Processor Logic Design (contd…)

Status Register

- It is sometimes convenient to supplement the ALU with a status register where the status bit (overflow, zero indication, sign) conditions are stored for further analysis.

- Status-bit conditions are sometimes called condition-code bits or flag bits.

Setting bits in a status register

8-bit ALU with a 4-bit status register

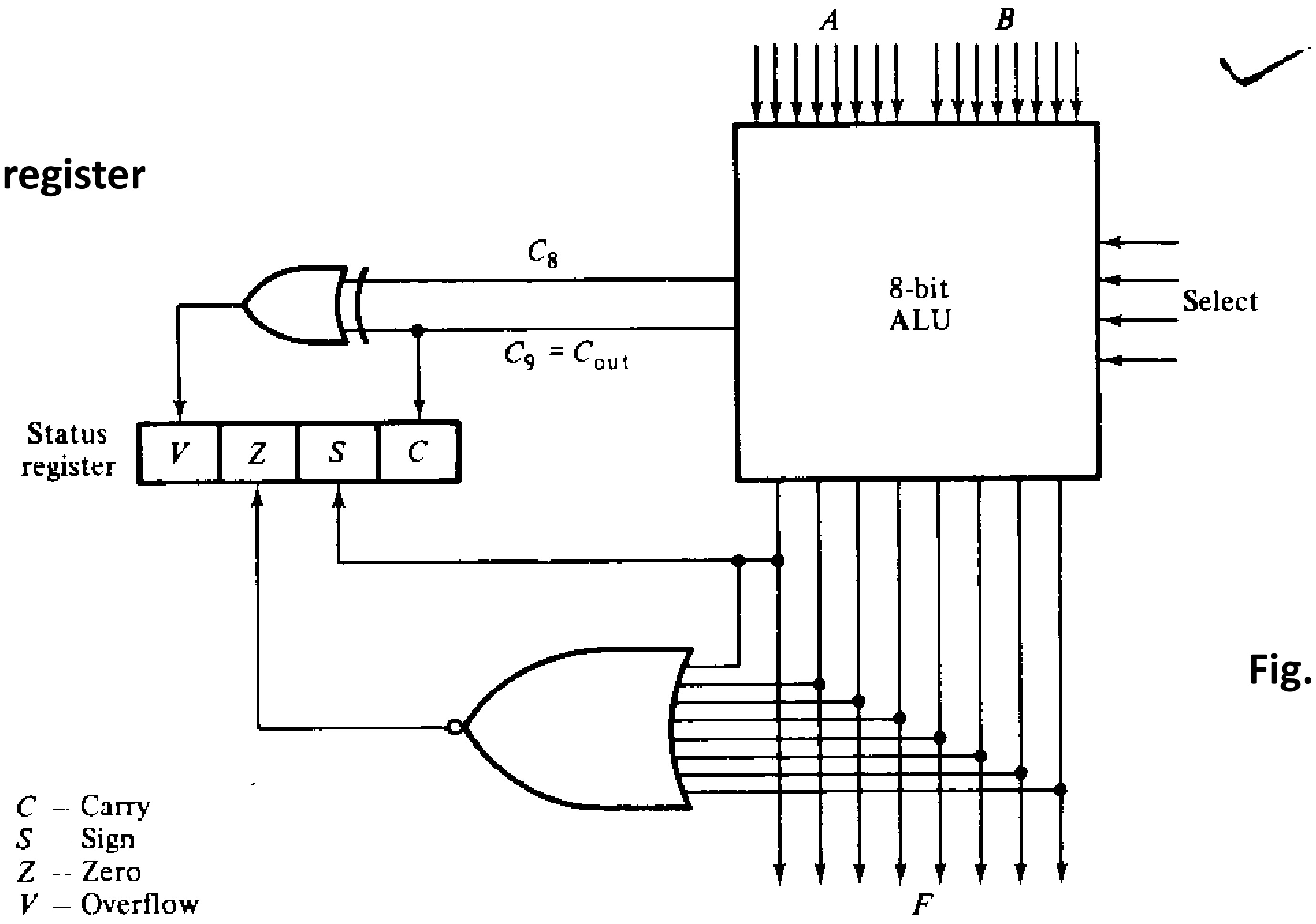


Fig. 9-14

Figure 9-14 shows the block diagram of an 8-bit ALU with a 4-bit status register. The four status bits are symbolized by C, S, Z, and V. The bits are set or cleared as a result of an operation performed in the ALU.

1. Bit C is set if the output carry of the ALU is 1. It is cleared if the output carry is 0.
2. Bit S is set if the highest-order bit of the result in the output of the ALU (the sign bit) is 1. It is cleared if the highest-order bit is 0.
3. Bit Z is set if the output of the ALU contains all 0's, and cleared otherwise. $Z = 1$ if the result is zero, and $Z = 0$ if the result is nonzero.
4. Bit V is set if the exclusive-OR of carries C_8 and C_9 is 1, and cleared otherwise. This is the condition for overflow when the numbers are in sign-2's-complement representation (see Section 8-6). For the 8-bit ALU, V is set if the result is greater than 127 or less than -128.

