IAA6007: Computer Architecture
Ch.4. Register Transfer & Micro-operations

Wooil Kim
Dept. of Computer Science & Engineering
Incheon National University

2019 Fall

Outline

- 1. Register transfer language
- 2. Register transfer
- 3. Bus and memory transfer
- 4. Arithmetic micro-operations
- 5. Logic micro-operations
- 6. Shift micro-operations
- 7. Arithmetic logic shift unit

4.1 Register transfer language

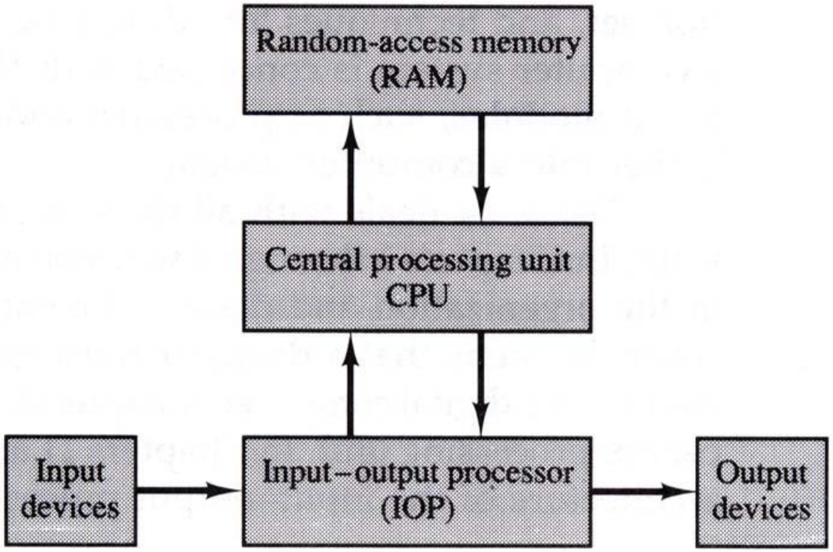


Figure 1-1 Block diagram of a digital computer.

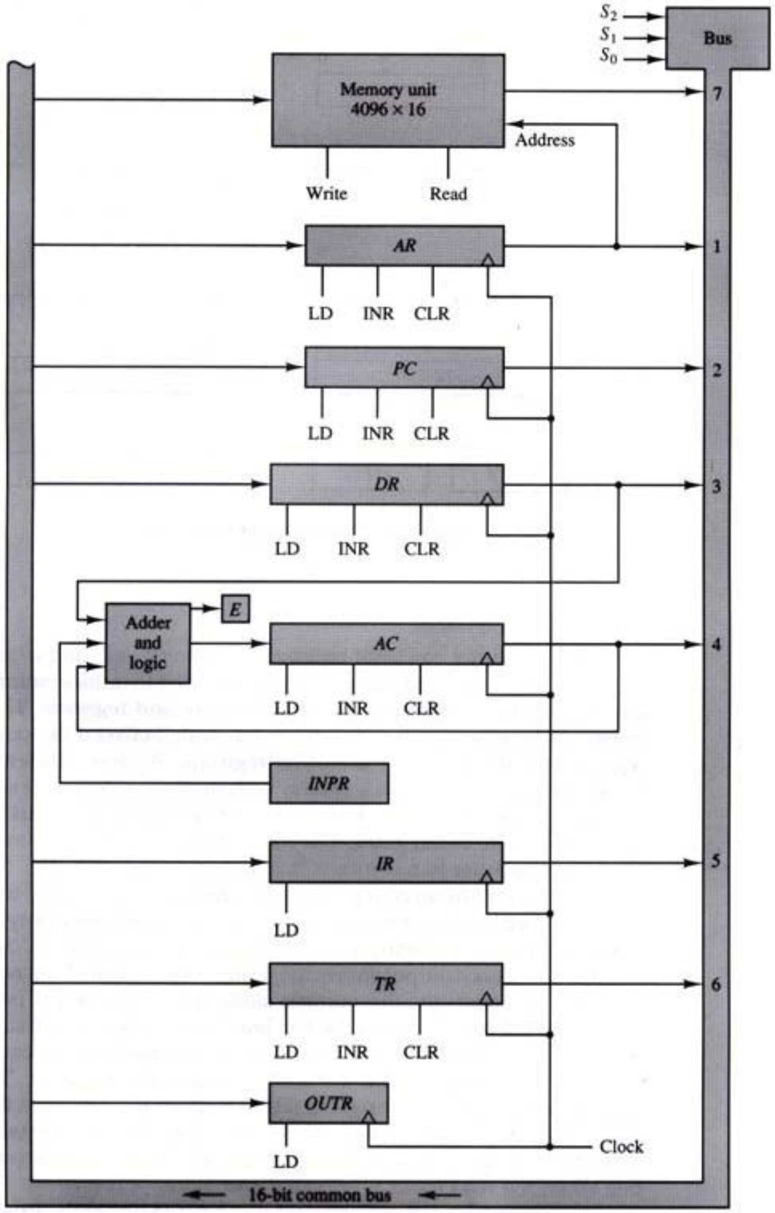


Figure 5-4 Basic computer registers connected to a common bus.

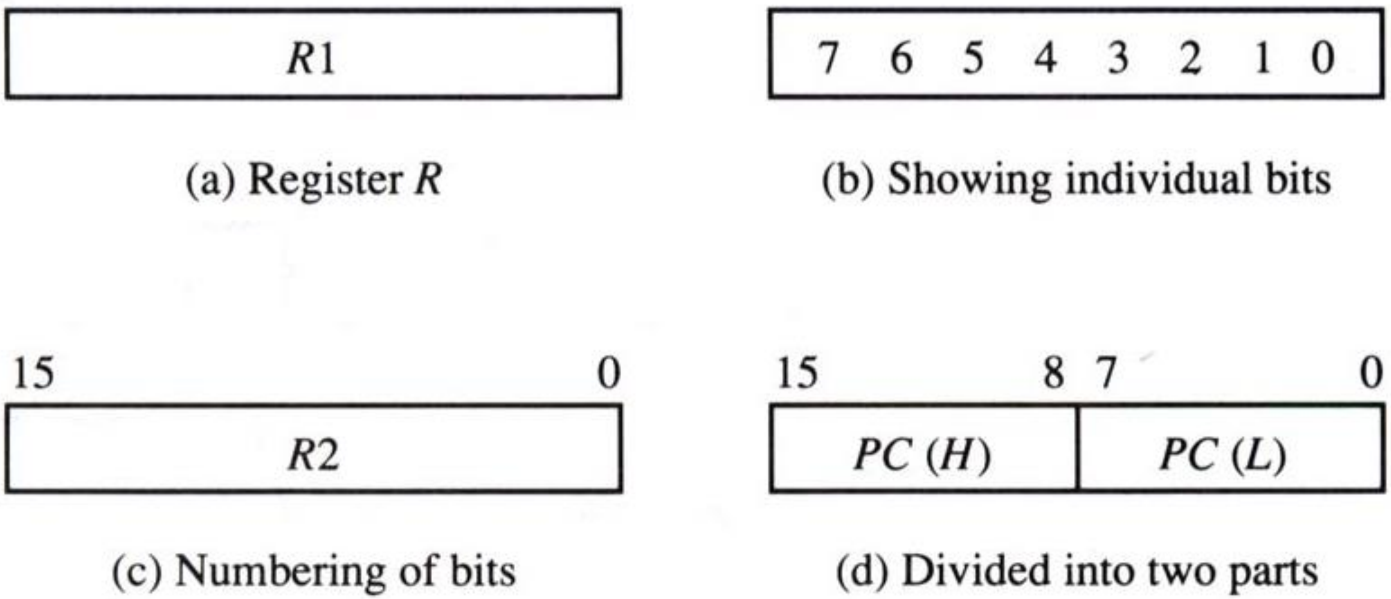
4.1 Register transfer language

- Micro-operations
  - Operations executed on data stored in registers
  - During a single clock pulse
  - Ex) shift, count, clear, load
- Register transfer language
  - Symbolic notation used to describe the micro-operation transfers among registers

4.2 Register transfer

- Representation of register

Figure 4-1 Block diagram of register.



