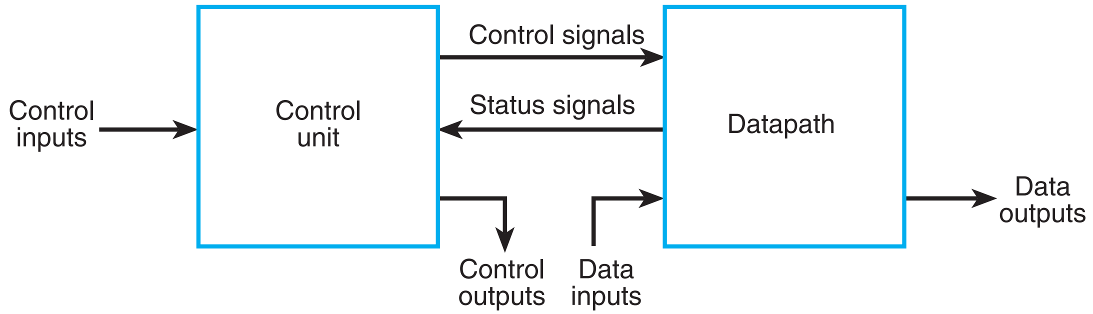


Fig. 7-1 Interaction between Datapath and Control Unit

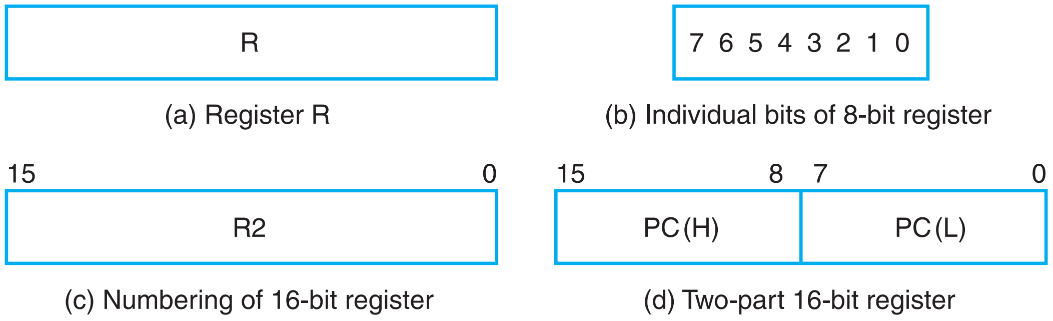

R
(a) Register R
7 6 5 4 3 2 1 0
(b) Individual bits of 8-bit register
15
0
R2
(c) Numbering of 16-bit register
15
8
7
0
PC (H)
PC (L)
(d) Two-part 16-bit register

Fig. 7-2 Block Diagrams of Registers

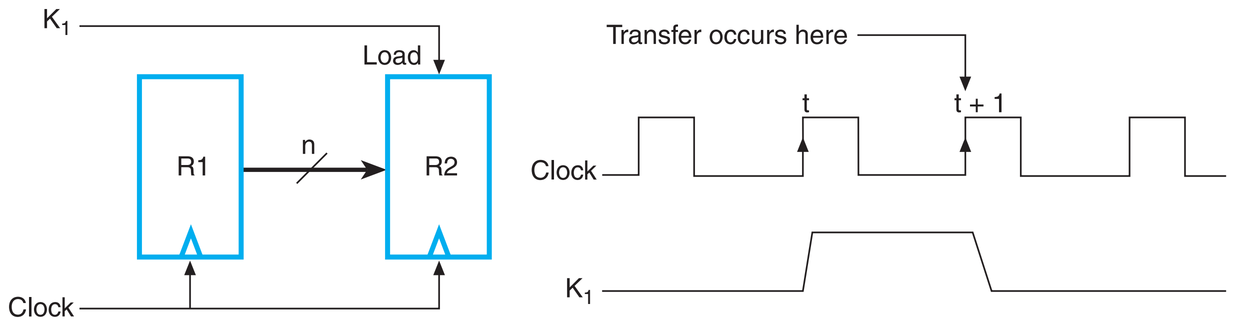


Fig. 7-3 Transfer from *R*1 to *R*2 when $K_1 = 1$

TABLE 7-1
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	$DR \leftarrow M[AR]$

Table 7-1 Basic Symbols for Register Transfers

TABLE 7-2
Textbook RTL, VHDL, and Verilog Symbols for Register Transfers

Operation	Text RTL	VHDL	Verilog
Combinational Assignment	=	<= (concurrent)	assign = (non-blocking)
Register Transfer	$\leftarrow$	<= (concurrent)	<= (non-blocking)
Addition	+	+	+
Subtraction	–	–	–
Bitwise AND	$\wedge$	and	&
Bitwise OR	$\vee$	or	|
Bitwise XOR	$\oplus$	xor	^
Bitwise NOT	$\overline{\ }$	not	~
Shift left (logical)	sl	sll	<<
Shift right(logical)	sr	srl	>>
Vectors/Registers	A(3:0)	A(3 downto 0)	A[3:0]
Concatenation	||	&	{ , }

Table 7-2 Textbook RTL, VHDL, and Verilog Symbols for Register Transfers

TABLE 7-3
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)

