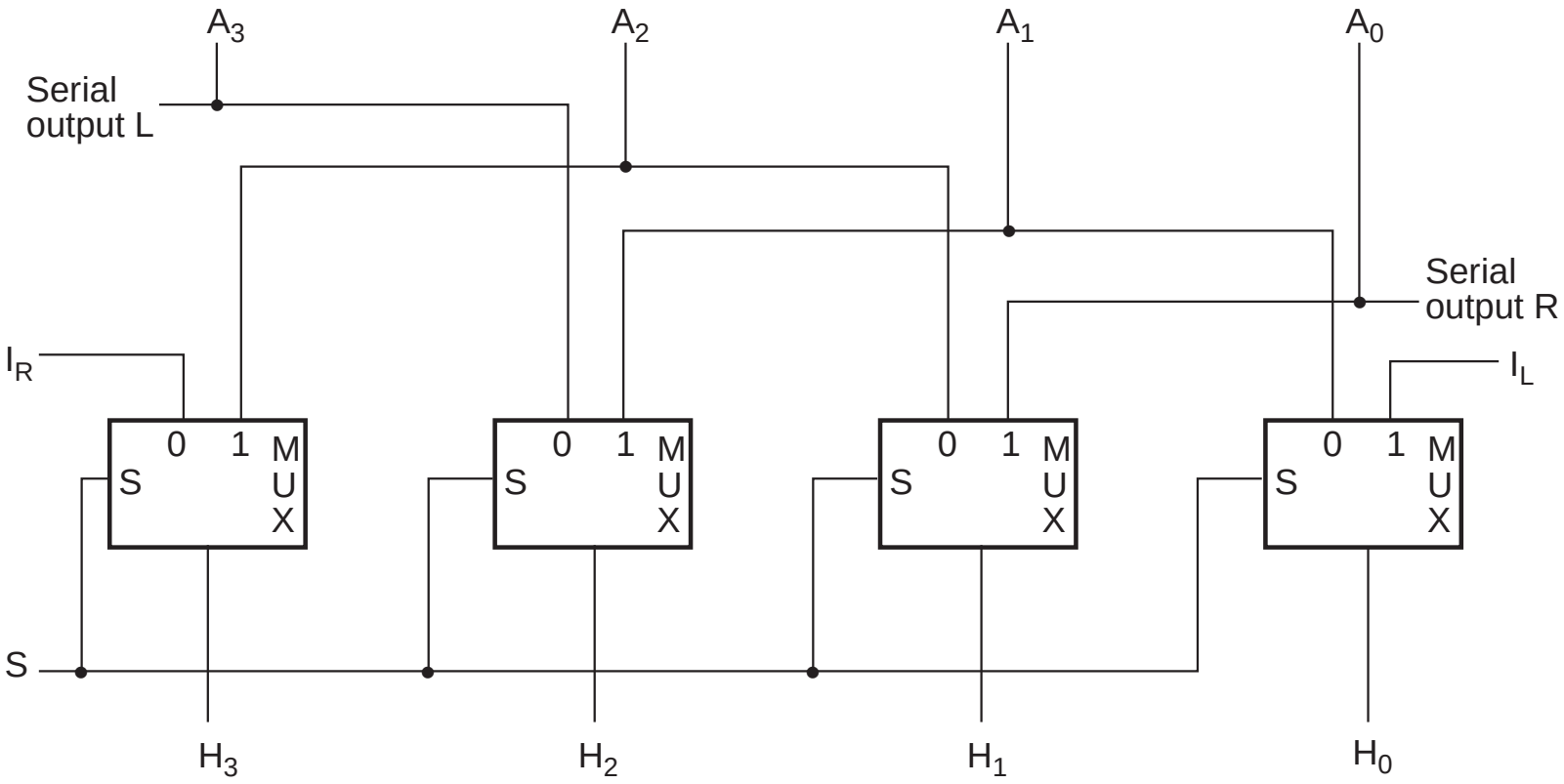
S_1	S_0	Output	Operation
0	0	$G = A \wedge B$	AND
0	1	$G = A \vee B$	OR
1	0	$G = A \oplus B$	XOR
1	1	$G = \overline{A}$	NOT