(a) Register R

(b) Showing individual bits

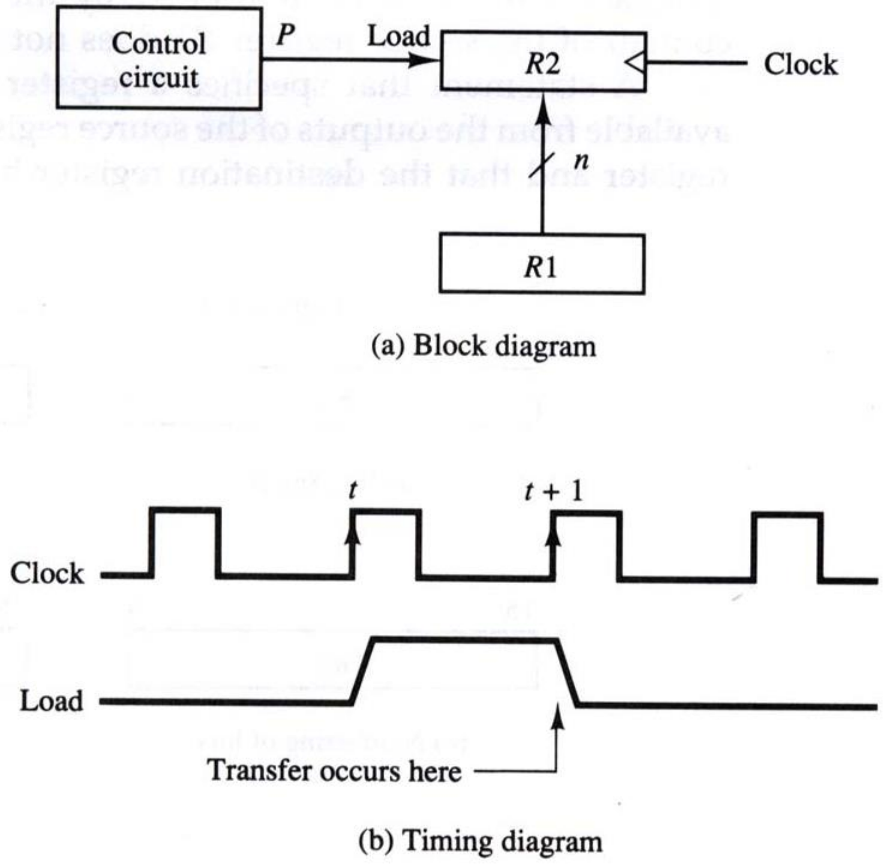
(c) Numbering of bits

(d) Divided into two parts

4.2 Register transfer

- Register transfer
  - Ex) if (P=1) then ($R_2 \leftarrow R_1$)
  - P: $R_2 \leftarrow R_1$
  - $\leftarrow$: register transfer language
  - Ex) T: $R_2 \leftarrow R_1$, $R_1 \leftarrow R_2$; simultaneous operations

Figure 4-2 Transfer from $R1$ to $R2$ when $P = 1$.



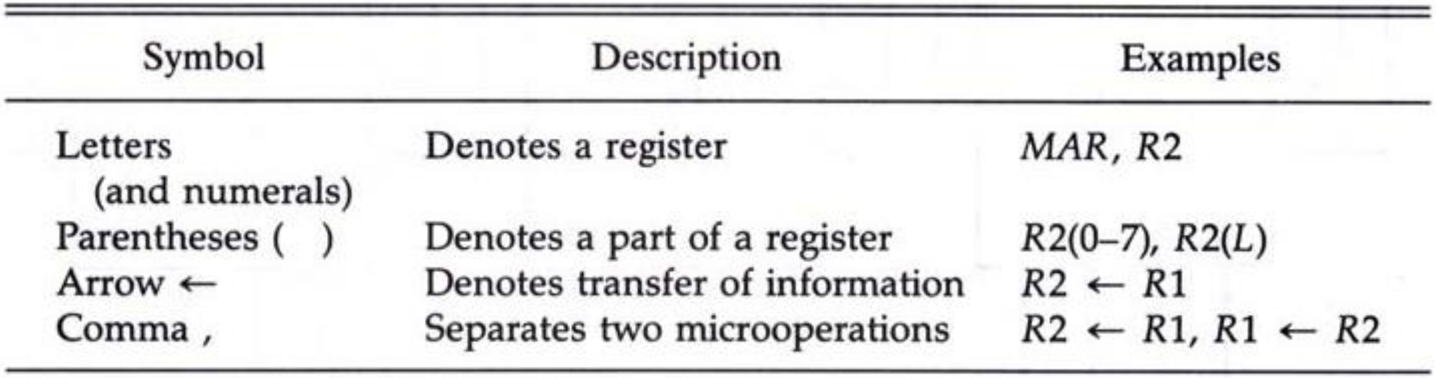
(a) Block diagram

(b) Timing diagram

4.2 Register transfer

- Register transfer

TABLE 4-1 Basic Symbols for Register Transfers



Symbol	Description	Examples
Letters (and numerals)	Denotes a register	*MAR*, *R*2
Parentheses ( )	Denotes a part of a register	*R*2(0–7), *R*2(*L*)
Arrow ←	Denotes transfer of information	*R*2 ← *R*1
Comma ,	Separates two microoperations	*R*2 ← *R*1, *R*1 ← *R*2

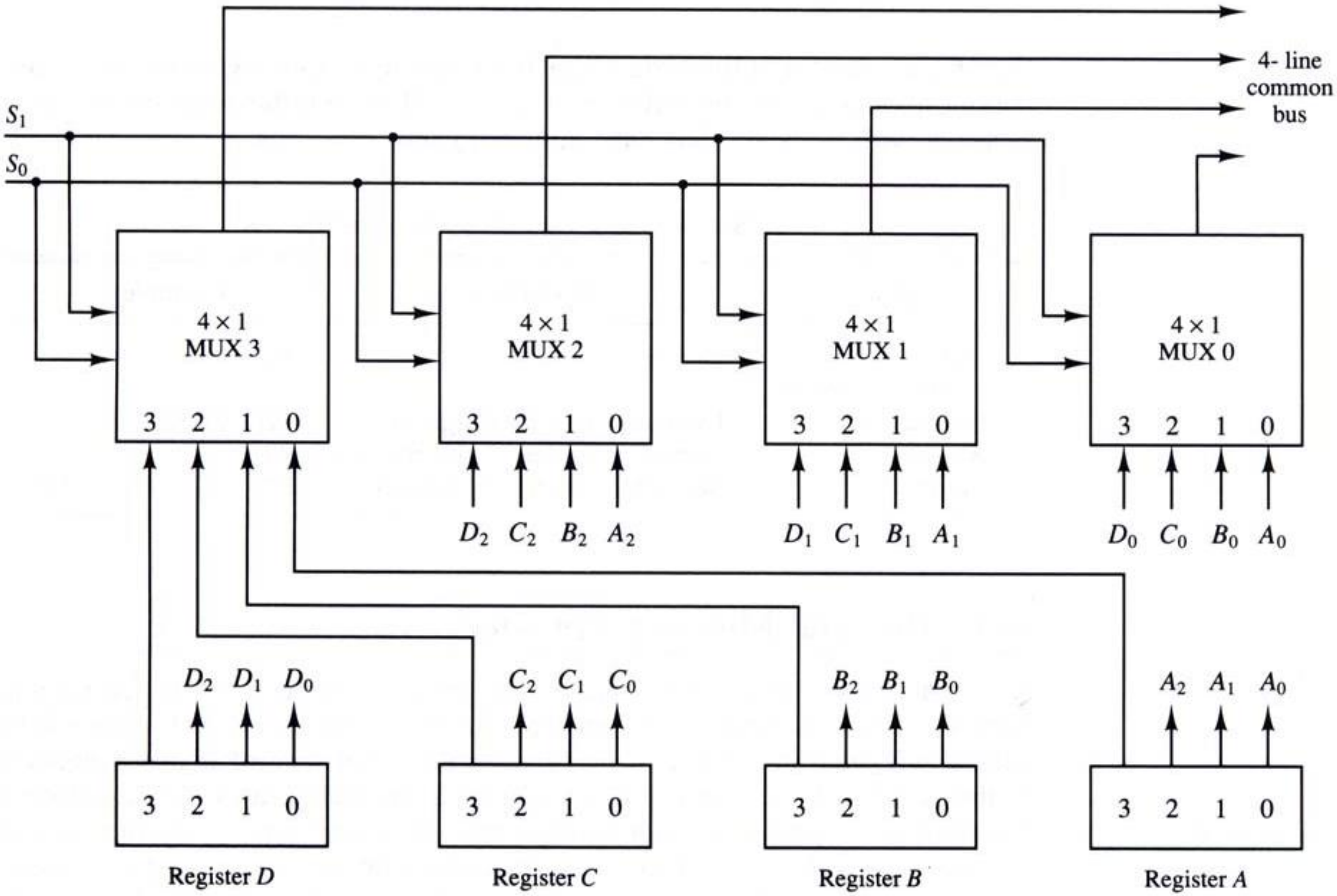
4.3 Bus and memory transfers

- Bus
  - A set of common lines
  - Efficient scheme for transferring information between registers in a multiple register configuration
  - When bus is included in the statement, the register transfer is symbolized as