Processor Unit

The selection variables in a processor unit control the microoperations executed within the processor during any given clock pulse. The selection variables control the buses, the ALU, the shifter, and the destination register. We will now demon-strate by means of an example how the control variables select the microoperations in a processor unit. The example defines a processor unit together with all selection variables. Then we will discuss the choice of control variables for some typical microoperations.

A block diagram of a processor unit is shown in Fig. 9-16(a). It consists of seven registers $R1$ through $R7$ and a status register. The outputs of the seven registers go through two multiplexers to select the inputs to the ALU. Input data from an external source are also selected by the same multiplexers. The output of the ALU goes through a shifter and then to a set of external output terminals. The output from the shifter can be transferred to any one of the registers or to an external destination.

There are 16 selection variables in the unit, and their function is specified by a *control word* in Fig. 9-16(b). The 16-bit control word, when applied to the selection variables in the processor, specifies a given microoperation. The control word is partitioned into six fields, with each field designated by a letter name. All fields, except C_{in}, have a code of three bits. The three bits of A select a source register for the input to left side of the ALU. The B field is the same, but it selects the source information for the right input of the ALU. The D field selects a destination register. The F field, together with the bit in C_{in}, selects a function for the ALU. The H field selects the type of shift in the shifter unit.

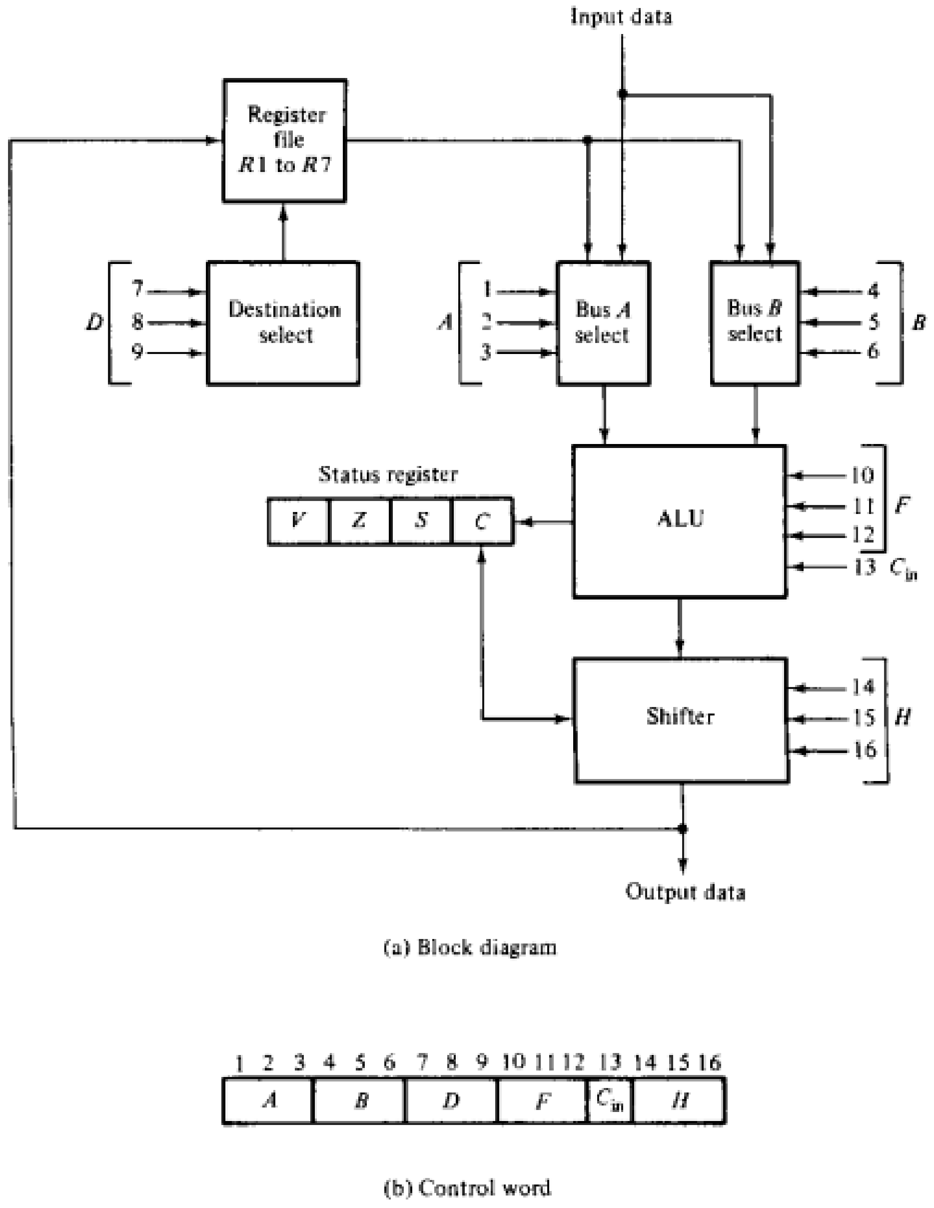
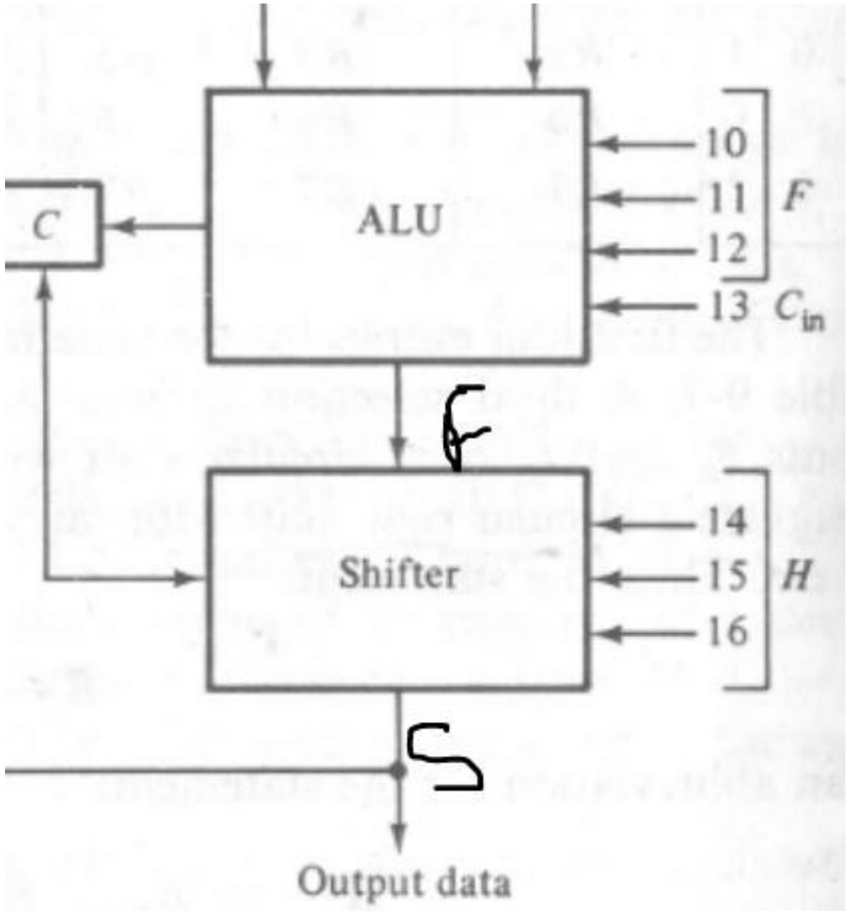