Table 7-3 Arithmetic Microoperations

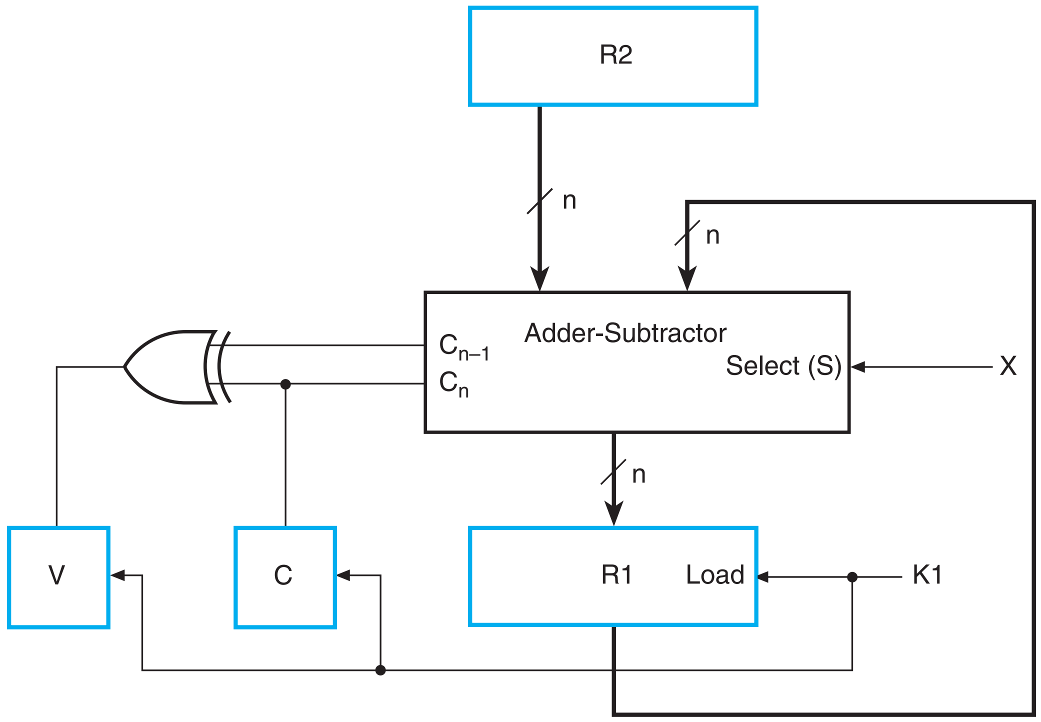


Fig. 7-4 Implementation of Add and Subtract Microoperations

TABLE 7-4
Logic Microoperations

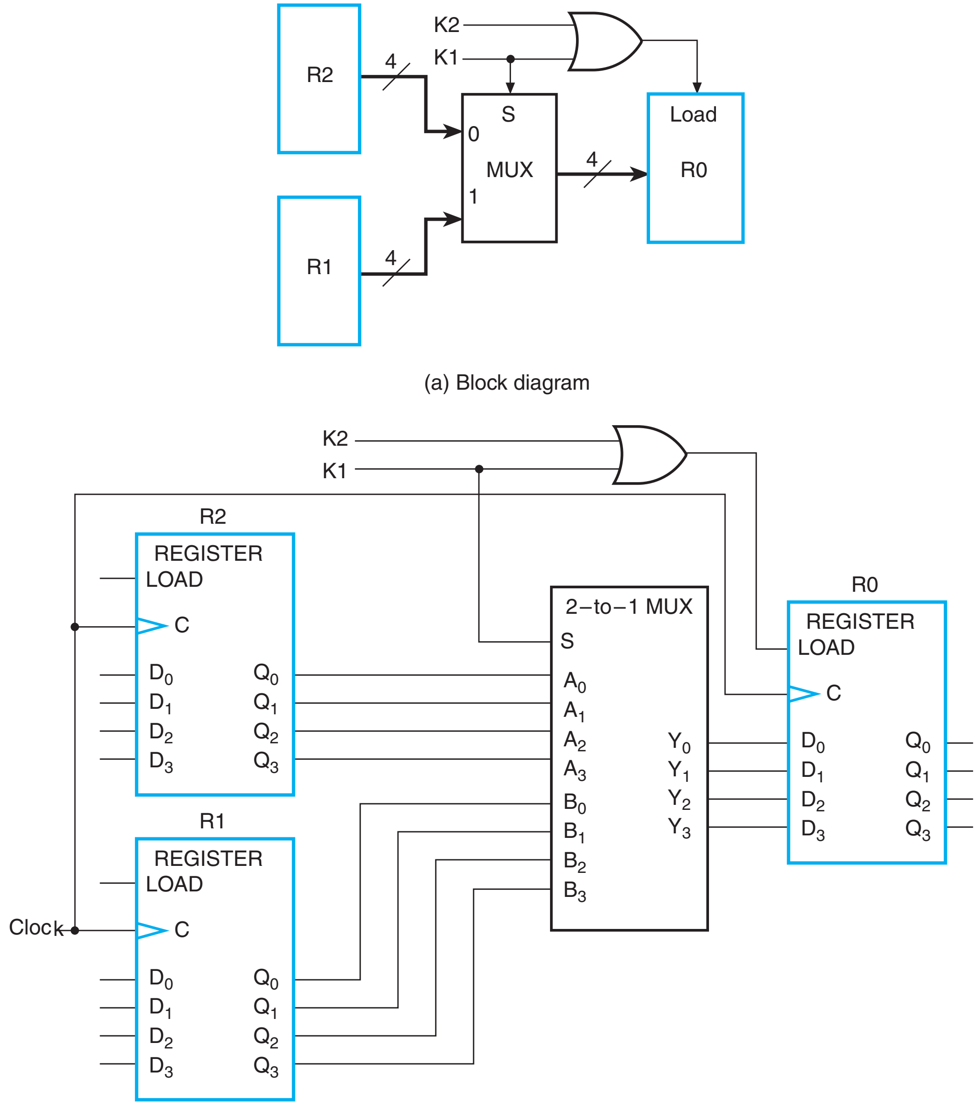
Symbolic designation	Description
$R0 \leftarrow \overline{R1}$	Logical bitwise NOT (1's complement)
$R0 \leftarrow R1 \wedge R2$	Logical bitwise AND (clears bits)
$R0 \leftarrow R1 \vee R2$	Logical bitwise OR (sets bits)
$R0 \leftarrow R1 \oplus R2$	Logical bitwise XOR (complements bits)

Table 7-4 Logic Microoperations

TABLE 7-5
Examples of Shifts

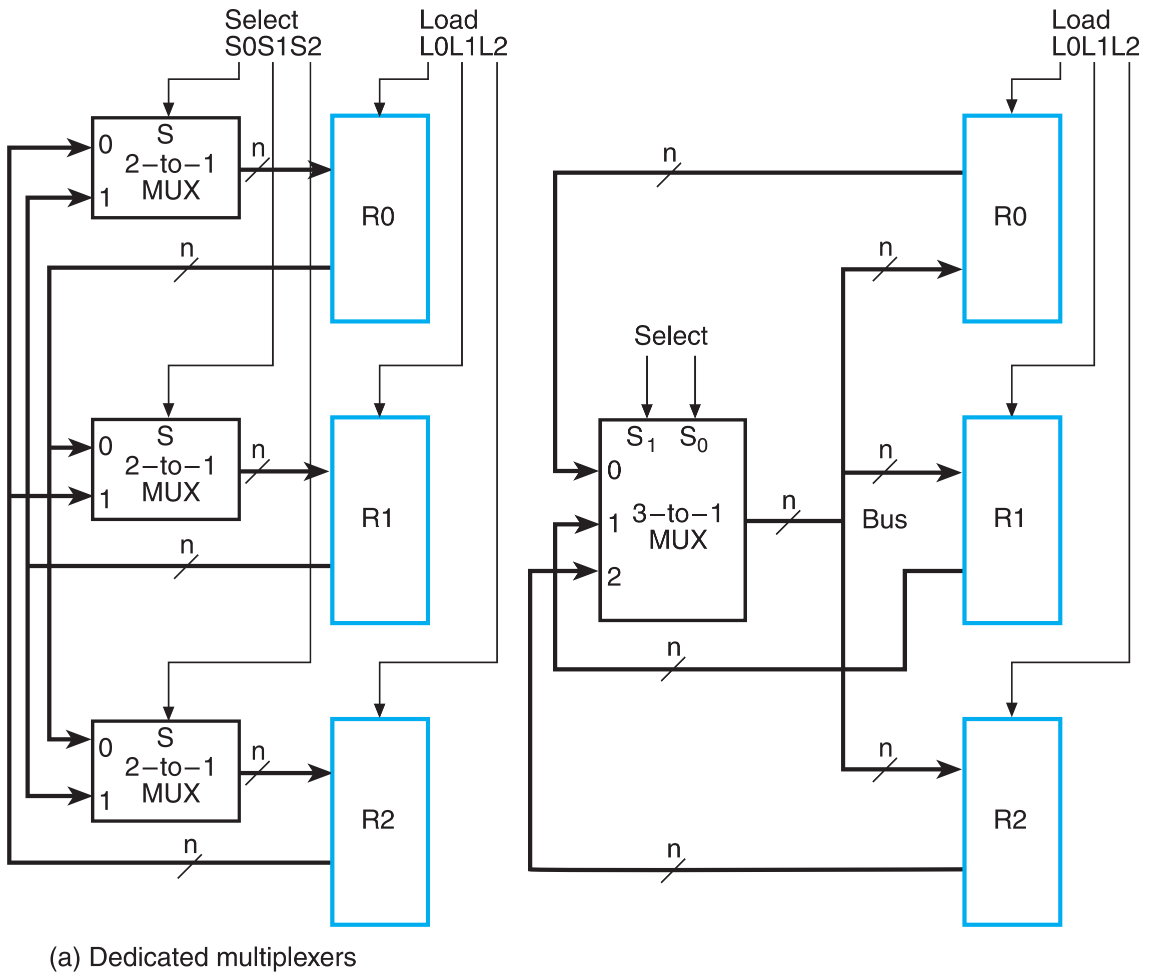
Type	Symbolic designation	Eight-bit examples: Source $R2$	Eight-bit examples: After shift: Destination $R1$
shift left	$R1 \leftarrow \mathrm{sl}\ R2$	10011110	00111100
shift right	$R1 \leftarrow \mathrm{sr}\ R2$	11100101	01110010