$$BUS \leftarrow C,\ R_1 \leftarrow BUS \quad \text{or} \quad R_1 \leftarrow C$$

4.3 Bus and memory transfers

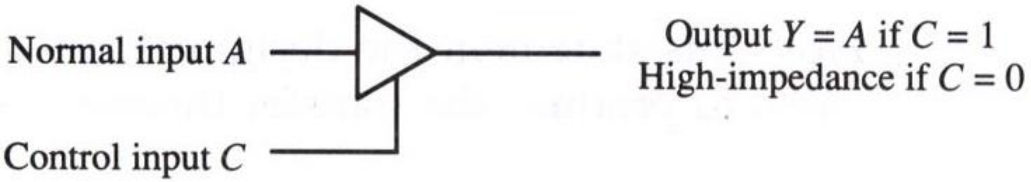
Figure 4-3 Bus system for four registers.



4.3 Bus and memory transfers

- Three-state bus buffer

Figure 4-4 Graphic symbols for three-state buffer.



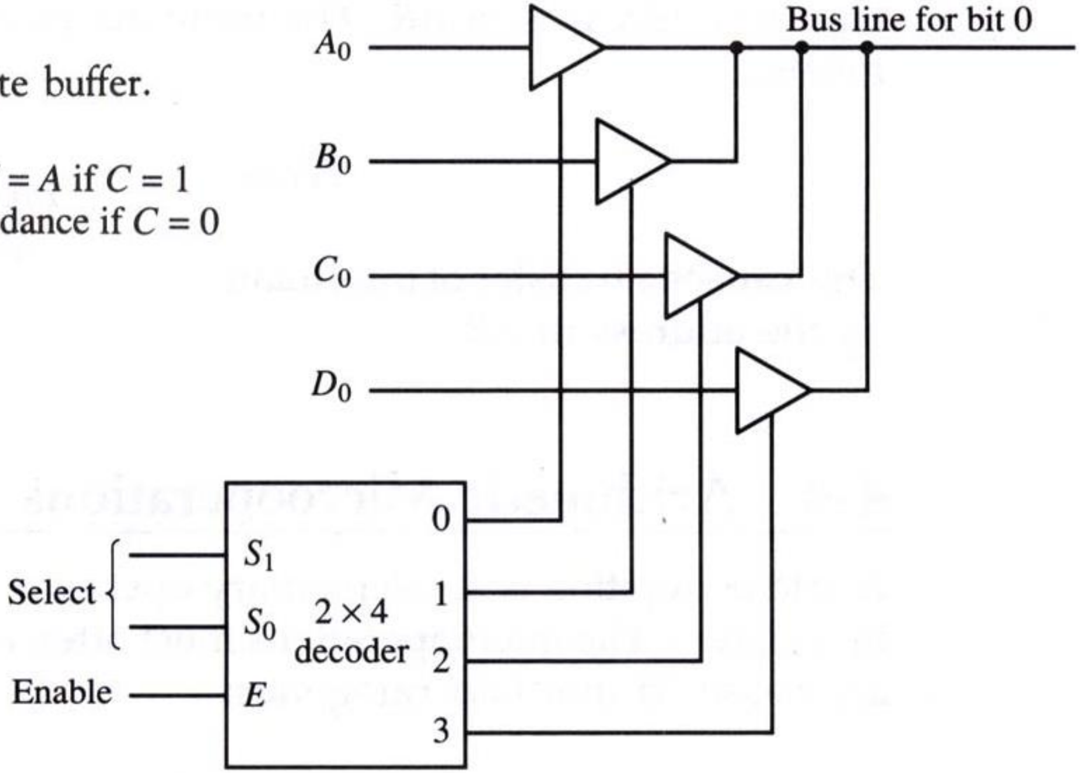


Figure 4-5 Bus line with three state-buffers.

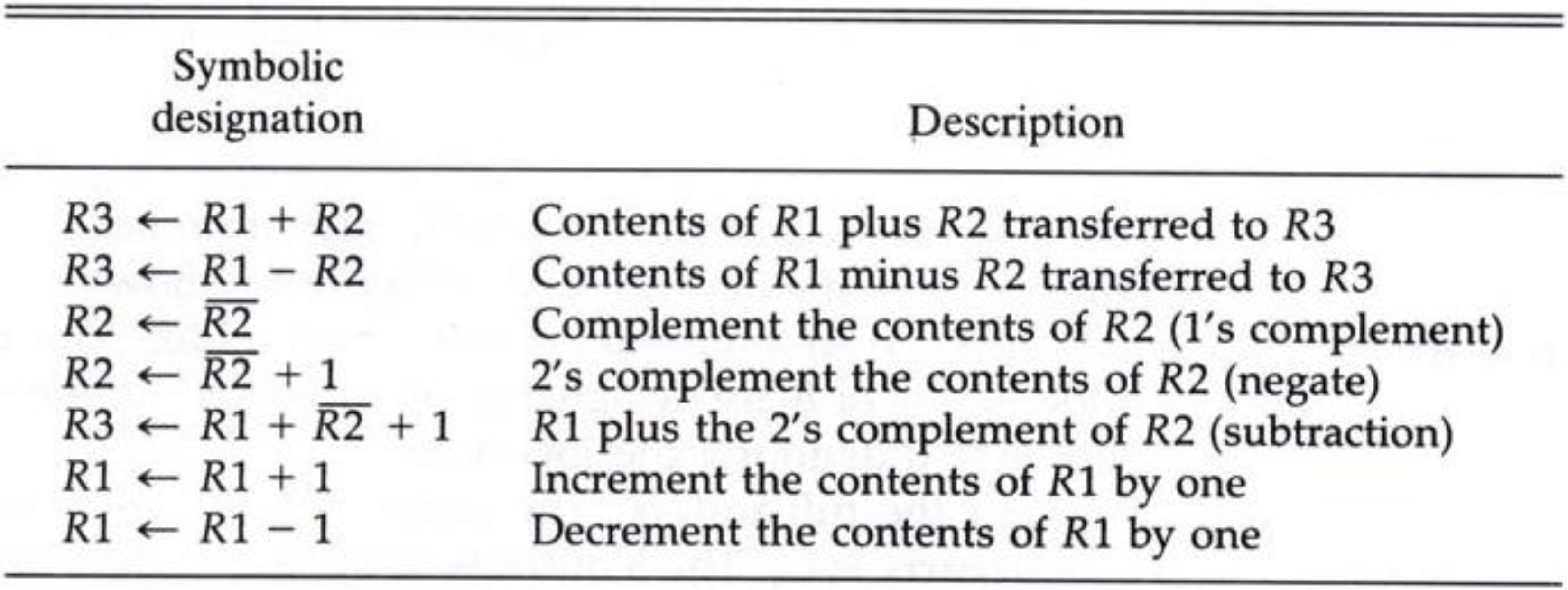
4.3 Bus and memory transfers

- Memory transfer
  - Address Register (AR) – register from which a memory unit receive the address
  - Data Register (DR) – register to which the data are transferred

$$\text{Read}: \ DR \leftarrow M[AR]$$
$$\text{Write}: M[AR] \leftarrow R_1$$

4.4 Arithmetic micro-operations

- Micro-operations
  - 1) Register transfer
  - 2) Arithmetic micro-operations
  - 3) Logic micro-operations
  - 4) shift micro-operations
- Arithmetic micro-operations



Symbolic designation	Description
$R3 \leftarrow R1 + R2$	Contents of $R1$ plus $R2$ transferred to $R3$
$R3 \leftarrow R1 - R2$	Contents of $R1$ minus $R2$ transferred to $R3$
$R2 \leftarrow \overline{R2}$	Complement the contents of $R2$ (1's complement)
$R2 \leftarrow \overline{R2} + 1$	2's complement the contents of $R2$ (negate)
$R3 \leftarrow R1 + \overline{R2} + 1$	$R1$ plus the 2's complement of $R2$ (subtraction)
$R1 \leftarrow R1 + 1$	Increment the contents of $R1$ by one
$R1 \leftarrow R1 - 1$	Decrement the contents of $R1$ by one