(b) Function Table

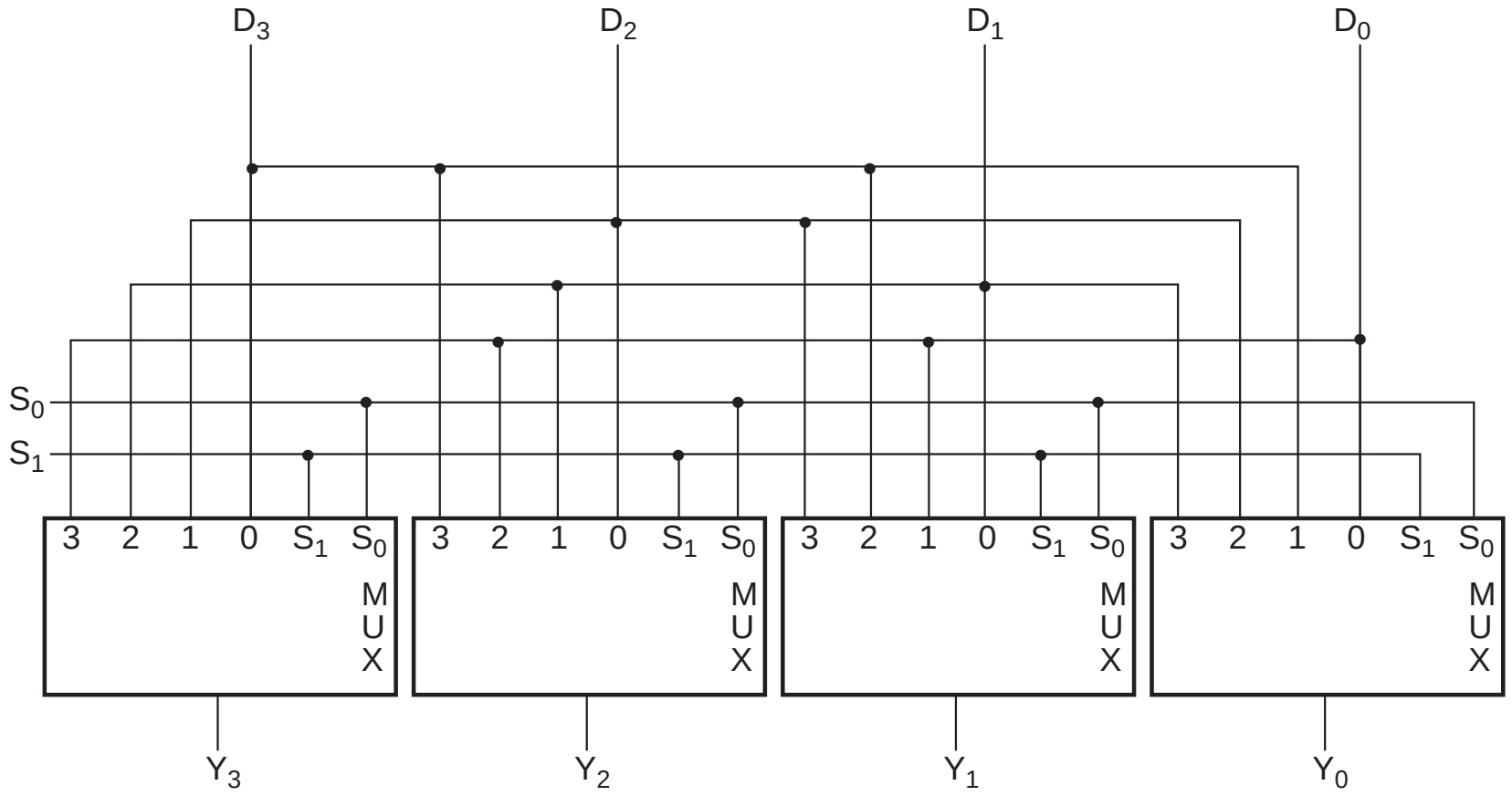
One Stage of ALU



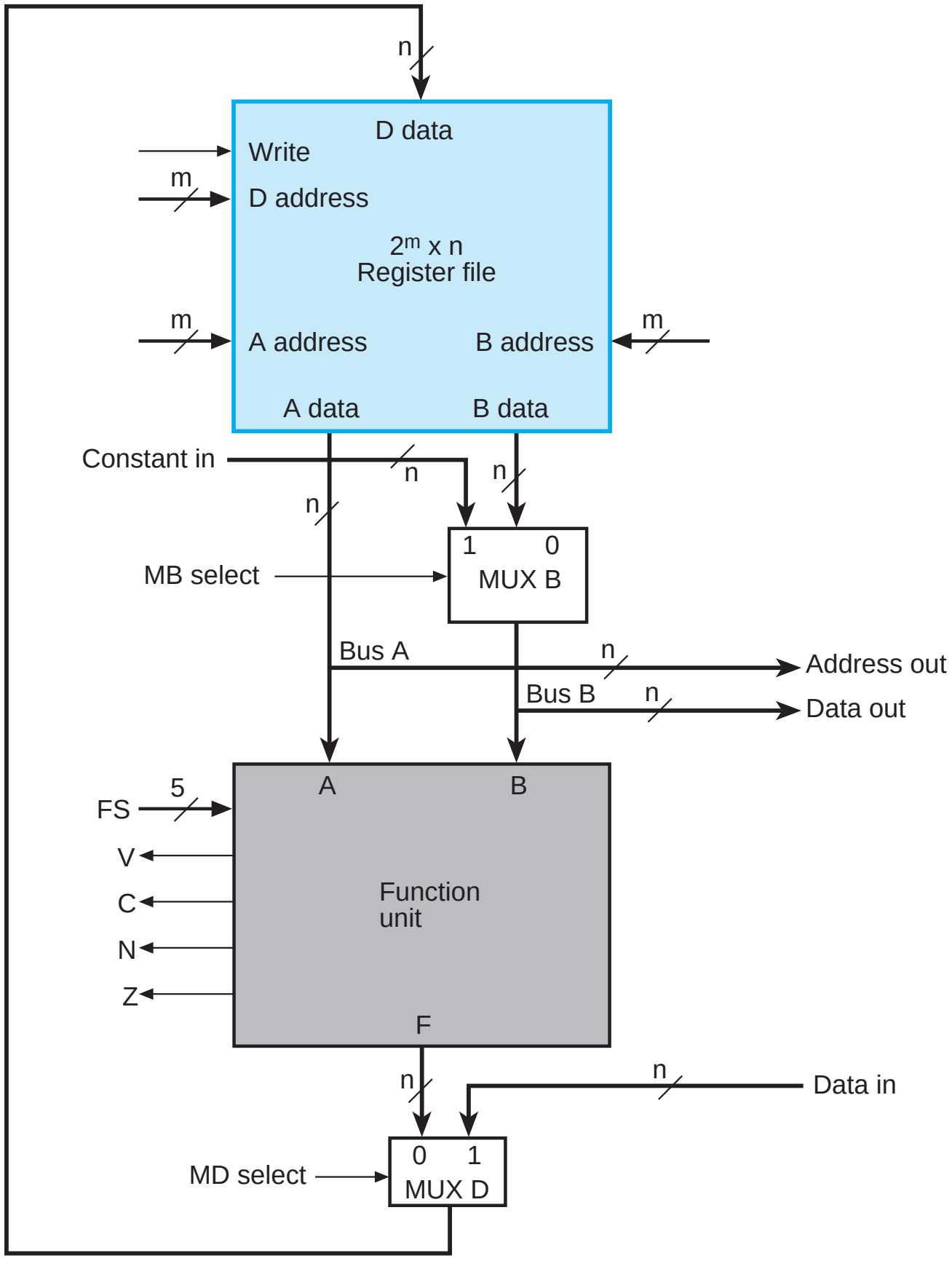