Table 7-5 Examples of Shifts



(b) Detailed logic

Fig. 7-5 Use of Multiplexers to Select between Two Registers



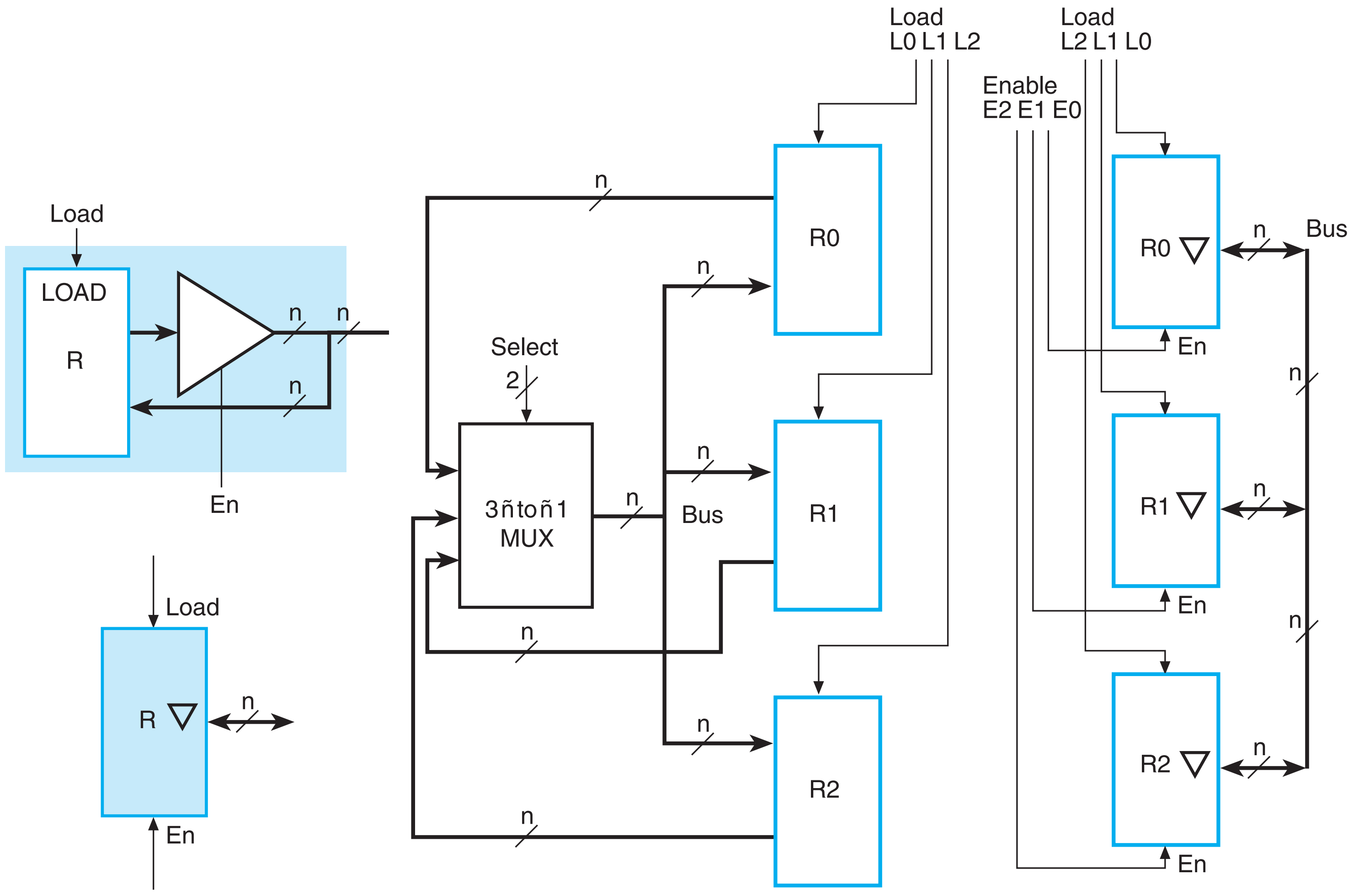
(b) Single Bus

Fig. 7-6 Single Bus versus Dedicated Multiplexers

TABLE 7-6
Examples of Register Transfers Using the Single Bus in Figure 7-6(b)

	Select		Load		
Register Transfer	**S1**	**S0**	**L2**	**L1**	**L0**
$R0 \leftarrow R2$	1	0	0	0	1
$R0 \leftarrow R1,\ R2 \leftarrow R1$	0	1	1	0	1
$R0 \leftarrow R1,\ R1 \leftarrow R0$	Impossible				

Table 7-6 Examples of Register Transfers Using the Single Bus in Figure 7-6(b)