4.4 Arithmetic micro-operations

- Binary adder

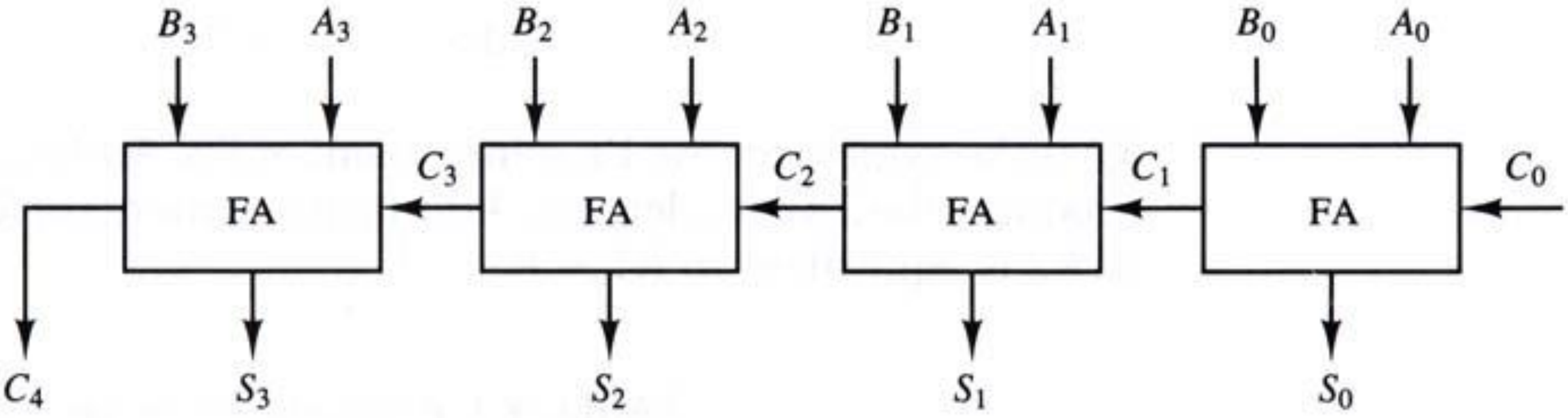


Figure 4-6 4-bit binary adder.

4.4 Arithmetic micro-operations

- Binary adder-subtractor

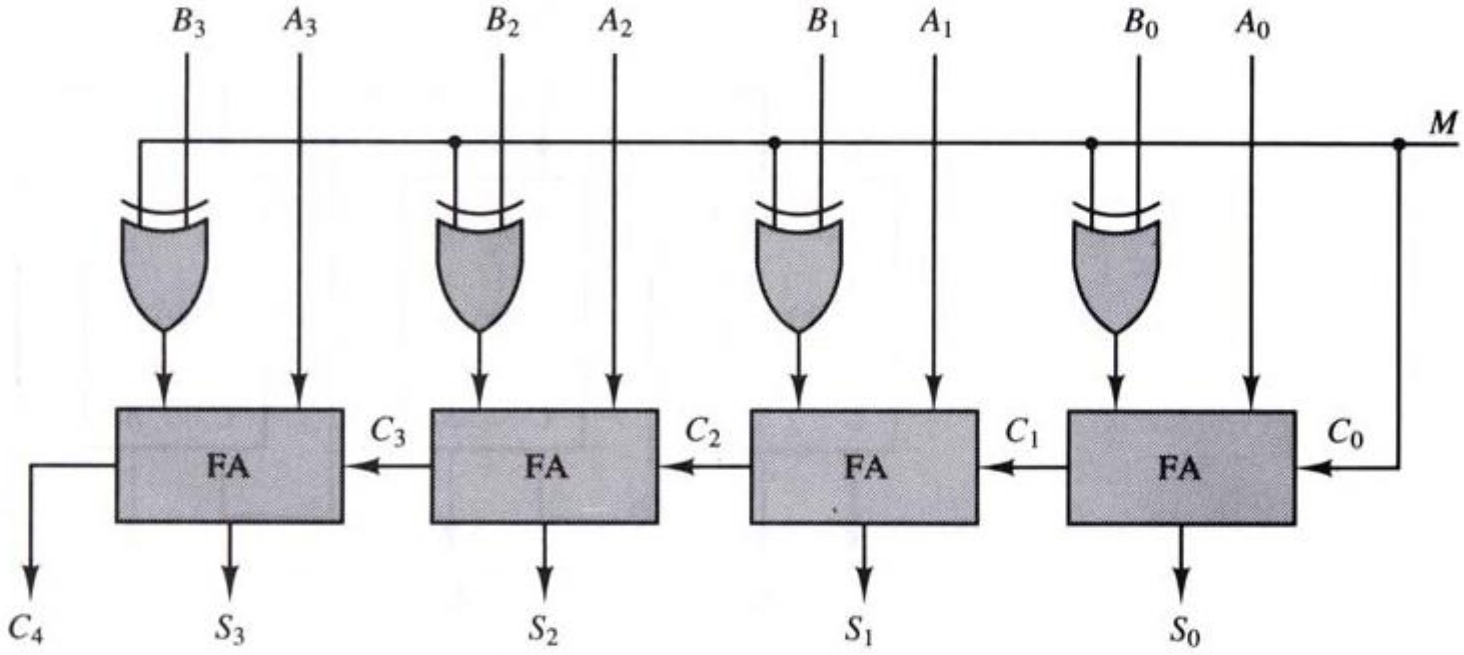


Figure 4-7 4-bit adder-subtractor.

- When M = 0, $C_0 = 0$ and $B_i \oplus 0 = B_i$ → addition
- When M = 1, $C_0 = 1$ and $B_i \oplus 1 = B_i'$ → A + 2's complement of B → subtraction

4.4 Arithmetic micro-operations

- Binary incrementer

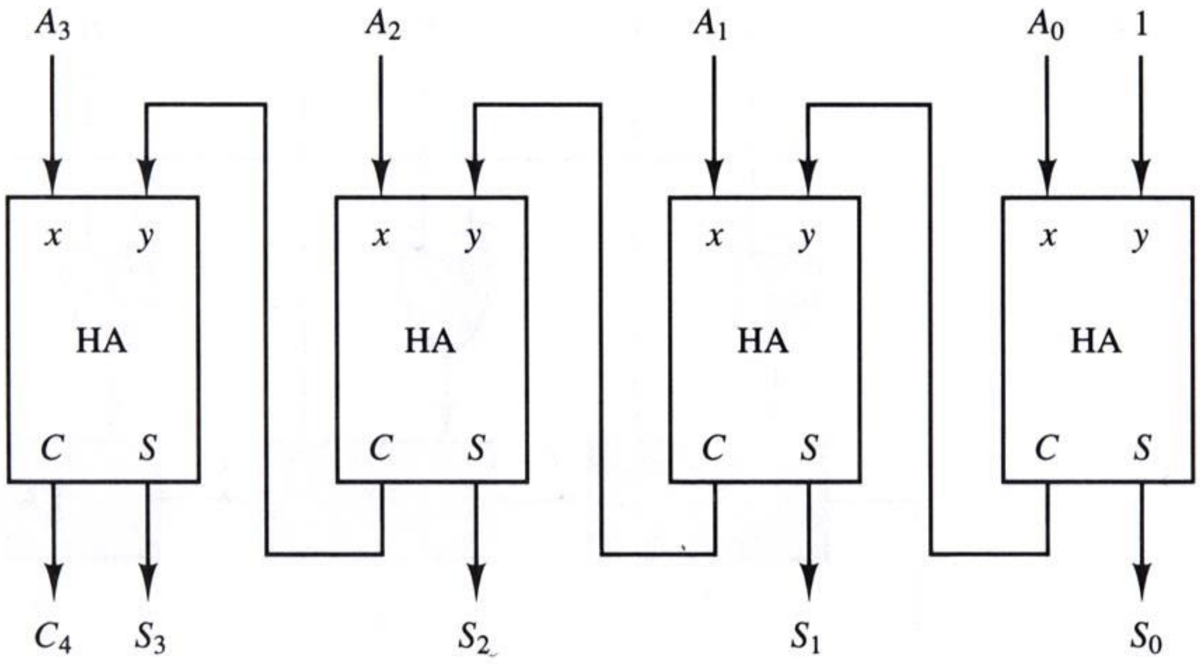


Figure 4-8 4-bit binary incrementer.