4-Bit Basic Shifter



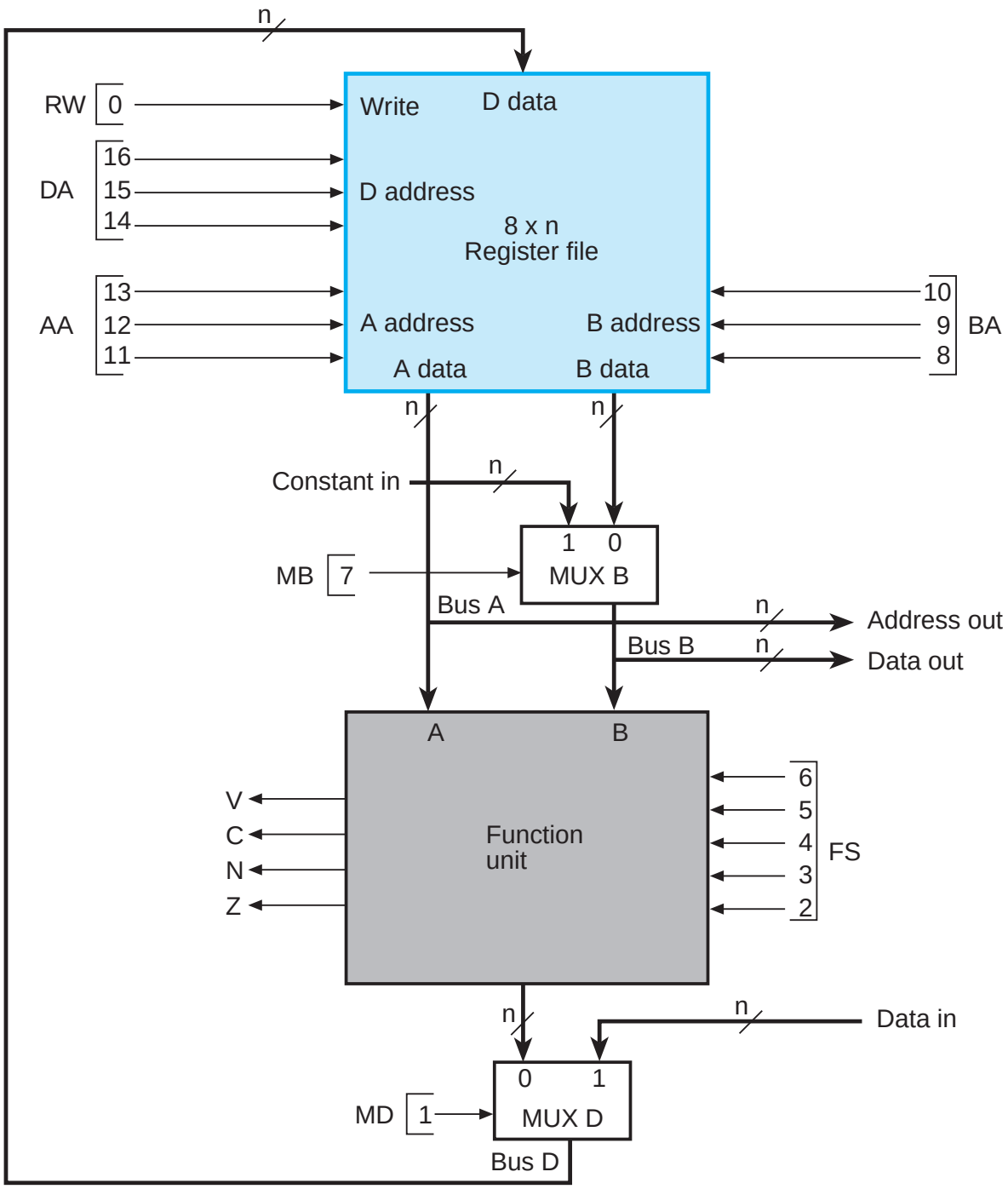
4-Bit Barrel Shifter



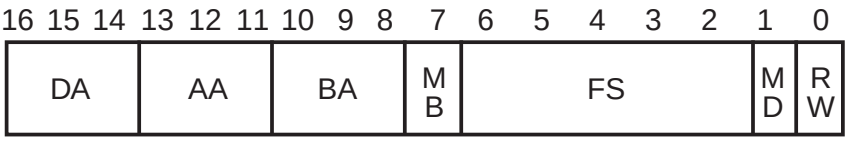
Block Diagram of Datapath Using the Register File and Function Unit