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

(b) Multiplexer bus

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

Fig. 7-7 Three-State Bus versus Multiplexer Bus

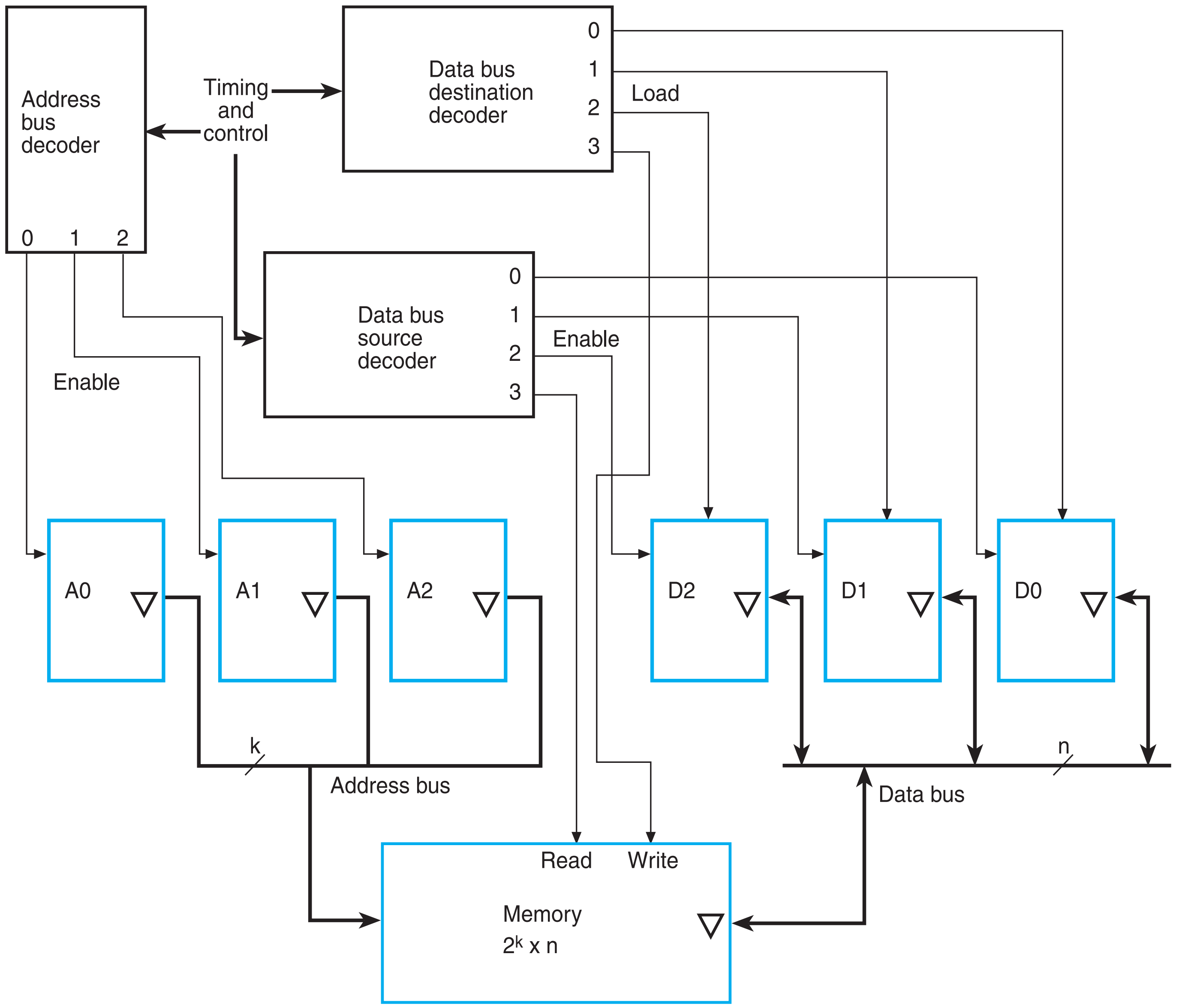


Fig. 7-8 Memory Unit Connected to Address and Data Buses

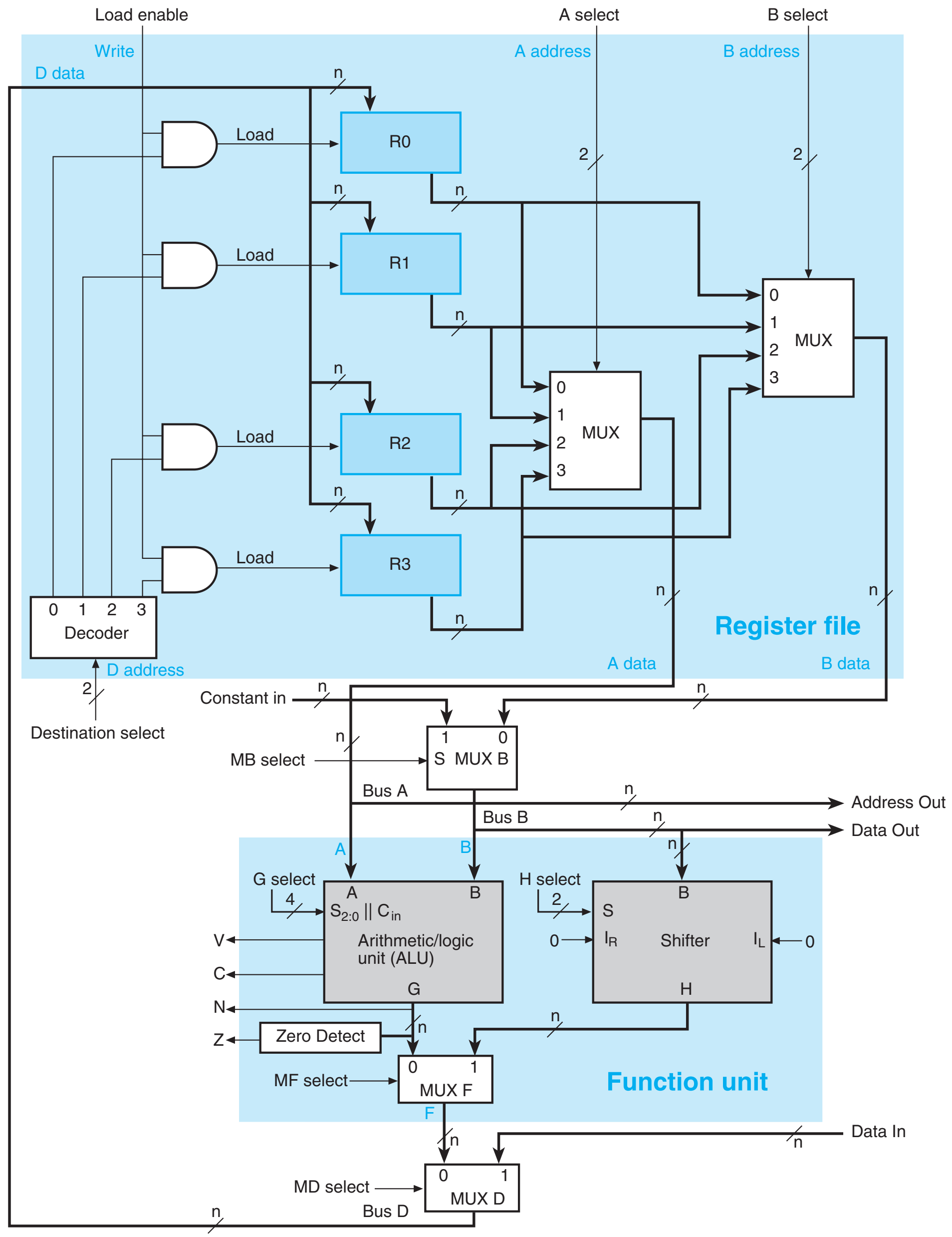


Fig. 7-9 Block Diagram of a Datapath

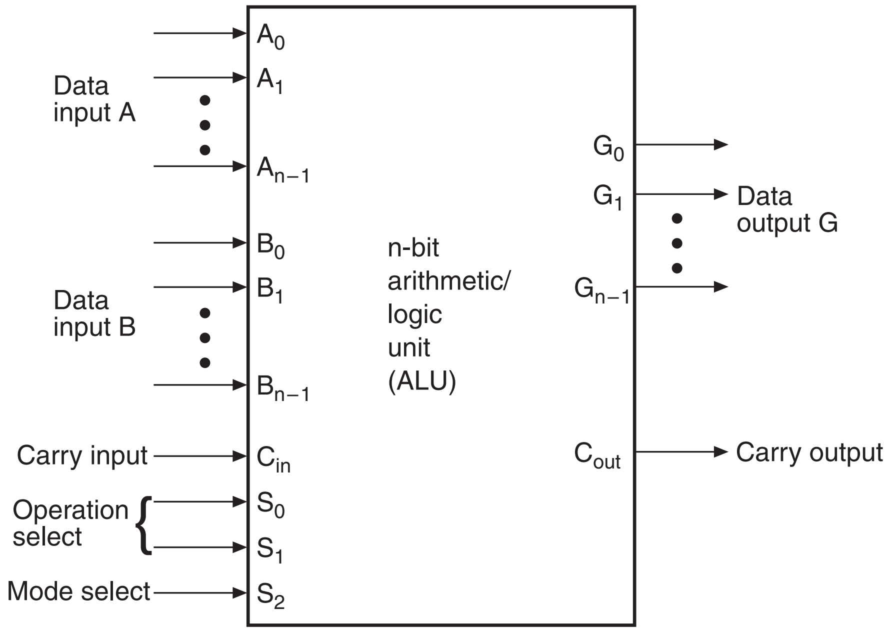


Fig. 7-10 Symbol for an *n*-Bit ALU

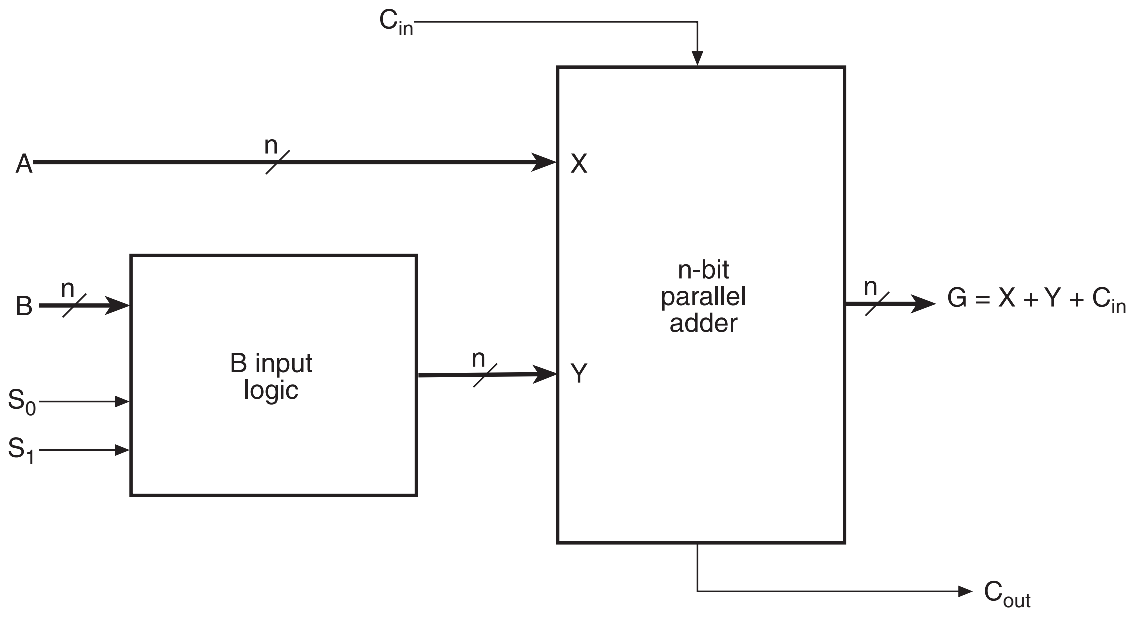


Fig. 7-11 Block Diagram of an Arithmetic Circuit

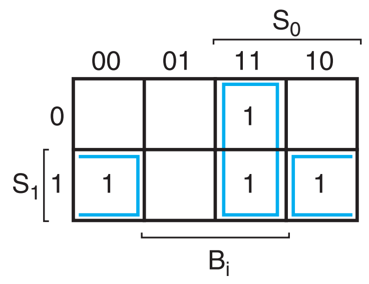
TABLE 7-7
Function Table for Arithmetic Circuit