4.4 Arithmetic micro-operations

- Arithmetic circuit

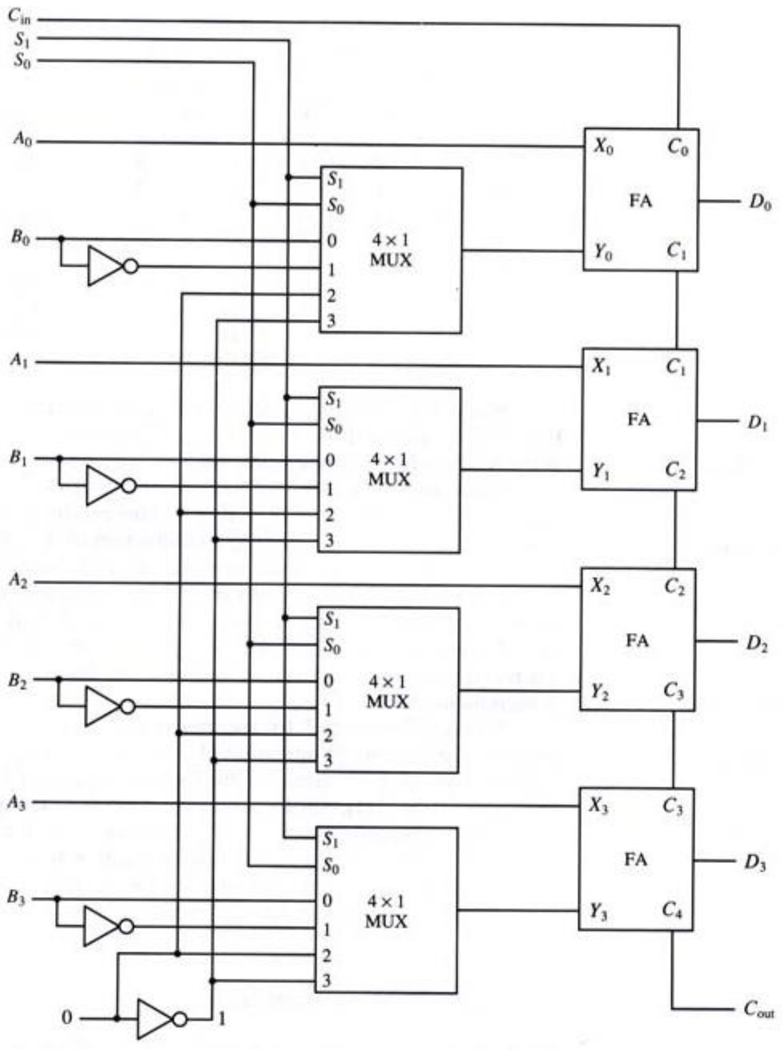
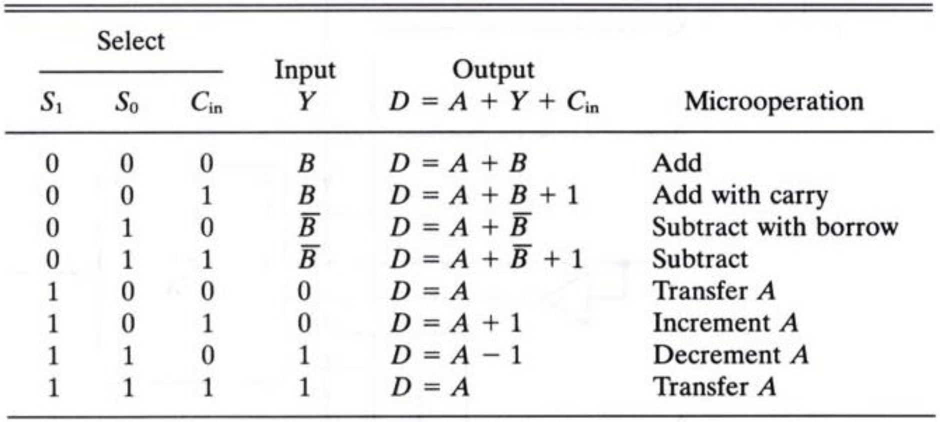


Figure 4-9 4-bit arithmetic circuit.

4.4 Arithmetic micro-operations

- Arithmetic circuit



Select			Input	Output	
S_1	S_0	C_{in}	Y	$D = A + Y + C_{in}$	Microoperation
0	0	0	B	$D = A + B$	Add
0	0	1	B	$D = A + B + 1$	Add with carry
0	1	0	$\overline{B}$	$D = A + \overline{B}$	Subtract with borrow
0	1	1	$\overline{B}$	$D = A + \overline{B} + 1$	Subtract
1	0	0	0	$D = A$	Transfer A
1	0	1	0	$D = A + 1$	Increment A
1	1	0	1	$D = A - 1$	Decrement A
1	1	1	1	$D = A$	Transfer A

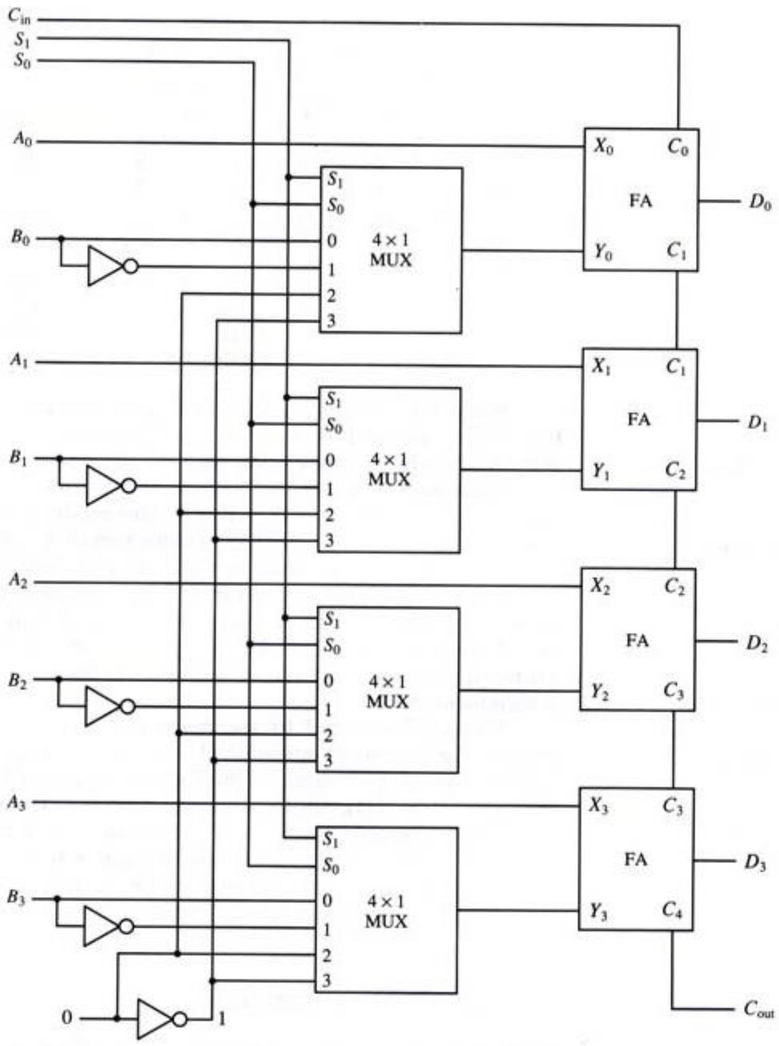


Figure 4-9 4-bit arithmetic circuit.