Datapath with Control Variables

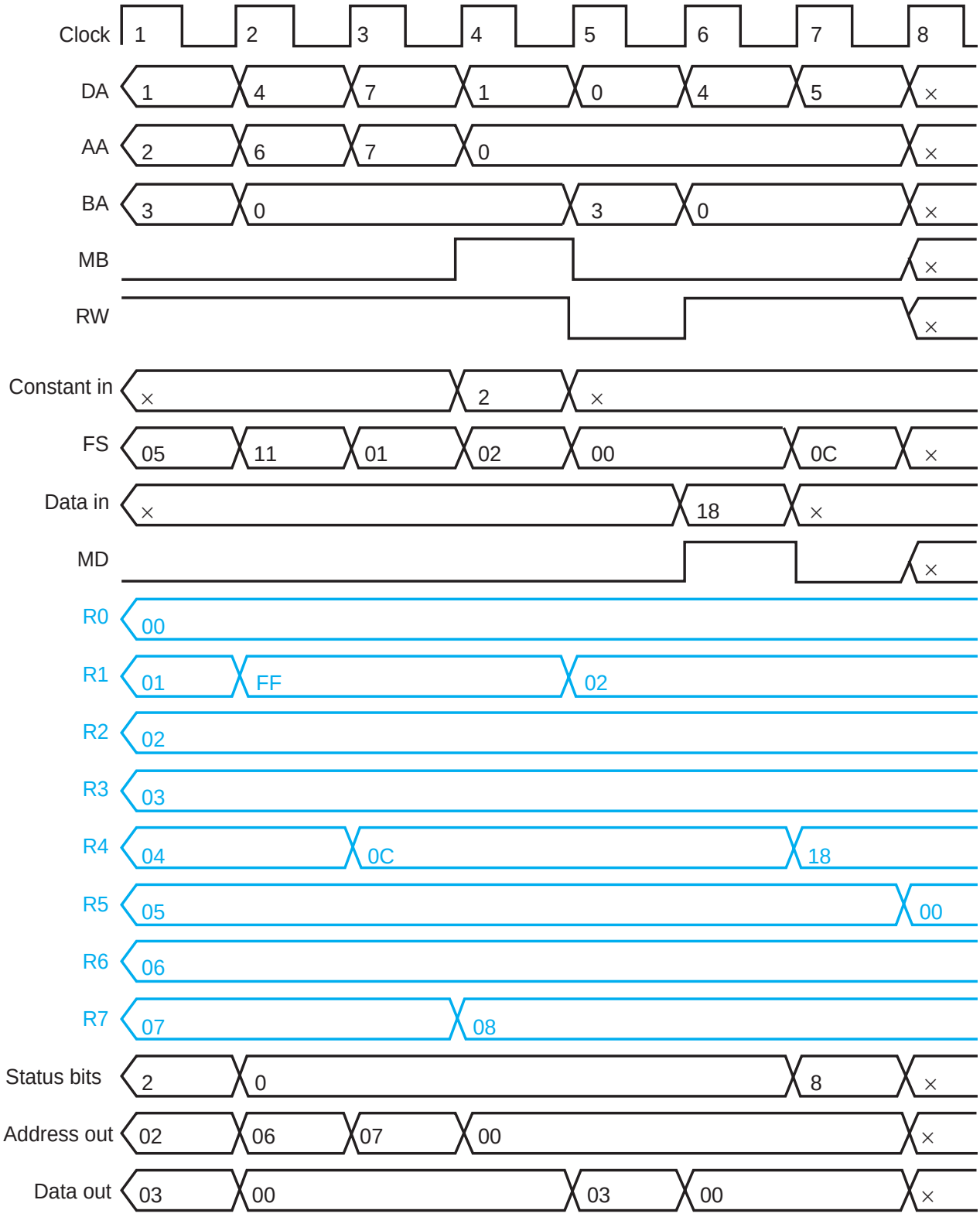


(a) Block Diagram

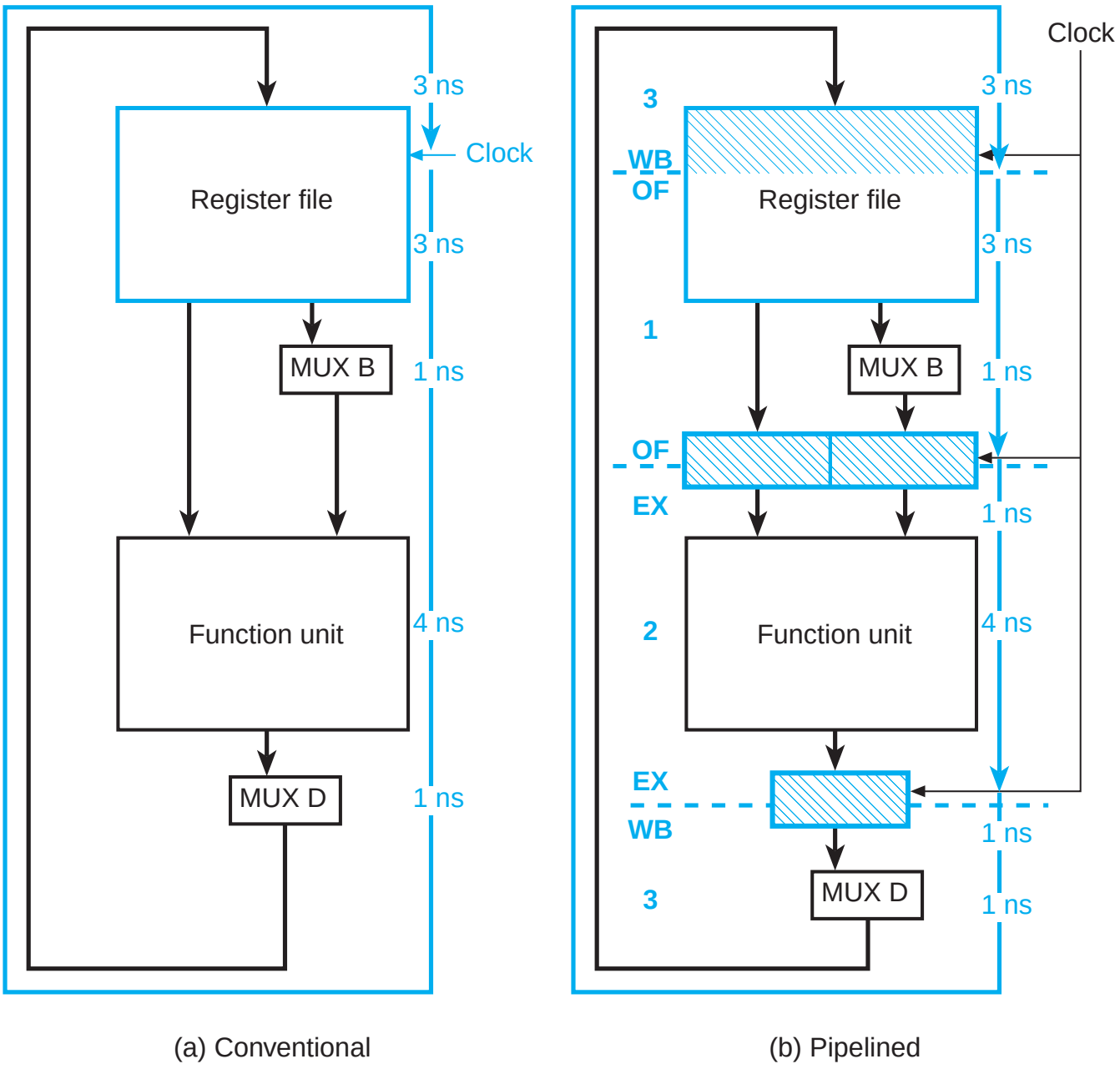