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)

Table 7-7 Function Table for 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	$Y_i = \overline{B}_i$
1	0	1	0	
1	1	0	1	$Y_i = 1$
1	1	1	1	

(a) Truth table



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

Fig. 7-12 *B* Input Logic for One Stage of Arithmetic Circuit

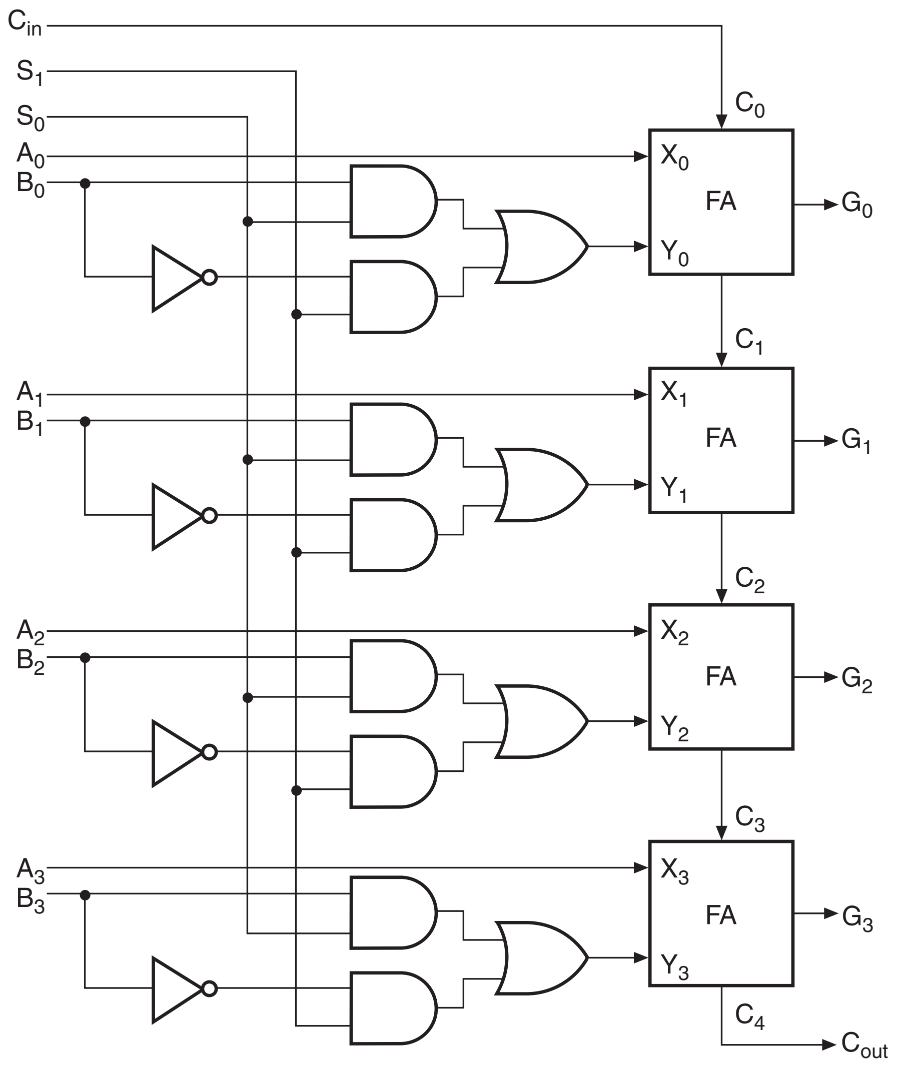
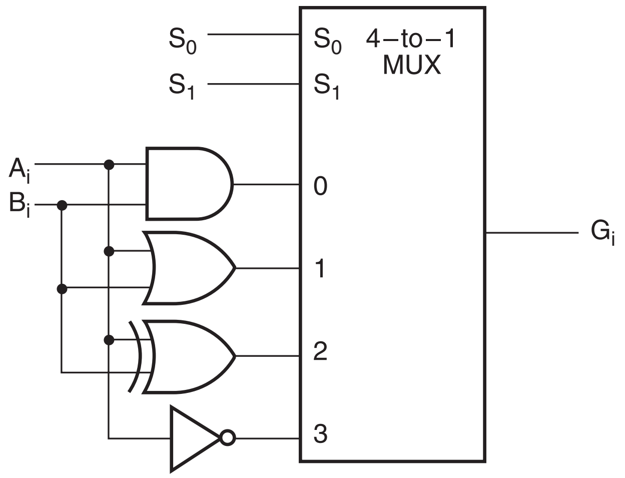


Fig. 7-13 Logic Diagram of a 4-Bit Arithmetic Circuit



(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

Fig. 7-14 One Stage of Logic Circuit

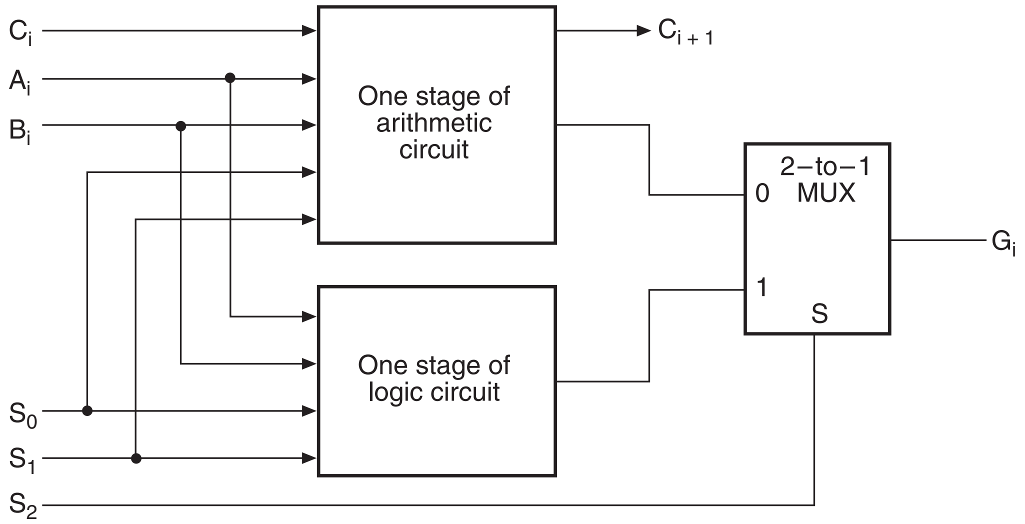


Fig. 7-15 One Stage of ALU

TABLE 7-8
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)