Figure 9-16 Processor unit with control variables

Shifter

The shift unit attached to a processor transfer the output of the ALU onto the output bus. The shifter may transfer the information directly without a shift or it may shift the information to the right or left.

The shifter provides the shift microoperations commonly not available in an ALU.



Function table for shifter

H_1	H_0	Operation	Function
0	0	$S \leftarrow F$	Transfer F to S (no shift)
0	1	$S \leftarrow \text{shr } F$	Shift-right F into S
1	0	$S \leftarrow \text{shl } F$	Shift-left F into S
1	1	$S \leftarrow 0$	Transfer 0's into S

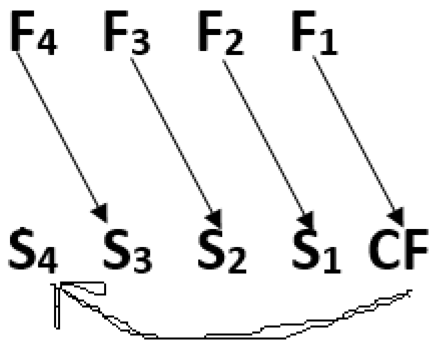
To add more operations in the shifter
8 X 1 MUX are needed with a third selection variable **H_2**.

If CLC or CRC is used, I_L and I_R must be connected to Carry (C).

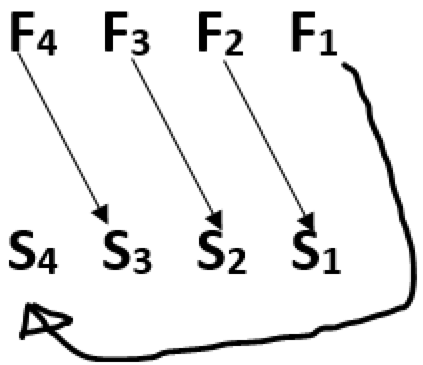
CLC – Circulate Left with Carry
CRC – Circulate Right with Carry

Shifting Operations

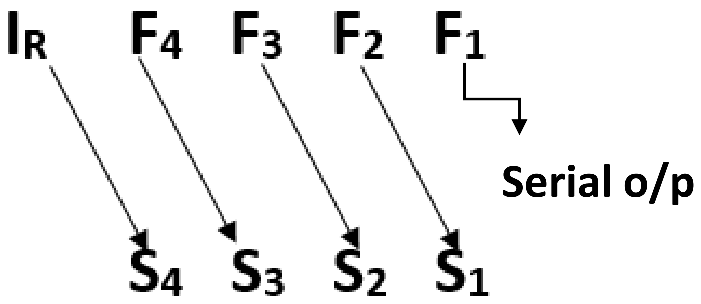
CRC (Circulate Right with Carry)



ROR (Rotate Right/Circulate Right)