(b) Control word

T-187 Simulation of the Microoperation Sequence in Table 7-12



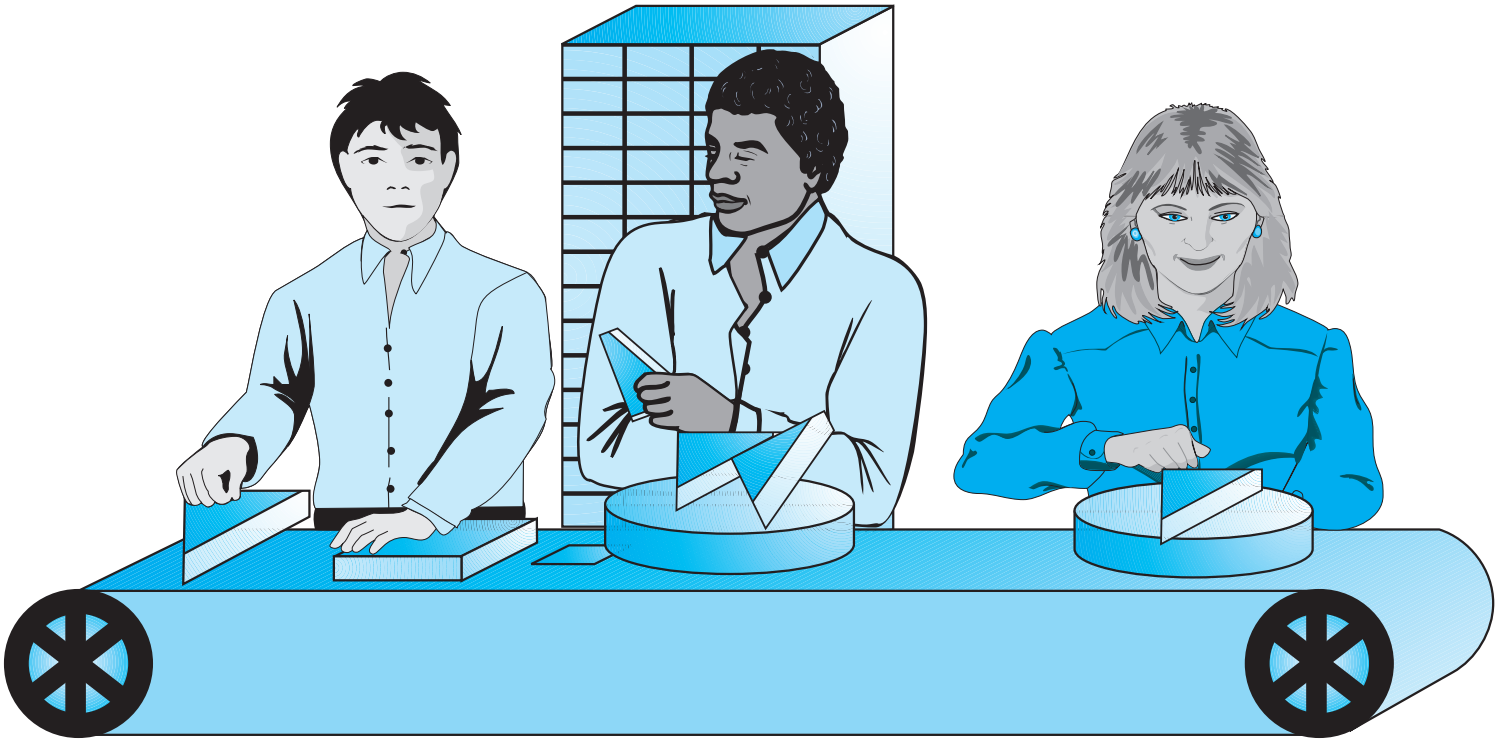