Table 7-8 Function Table for ALU

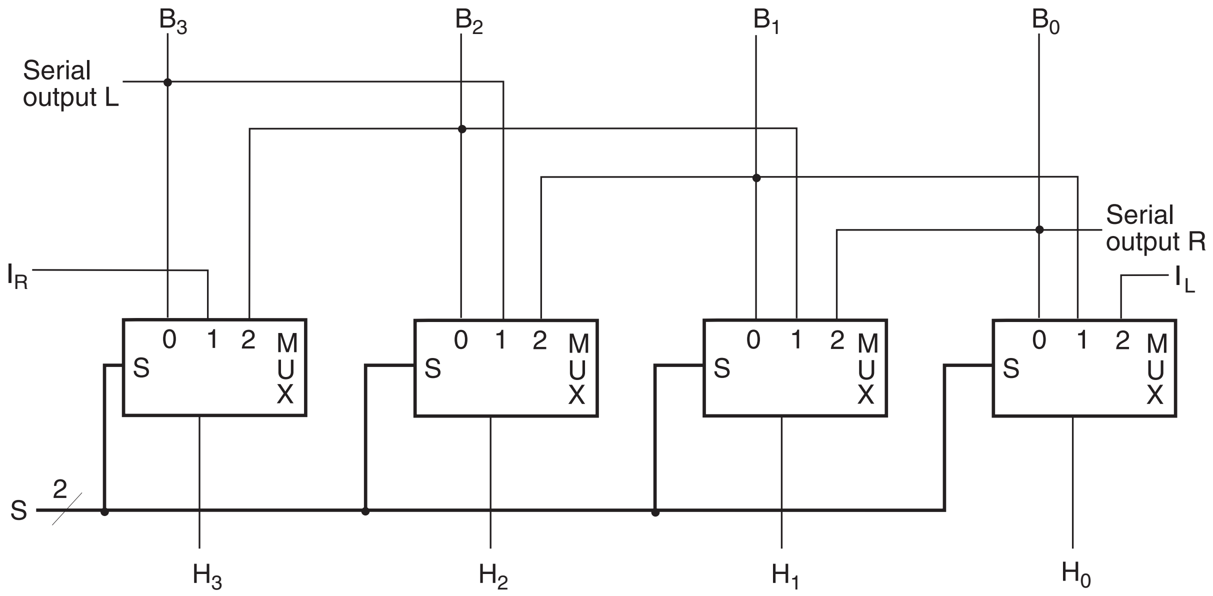


Fig. 7-16 4-Bit Basic Shifter

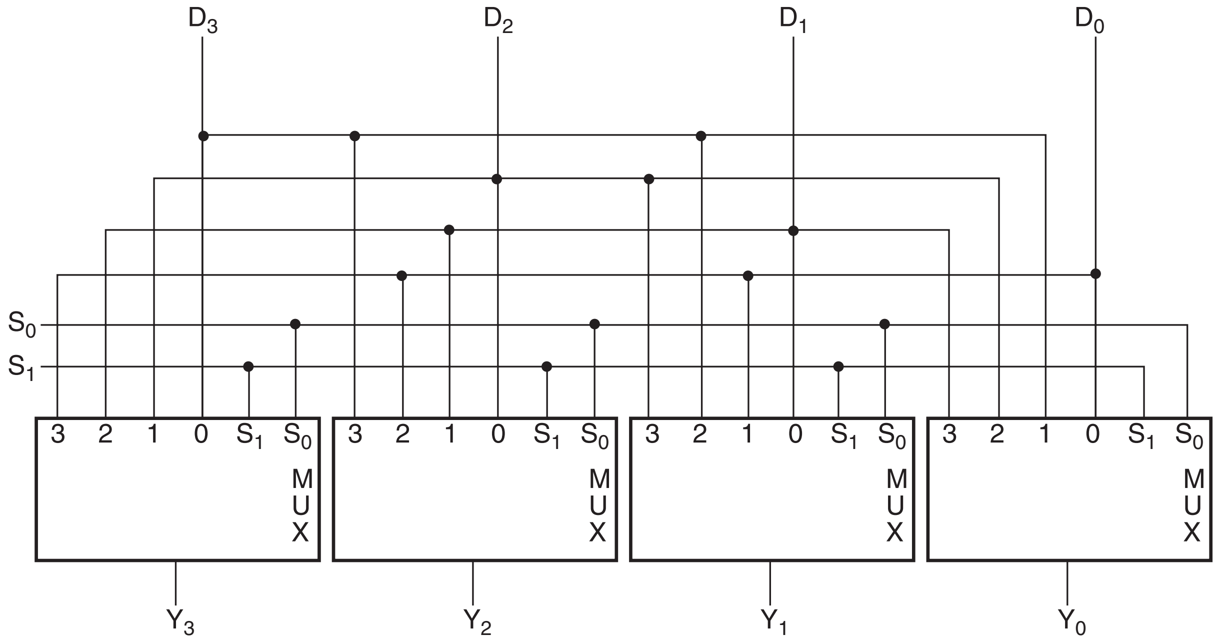


Fig. 7-17 4-Bit Barrel Shifter

TABLE 7-9
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

Table 7-9 Function Table for 4-Bit Barrel Shifter

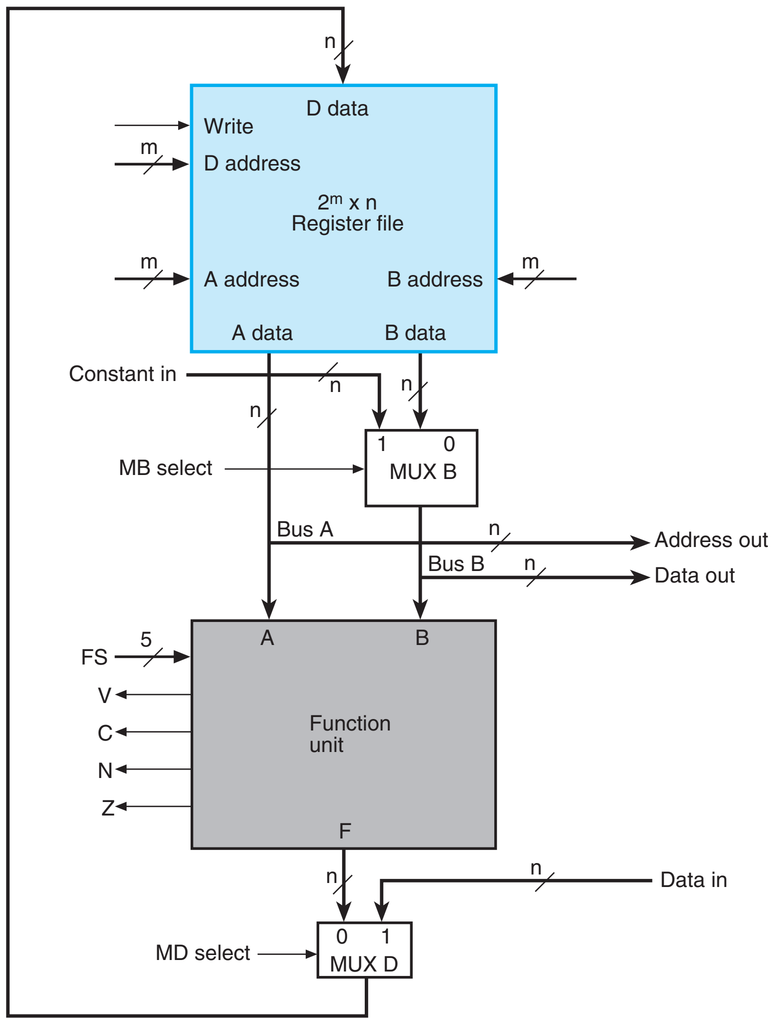
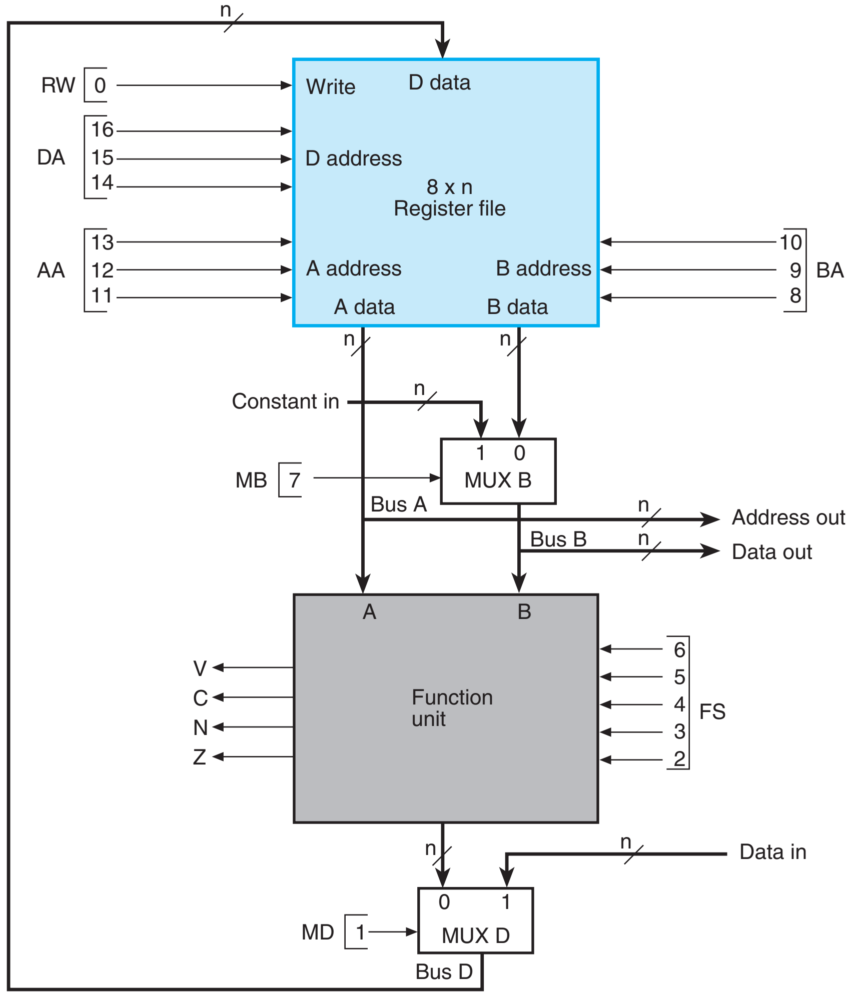