SHR (Shift Right) with I_R



Now do for other operations

CLC

ROL

SHL with I_L

Design of Shifter

H_1	H_0	Operation	Function
0	0	$S \leftarrow F$	Transfer F to S (no shift)
0	1	$S \leftarrow$ shr F	Shift-right F into S
1	0	$S \leftarrow$ shl F	Shift-left F into S
1	1	$S \leftarrow 0$	Transfer 0's into S

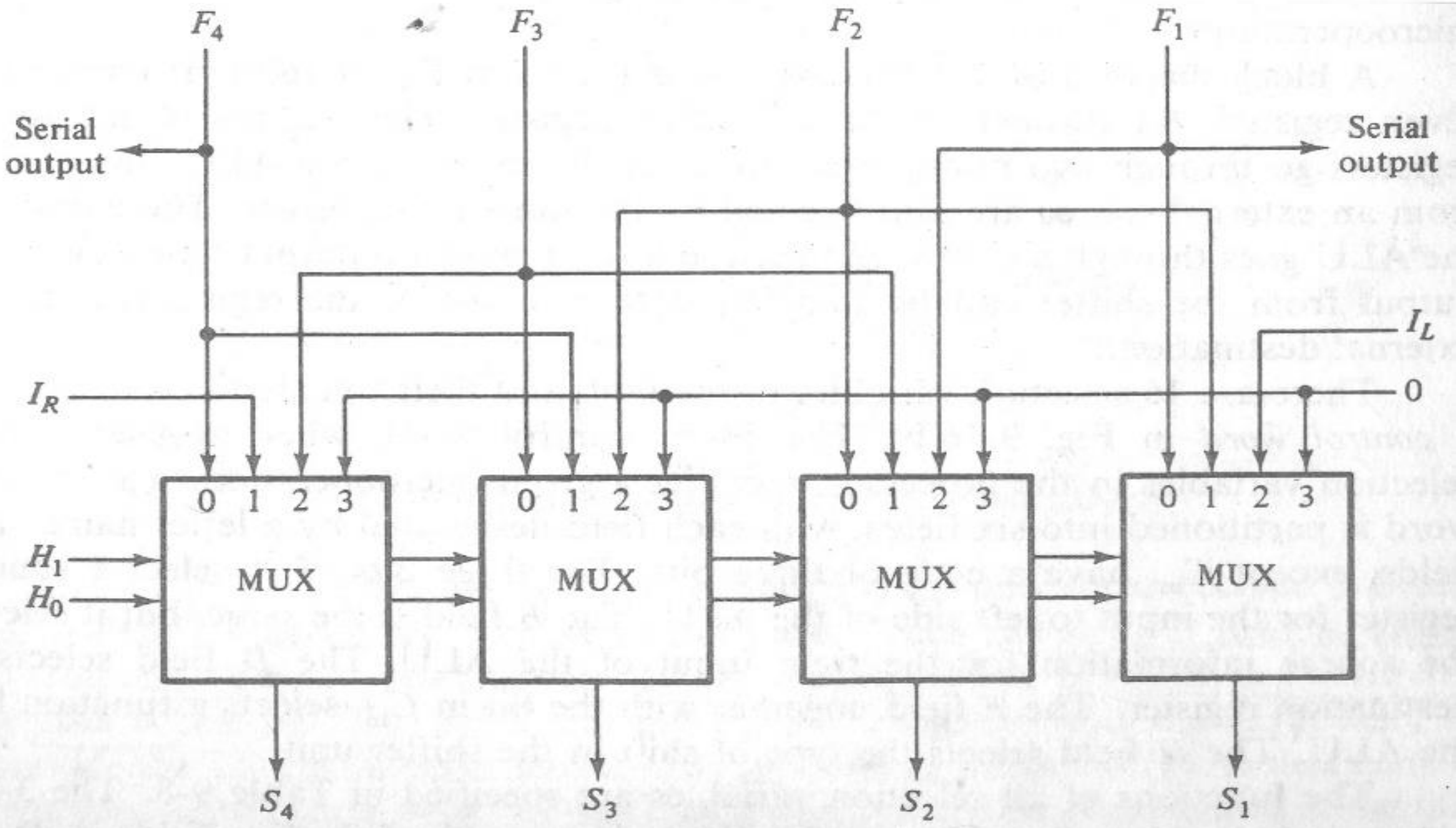


Fig. 4 bit shifter

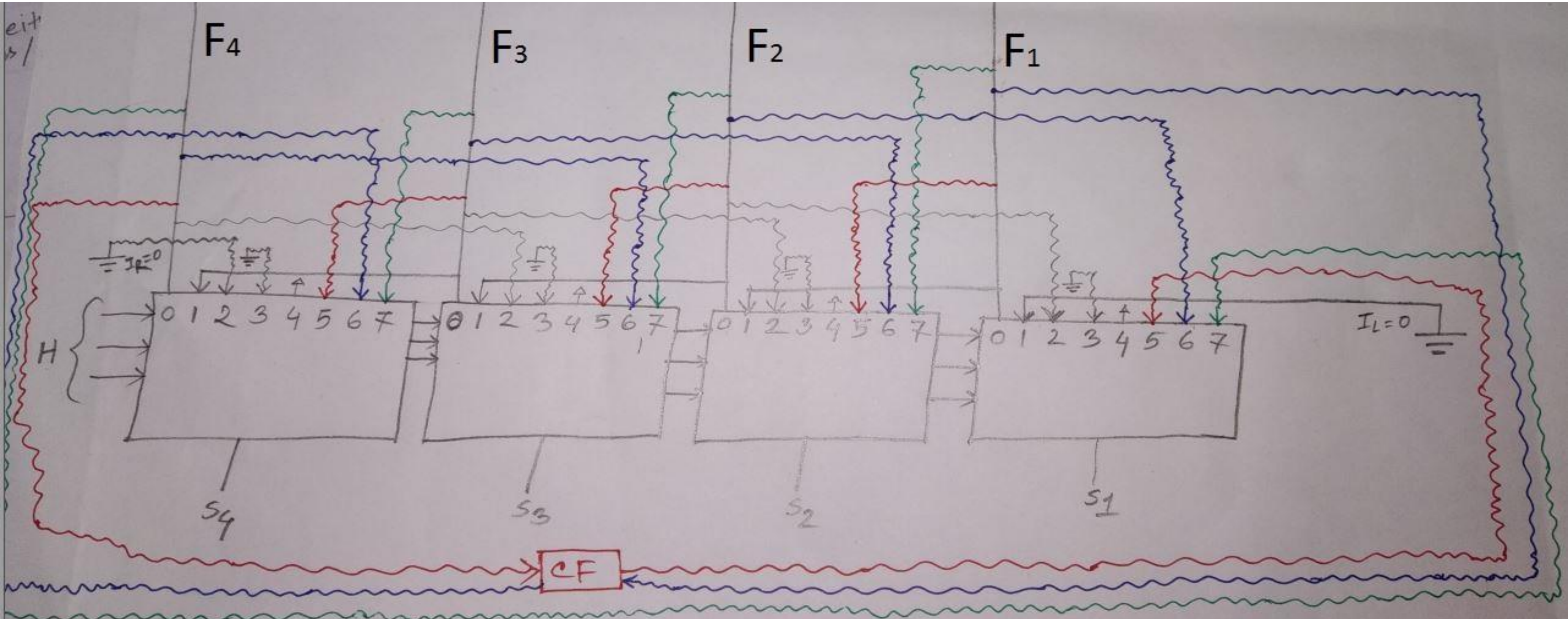
Ex: Design a 4 bit Shifter for the following shifting operation ?

Binary Code	Function of selection variables					
	A	B	F with $C_{in} = 0$	F with $C_{in} = 1$	D	H
0 0 0	Input Data	Input Data	A	A+1	None	No Shift
0 0 1	R1	R1	A+B	A+B+1	R1	Shift Left with I_L=0
0 1 0	R2	R2	A-B-1	A-B	R2	Shift Right with I_R=0
0 1 1	R3	R3	A-1	A	R3	0's to the output Bus
1 0 0	R4	R4	A'	A'	R4	1's to the output Bus
1 0 1	R5	R5	A XOR B	A XOR B	R5	Circulate-Left with Carry
1 1 0	R6	R6	A OR B	A OR B	R6	Circulate-Right with Carry
1 1 1	R7	R7	A AND B	A AND B	R7	Circulate Left/ROL

Soln:

$H_2H_1H_0$	Functions
0 0 0	No Shift
0 0 1	Shift Left with I_L=0
0 1 0	Shift Right with I_R=0
0 1 1	0's to the output Bus
1 0 0	1's to the output Bus
1 0 1	Circulate-Left with Carry
1 1 0	Circulate-Right with Carry
1 1 1	Circulate Left/ROL