Conventional and Pipelined Datapaths



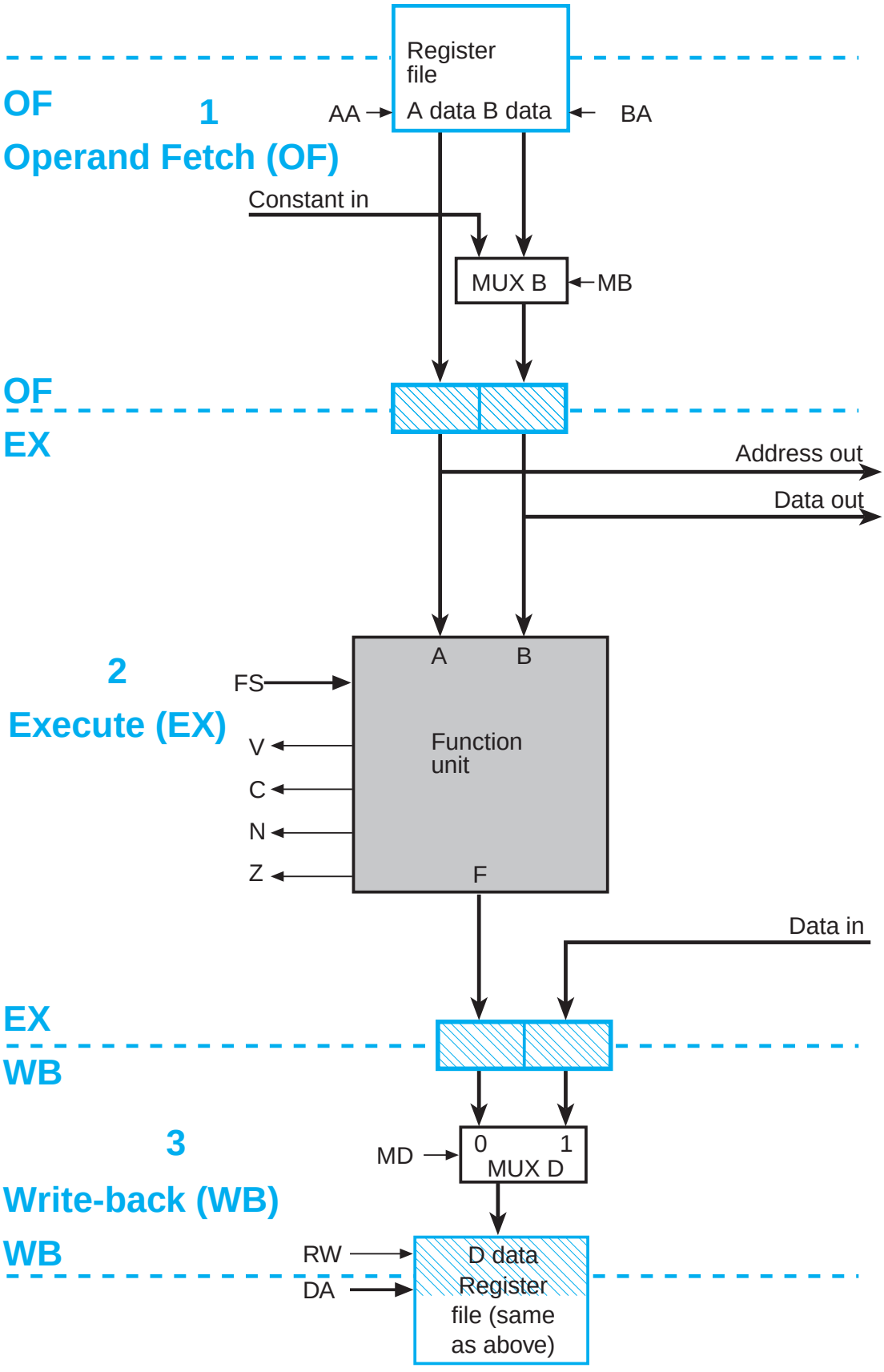
(a) Conventional

(b) Pipelined

Assembly Line Analogy to Datapath Pipeline



Block Diagram of Pipelined Datapath



Pipeline Execution Pattern for Microoperation Sequence in Table 7-12