Fig. 7-18 Block Diagram of Datapath Using the Register File and Function Unit

TABLE 7-10
***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	00	$F = A$
00001	0	0001	00	$F = A + 1$
00010	0	0010	00	$F = A + B$
00011	0	0011	00	$F = A + B + 1$
00100	0	0100	01	$F = A + \overline{B}$
00101	0	0101	01	$F = A + \overline{B} + 1$
00110	0	0110	01	$F = A - 1$
00111	0	0111	01	$F = A$
01000	0	1000	0	$F = A \wedge B$
01010	0	1010	10	$F = A \vee B$
01100	0	1100	10	$F = A \oplus B$
01110	0	1110	10	$F = \overline{A}$
10000	1	0000	00	$F = B$
10100	1	0100	01	$F = \text{sr } B$
11000	1	1000	10	$F = \text{sl } B$

Table 7-10 *G* Select, *H* Select, and *MF* Select Codes Defined in Terms of *FS* Codes



(a) Block Diagram

16 15 14	13 12 11	10 9 8	7	6 5 4 3 2	1	0
DA	AA	BA	MB	FS	MD	RW

(b) Control word

Fig. 7-19 Datapath with Control Variables

TABLE 7-11
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			$F = A + \overline{B}\ \ 1$	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				
				$F = B$	10000				
				$F = \text{sr } B$	10100				
				$F = \text{sl } B$	11000				

Table 7-11 Encoding of Control Word for the Datapath

TABLE 7-12
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 B	Function	Write
$R7 \leftarrow R7 + 1$	$R7$	$R7$	—	Register	$F = A + 1$	Function	Write
$R1 \leftarrow R0 + 2$	$R1$	$R0$	—	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

Table 7-12 Examples of Microoperations for the Datapath, Using Symbolic Notation

TABLE 7-13
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	000	110	0	11000	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

Table 7-13 Examples of Microoperations from Table 7-11, Using Binary Control Words