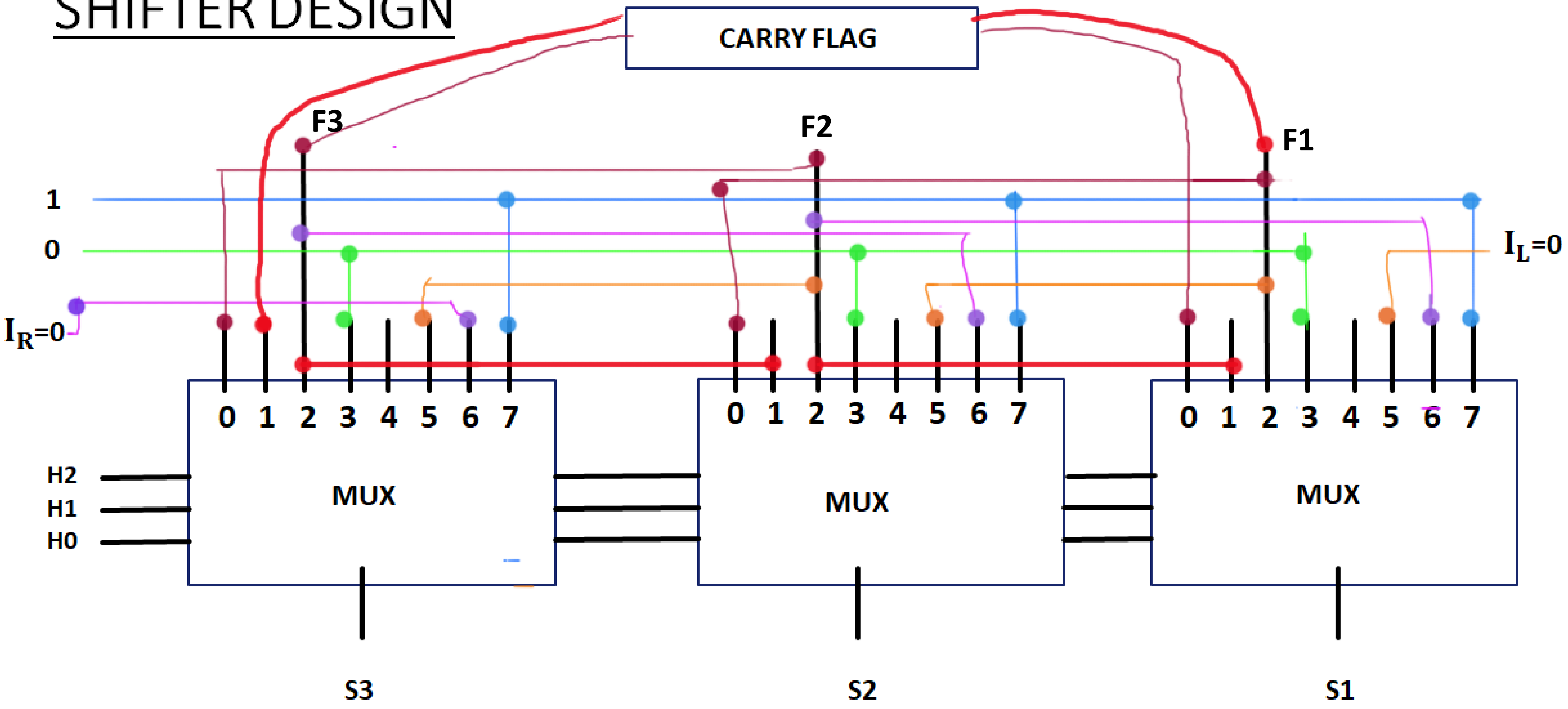
AMERICAN INTERNATIONAL UNIVERSITY- BANGLADESH



Design a 3-bit shifter for the shifting operations listed in the table

Binary Code	A	B	F with C_{in} = 0	F with C_{in} = 1	D	H
0 0 0	Input Data	Input Data	A	A+1	None	Circulate-Left with Carry
0 0 1	R1	R1	A+B	A+B+1	R1	Circulate-Right with Carry
0 1 0	R2	R2	A+B\`	A+B\`+1	R2	No shift
0 1 1	R3	R3	A-1	A	R3	0's to the output Bus
1 0 0	R4	R4	A OR B	A OR B	R4	-
1 0 1	R5	R5	A XOR B	A XOR B	R5	Shift Left with I_L=0
1 1 0	R6	R6	A AND B	A AND B	R6	Shift Right with I_R=0
1 1 1	R7	R7	A′	A′	R7	1's to the output Bus

SHIFTER DESIGN



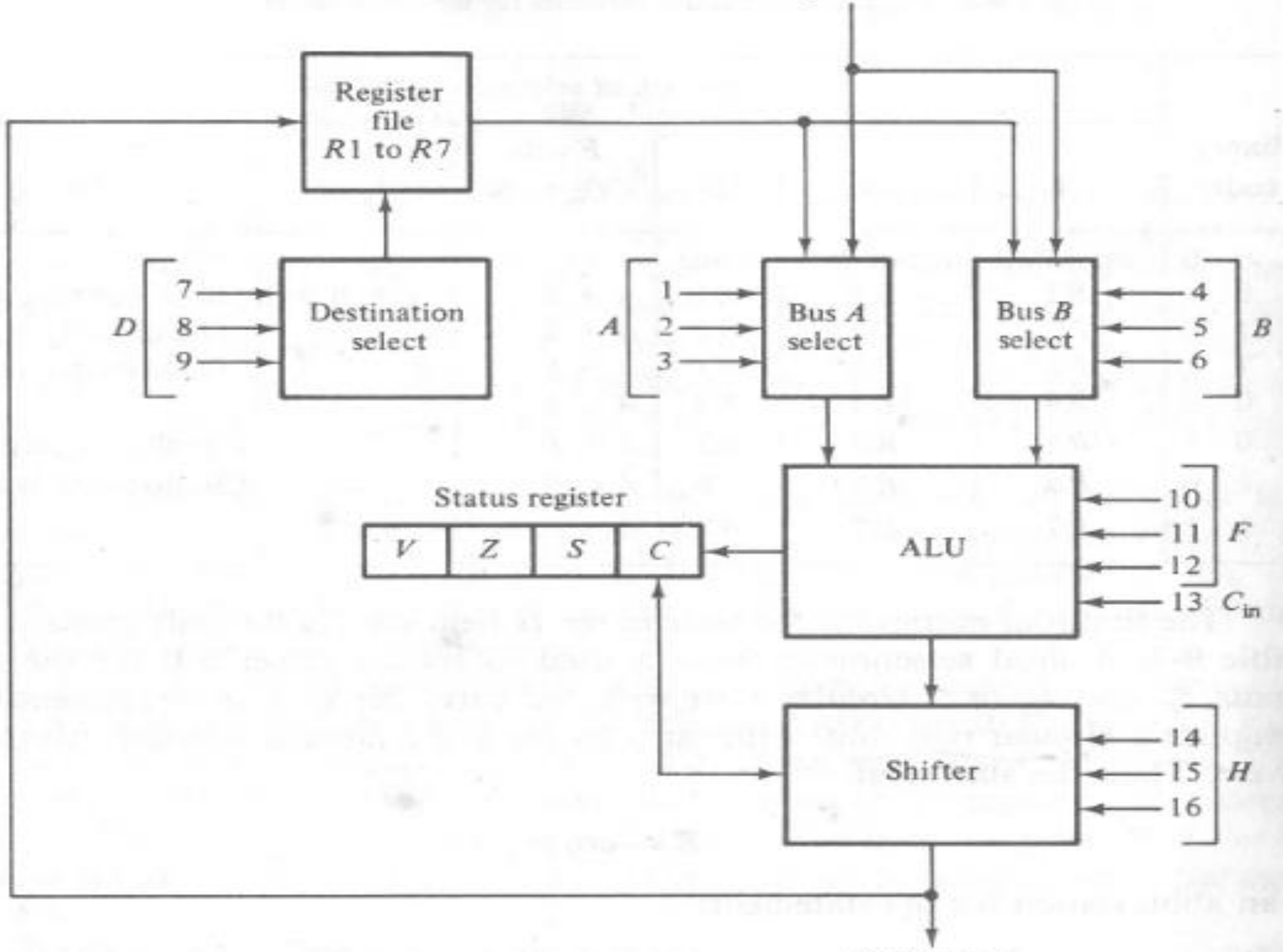
AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Q1. Design a 3-bit shifter for the shift operations listed in Table