4.5 Logic micro-operations

- Ex) P: $R_1 \leftarrow R_1 \oplus R_2$, $R_1 = 1010$, $R_2 = 1100$
  - $R_1 = 0110$ after P =1
- Special symbols
  - ^, v will be used to denote AND and OR micro-operations
  - + denotes arithmetic plus in a micro-operations
  - + denotes OR operation in a control function
  - Ex) $P + Q : R_1 \leftarrow R_2 + R_3,\ R_4 \leftarrow R_5 \vee R_6$
- There are 16 different binary logic operations

4.5 Logic micro-operations

- 16 different binary logic operations

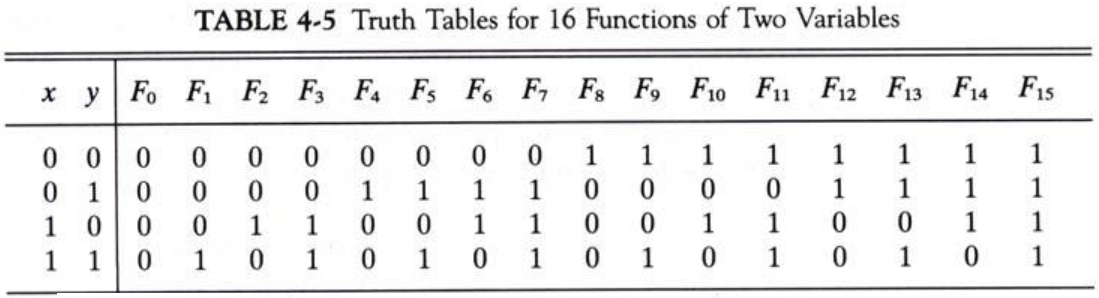


TABLE 4-5 Truth Tables for 16 Functions of Two Variables

x	y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

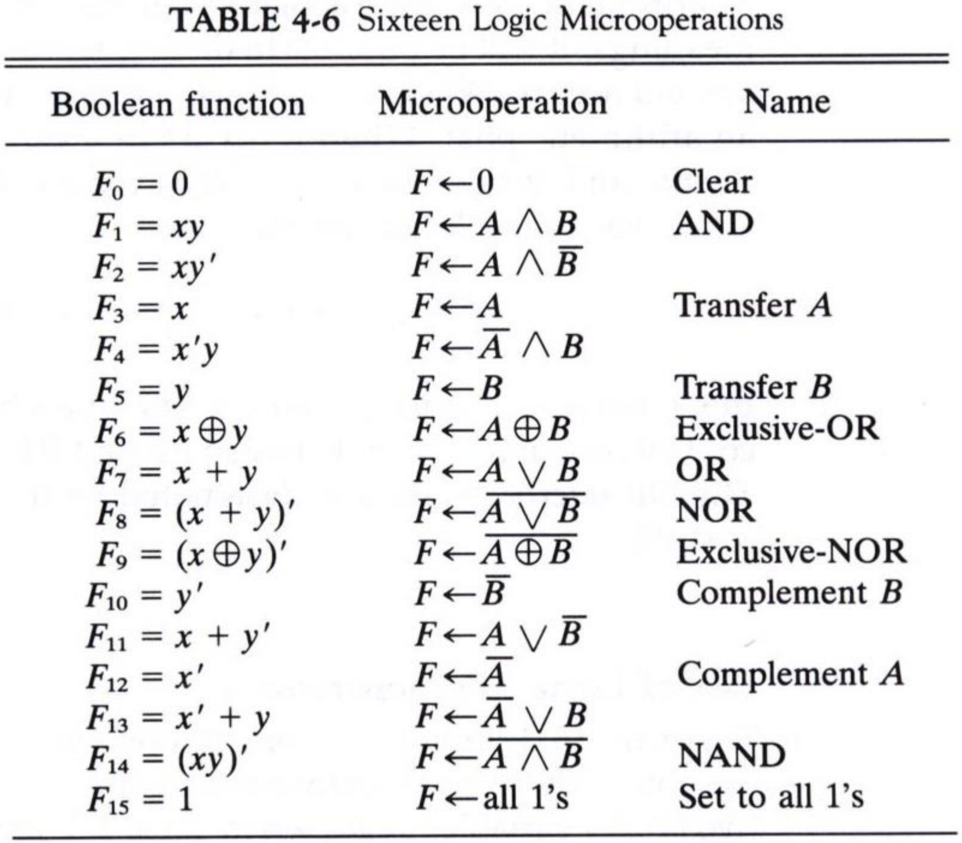


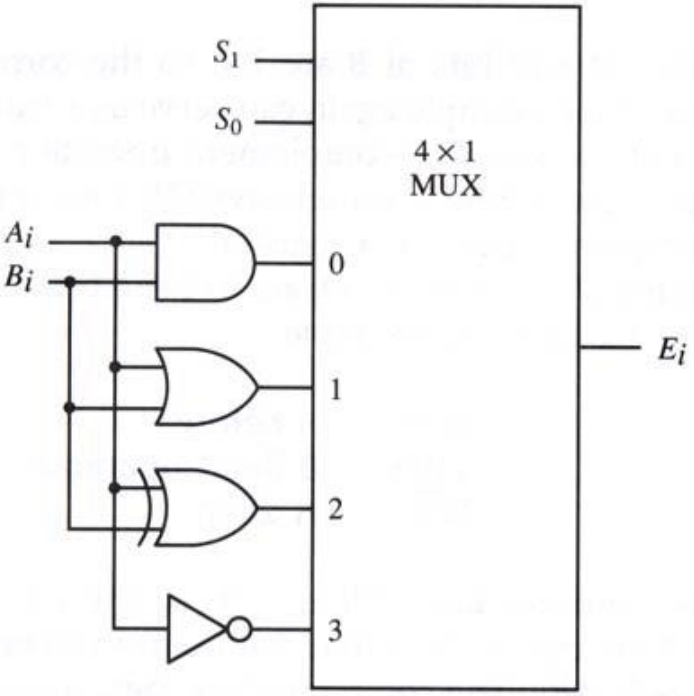
TABLE 4-6 Sixteen Logic Microoperations

Boolean function	Microoperation	Name
$F_0 = 0$	$F \leftarrow 0$	Clear
$F_1 = xy$	$F \leftarrow A \wedge B$	AND
$F_2 = xy'$	$F \leftarrow A \wedge \overline{B}$	
$F_3 = x$	$F \leftarrow A$	Transfer A
$F_4 = x'y$	$F \leftarrow \overline{A} \wedge B$	
$F_5 = y$	$F \leftarrow B$	Transfer B
$F_6 = x \oplus y$	$F \leftarrow A \oplus B$	Exclusive-OR
$F_7 = x + y$	$F \leftarrow A \vee B$	OR
$F_8 = (x + y)'$	$F \leftarrow \overline{A \vee B}$	NOR
$F_9 = (x \oplus y)'$	$F \leftarrow \overline{A \oplus B}$	Exclusive-NOR
$F_{10} = y'$	$F \leftarrow \overline{B}$	Complement B
$F_{11} = x + y'$	$F \leftarrow A \vee \overline{B}$	
$F_{12} = x'$	$F \leftarrow \overline{A}$	Complement A
$F_{13} = x' + y$	$F \leftarrow \overline{A} \vee B$	
$F_{14} = (xy)'$	$F \leftarrow \overline{A \wedge B}$	NAND
$F_{15} = 1$	$F \leftarrow$ all 1's	Set to all 1's

4.5 Logic micro-operations

- Hardware implementation
  - Using only 4 operations; AND, OR, XOR, & NOT(Complement)

Figure 4-10 One stage of logic circuit.



(a) Logic diagram

S_1	S_0	Output	Operation
0	0	$E = A \wedge B$	AND
0	1	$E = A \vee B$	OR
1	0	$E = A \oplus B$	XOR
1	1	$E = \bar{A}$	Complement

(b) Function table

4.5 Logic micro-operations

- Applications of logic micro-operations
  - Selective set
  - Selective complement
  - Selective clear
  - Mask operation
  - Insert operation
  - Clear operation

4.5 Logic micro-operations

- Applications of logic micro-operations
  - Selective set
    - $A \vee B$

1 0 1 0	A	before
1 1 0 0	B	
1 1 1 0	A	after

  - Selective complement
    - $A \oplus B$

1 0 1 0	A	before
1 1 0 0	B	
0 1 1 0	A	after

  - Selective clear
    - $A \wedge B'$

1 0 1 0	A	before
1 1 0 0	B	
0 0 1 0	A	after

4.5 Logic micro-operations

- Applications of logic micro-operations
  - Mask operation
    - A ∧ B

1 0 1 0	A	before
1 1 0 0	B	
1 0 0 0	A	after

  - Insert operation

1 0 1 0 1 0 1 0	A	before
0 0 0 0 1 1 1 1	B	mask
0 0 0 0 1 0 1 0	A	after

0 0 0 0 1 0 1 0	A	before
1 0 0 1 0 0 0 0	B	insert
1 0 0 1 1 0 1 0	A	after

  - Clear operation
    - A ⊕ B

1 0 1 0	A	before
1 0 1 0	B	
0 0 0 0	A	after

4.6 Shift micro-operations

- Logical shift – transfer 0 through the serial input
- Circular shift – circulates the bits around the two ends
- Arithmetic shift
  - Shift right – divide by 2 – leave the sign bit unchanged
  - Shift left – multiply by 2 0 insert 0 into R_0
  - Overflow condition: $V_s = R_{n-1} \oplus R_{n-2}$ before the shift