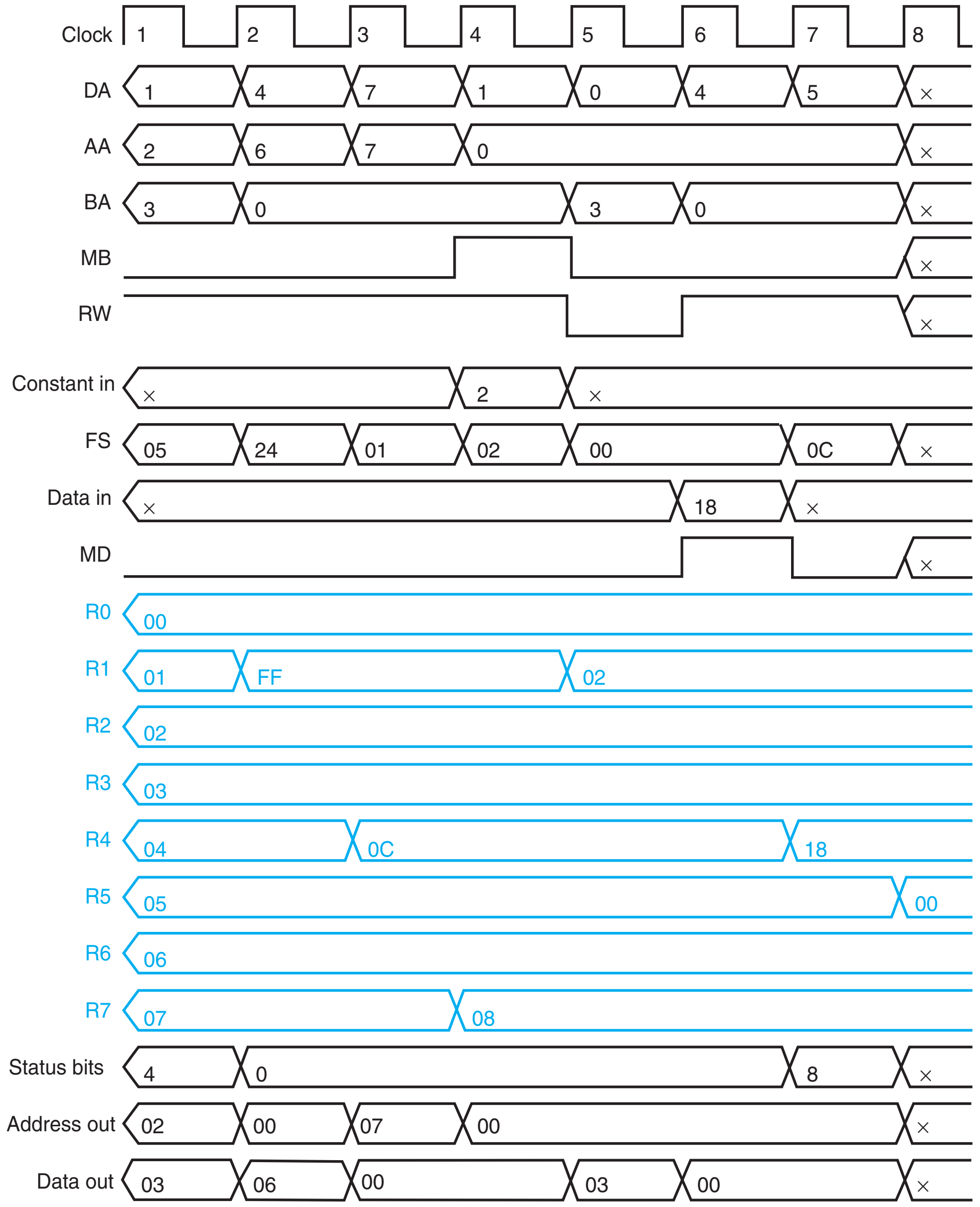


Fig. 7-20 Simulation of the Microoperation Sequence in Table 7-13

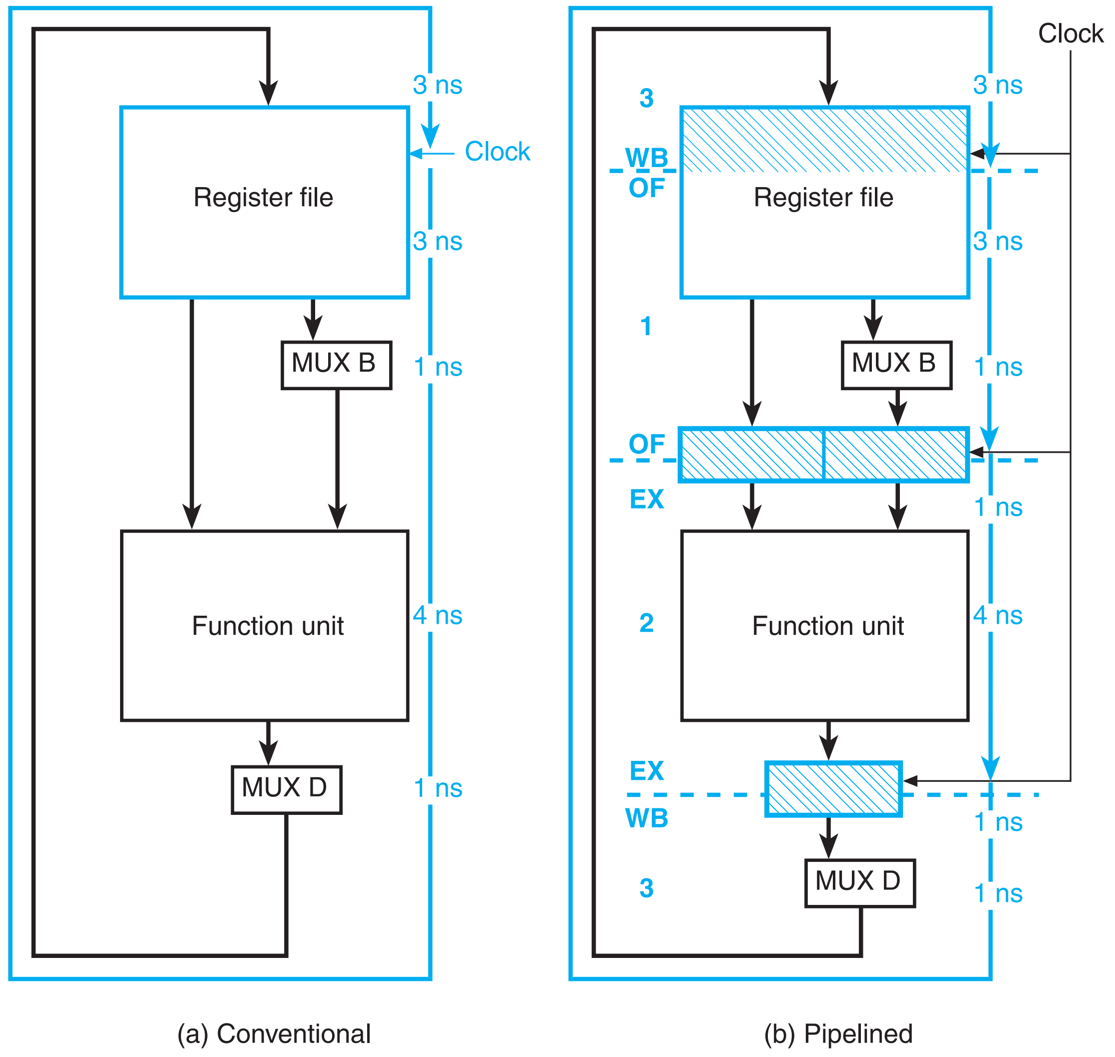


Fig. 7-21 Datapath Timing

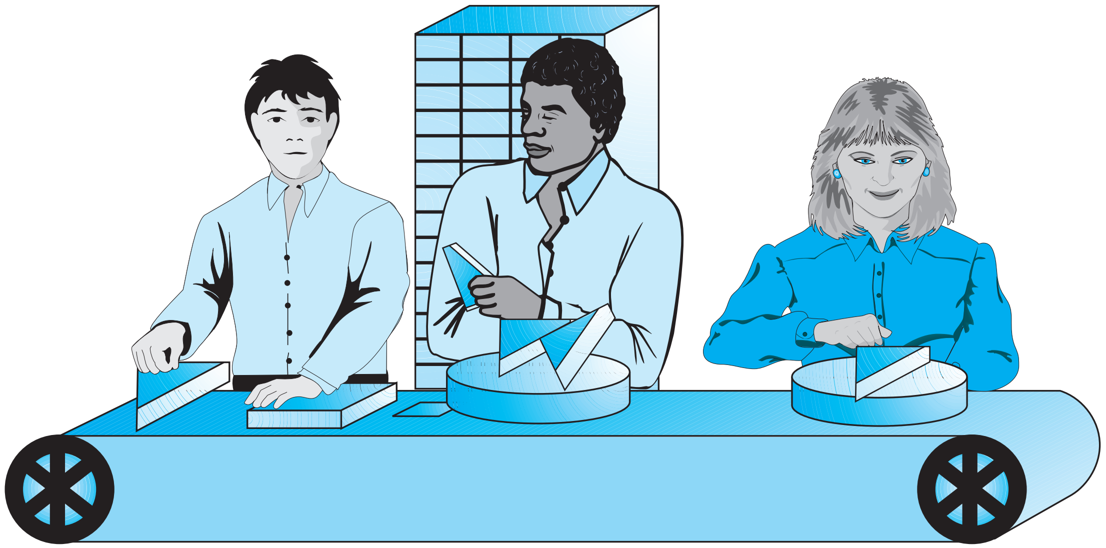


Fig. 7-22 Assembly Line Analogy to Datapath Pipeline

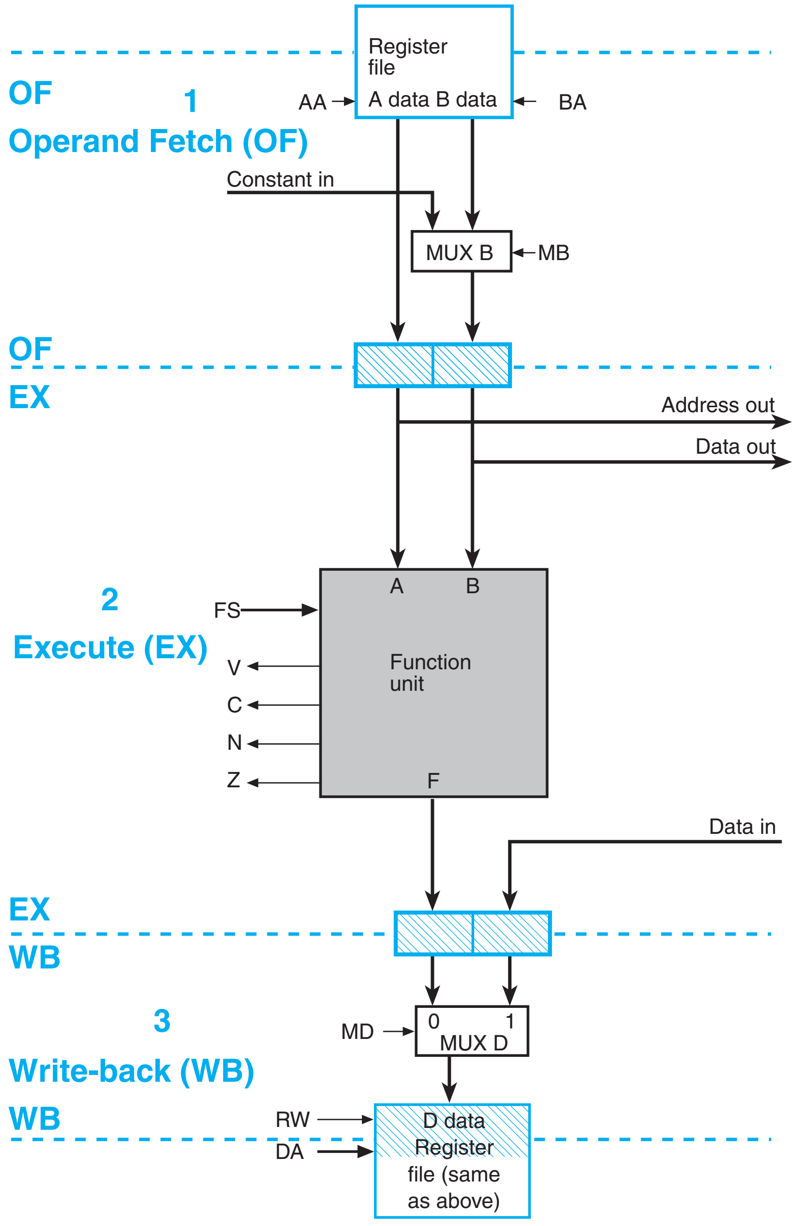


Fig. 7-23 Block Diagram of Pipelined Datapath

Clock cycle

		1	2	3	4	5	6	7	8	9
R1 ← R2 – R3	1	OF	EX	WB						
R4 ← sl R6	2		OF	EX	WB					
R7 ← R7 + 1	3			OF	EX	WB				
R1 ← R0 + 2	4				OF	EX	WB			
Data out ← R3	5					OF	EX	WB		
R4 ← Data in	6						OF	EX	WB	
R5 ← 0	7							OF	EX	WB

Microoperation

Fig. 7-24 Pipeline Execution Pattern for Microoperation Sequence in Table 7-13