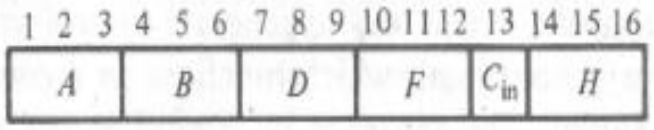
Binary Code	Function of selection variables					
	A	B	F with $C_{in} = 0$	F with $C_{in} = 1$	D	H
0 0 0	Input Data	Input Data	A-B-1	A-B	None	No Shift
0 0 1	R1	R1	A+B	A+B+1	R1	Shift Left with I_L=1
0 1 0	R2	R2	A+1	A	R2	Shift Right with I_R=1
0 1 1	R3	R3	A	A-1	R3	0’s to the output Bus
1 0 0	R4	R4	A XOR B	A XOR B	R4	1’s to the output Bus
1 0 1	R5	R5	A′	A′	R5	Circulate-Left with Carry
1 1 0	R6	R6	A AND B	A AND B	R6	Circulate-Right with Carry
1 1 1	R7	R7	A OR B	A OR B	R7	High impedance to the output Bus

* [In Table 1, consider ‘A’ & ‘B’ as ALU source selection, ‘F’ as the ALU operation selection, ‘D’ as the destination selection and ‘H’ as shift operation selection variables]

Processor Unit with Control Variable



(a) Block diagram



(b) Control word

Examples of Microoperations

AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Table 9-9 Examples of microoperations for processor

Microoperation	A	B	D	F	C_{in}	H	Function
$R1 \leftarrow R1 - R2$	001	010	001	010	1	000	Subtract $R2$ from $R1$
$R3 - R4$	011	100	000	010	1	000	Compare $R3$ and $R4$
$R5 \leftarrow R4$	100	000	101	000	0	000	Transfer $R4$ to $R5$
$R6 \leftarrow$ Input	000	000	110	000	0	000	Input data to $R6$
Output $\leftarrow R7$	111	000	000	000	0	000	Output data from $R7$
$R1 \leftarrow R1, C \leftarrow 0$	001	000	001	000	0	000	Clear carry bit C
$R3 \leftarrow$ shl $R3$	011	011	011	100	0	010	Shift-left $R3$ with $I_L = 0$
$R1 \leftarrow$ crc $R1$	001	001	001	100	0	101	Circulate-right $R1$ with carry
$R2 \leftarrow 0$	000	000	010	000	0	011	Clear $R2$

Function selection variables

Binary code	Function of selection variables					
	A	B	D	F with $C_{in} = 0$	F with $C_{in} = 1$	H
0 0 0	Input data	Input data	None	$A, C \leftarrow 0$	$A + 1$	No shift
0 0 1	$R1$	$R1$	$R1$	$A + B$	$A + B + 1$	Shift-right, $I_R = 0$
0 1 0	$R2$	$R2$	$R2$	$A - B - 1$	$A - B$	Shift-left, $I_L = 0$
0 1 1	$R3$	$R3$	$R3$	$A - 1$	$A, C \leftarrow 1$	0's to output bus
1 0 0	$R4$	$R4$	$R4$	$A \vee B$	—	—
1 0 1	$R5$	$R5$	$R5$	$A \oplus B$	—	Circulate-right with C
1 1 0	$R6$	$R6$	$R6$	$A \wedge B$	—	Circulate-left with C
1 1 1	$R7$	$R7$	$R7$	$\bar{A}$	—	—

If we want to place the contents of a register into the Shifter without changing the carry bit, we can use OR Logic operations with same register selected for both ALU input A and B.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A			B			D			F			Cin	H		

Examples of Microoperations…

AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

R1←R1-R2 , R1 and R2 for the left and right input of the ALU. A-B for the ALU operation. No shift for the shifter. R1 for the destination.