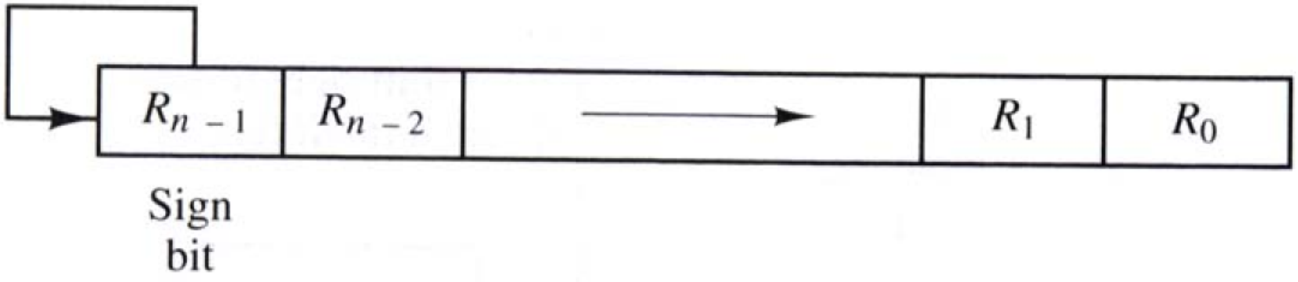


Figure 4-11 Arithmetic shift right.

4.6 Shift micro-operations

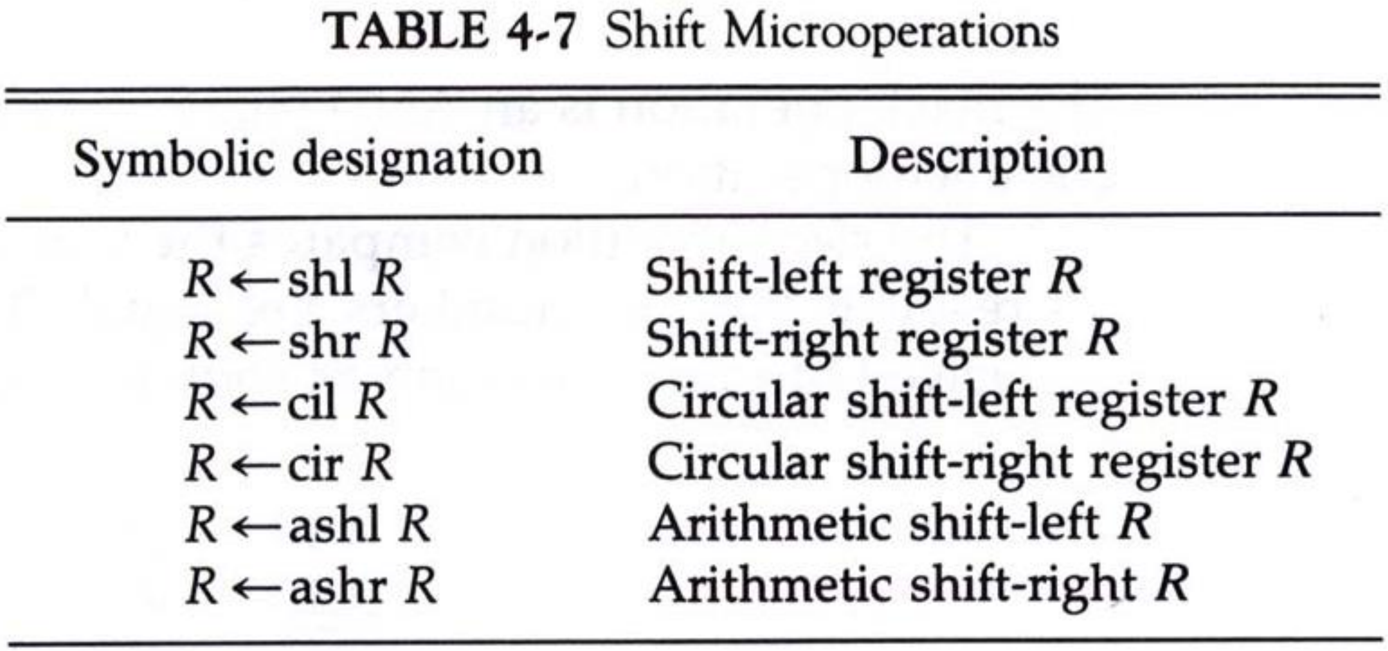
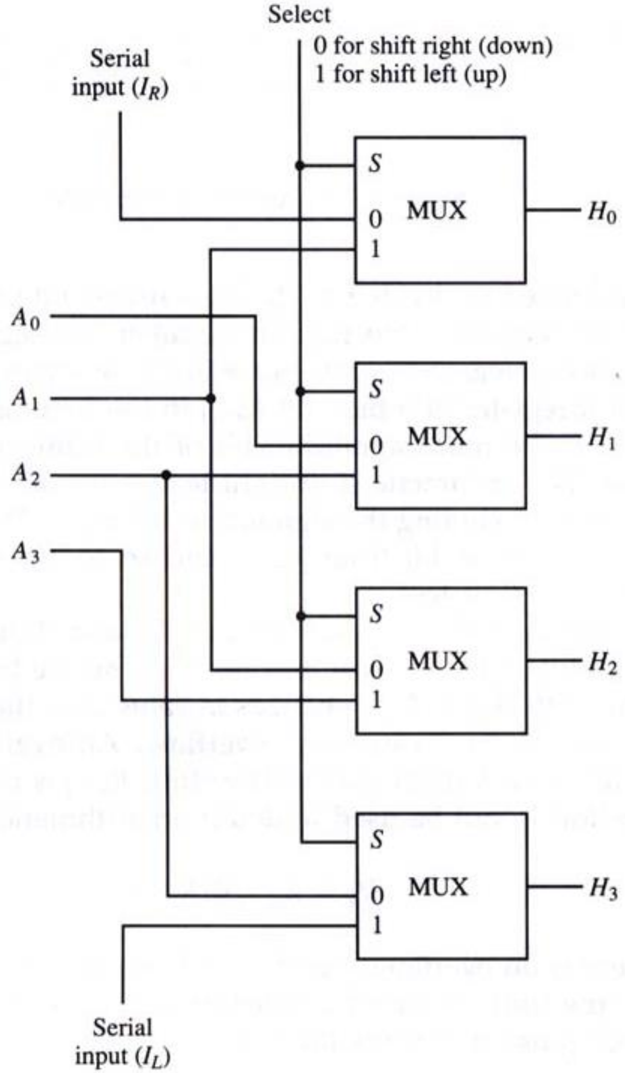


TABLE 4-7 Shift Microoperations

Symbolic designation	Description
$R \leftarrow \text{shl } R$	Shift-left register R
$R \leftarrow \text{shr } R$	Shift-right register R
$R \leftarrow \text{cil } R$	Circular shift-left register R
$R \leftarrow \text{cir } R$	Circular shift-right register R
$R \leftarrow \text{ashl } R$	Arithmetic shift-left R
$R \leftarrow \text{ashr } R$	Arithmetic shift-right R