R3-R4, compare operation is similar to the subtract microoperation, except that the difference is not transferred to a destination register.

R5←R4, The transfer of R4 into R5 requires an ALU operation, F=A. The source A is 100 and the destination D is 101. The B selection code could be anything because the ALU does not use it. This field is marked with 000 in the table for convenience but any other 3 bit code could be used.

R6←Input, To transfer the input data to R6, we must have A=000 to select the external input and D = 110 to select the destination register. Again, value of B does not matter and the ALU function is F=A.

Output←R7, To output data from R7, we make A=111 and D=000. The ALU operation, F=A places the information from R7 into the output bus.

Examples of Microoperations…

R1←R1, C←0, It is sometimes necessary to clear or set the carry bit **before a shift operation**. This can be done with ALU select code 0000 and 0111. With first select code the C bit is cleared and with the second code the C bit is set. The transfer R1←R1, C←0 does not change the contents of the register but it clears Carry.

R3←shl R3, If we want to place the contents of a register into the Shifter without changing the carry bit, we can use OR Logic operations with same register selected for both ALU input A and B. The operation **R3←(R3 or R3)** does not change the value of register R3. However, it does place the contents of R3 into the inputs of the shifter and it does not change the values of Carry.

To shift the contents of the register the value of the register must be placed into the shifter without any change through the ALU.

The shift left microoperation specifies the code for the shift select but not the code for the ALU. The contents of R3 can be placed into the shifter by specifying an OR operation between R3 and itself. The shifted information returns to R3 if R3 is specified as the destination register. This requires that select fields A, B and D have the code 011 for R3 that the Alu function code be 1000 for the OR operation and that the shift select H be 010 for the shift left.

Examples of Microoperations…

R1←crc R1, This microoperation specifies the code for the shift select but not the code for the ALU. To place the contents of R3 into the output terminals of the ALU without affecting the Carry bit, we use the OR operation as before. In this way C bit is not affected by the ALU operation but may be changed because of the circular shift.

R2←0, this microinstruction is for clearing the register to 0. To clear register R2, the output bus is made to contain all o's, with H=011. The destination field D is made equal to the code for register R2.

Function selection variables

Binary code	Function of selection variables					
	A	B	D	F with $C_{in} = 0$	F with $C_{in} = 1$	H
0 0 0	Input data	Input data	None	$A, C \leftarrow 0$	$A + 1$	No shift
0 0 1	$R1$	$R1$	$R1$	$A + B$	$A + B + 1$	Shift-right, $I_R = 0$
0 1 0	$R2$	$R2$	$R2$	$A - B - 1$	$A - B$	Shift-left, $I_L = 0$
0 1 1	$R3$	$R3$	$R3$	$A - 1$	$A, C \leftarrow 1$	0's to output bus
1 0 0	$R4$	$R4$	$R4$	$A \vee B$	—	—
1 0 1	$R5$	$R5$	$R5$	$A \oplus B$	—	Circulate-right with C
1 1 0	$R6$	$R6$	$R6$	$A \wedge B$	—	Circulate-left with C
1 1 1	$R7$	$R7$	$R7$	$\bar{A}$	—	—

Q2. For a processor unit with 16-bit control word variable, develop the control words using the below Table-1 in hexadecimal for the following listed micro-operations:

i) R1 ← R1 – R4
ii) R2 ← 1
iii) R2 AND R3
iv) R5 ← SHL R6
v) Output ← R5
vi) R1 ← 2(R2 – 0)/2

Table-1

Binary Code	Function of selection variables					
	A	B	F with $C_{in} = 0$	F with $C_{in} = 1$	D	H
0 0 0	Input Data	Input Data	A	A+1	None	No Shift
0 0 1	R1	R1	A+B	A+B+1	R1	Shift Left with I_L=0
0 1 0	R2	R2	A-B-1	A-B	R2	Shift Right with I_R=0
0 1 1	R3	R3	A-1	A	R3	0's to the output Bus
1 0 0	R4	R4	A'	-	R4	1's to the output Bus
1 0 1	R5	R5	A XOR B	-	R5	Circulate-Left with Carry
1 1 0	R6	R6	A OR B	-	R6	Circulate-Right with Carry
1 1 1	R7	R7	A AND B	-	R7	Circulate Left/ROL

Q3. For a processor unit with 16-bit control word variable as in Table 2, develop the micro-operations using Table 1 for the control words given below:

i) 2A28H
ii) 01B3H
iii) 156BH
iv) 458AH
v) 65ACH

Table-2

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A			B			D			F			Cin	H		

Next...

- Flowchart, State Diagram and Microprogrammed Control Unit Design for addition/subtraction of signed numbers.