4.6 Shift micro-operations

- Hardware implementation



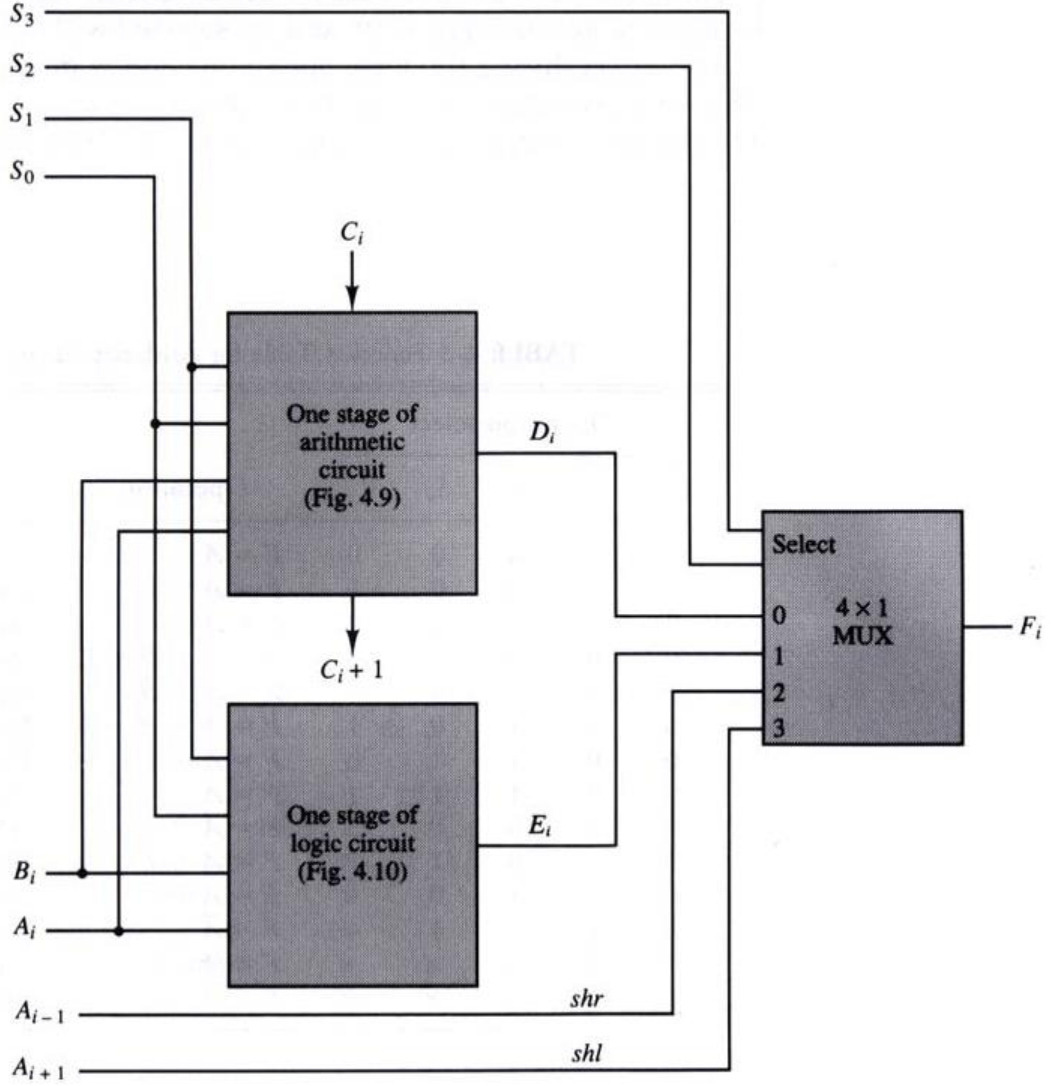
Function table

Select	Output			
S	H_0	H_1	H_2	H_3
0	I_R	A_0	A_1	A_2
1	A_1	A_2	A_3	I_L

Figure 4-12 4-bit combinational circuit shifter.

4.7 Arithmetic logic shift unit

Figure 4-13 One stage of arithmetic logic shift unit.



4.7 Arithmetic logic shift unit

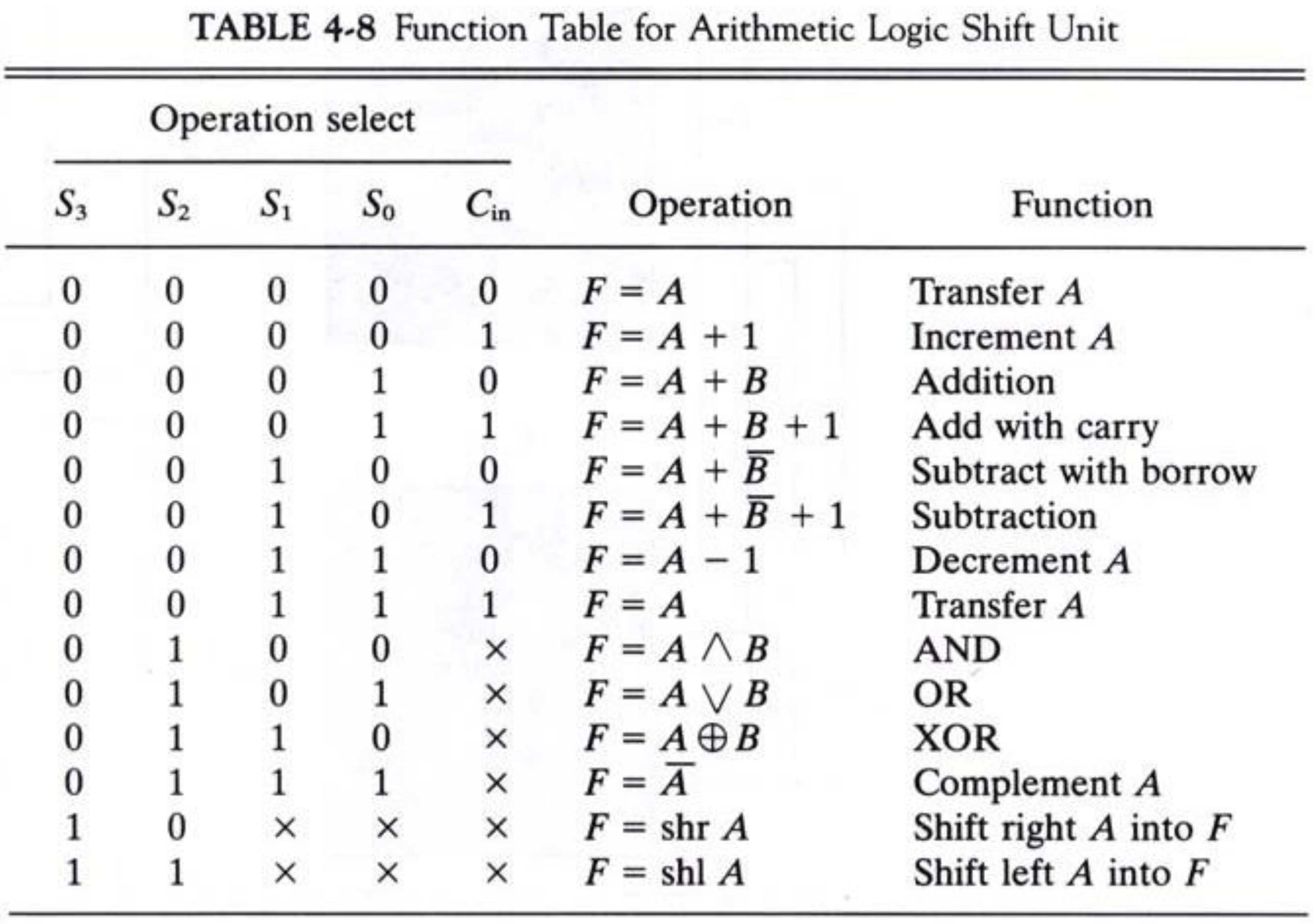


TABLE 4-8 Function Table for Arithmetic Logic Shift Unit

Operation select						
S_3	S_2	S_1	S_0	C_{in}	Operation	Function
0	0	0	0	0	$F = A$	Transfer A
0	0	0	0	1	$F = A + 1$	Increment A
0	0	0	1	0	$F = A + B$	Addition
0	0	0	1	1	$F = A + B + 1$	Add with carry
0	0	1	0	0	$F = A + \overline{B}$	Subtract with borrow
0	0	1	0	1	$F = A + \overline{B} + 1$	Subtraction
0	0	1	1	0	$F = A - 1$	Decrement A
0	0	1	1	1	$F = A$	Transfer A
0	1	0	0	×	$F = A \wedge B$	AND
0	1	0	1	×	$F = A \vee B$	OR
0	1	1	0	×	$F = A \oplus B$	XOR
0	1	1	1	×	$F = \overline{A}$	Complement A
1	0	×	×	×	$F = \text{shr } A$	Shift right A into F
1	1	×	×	×	$F = \text{shl } A$	Shift left A into F

Problems

- 4-2, 4-3, 4-4, 4-7, 4-8, 4-10,
- 4-12, 4-15, 4-19, 4-20, 4-21, 4-22