AMERICAN INTERNAIONAL UNIVERSITY – BANGLADESH
Where leaders are created

Lecture-2 (Final)
Processor Logic Design Contd…

Department of EEE

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

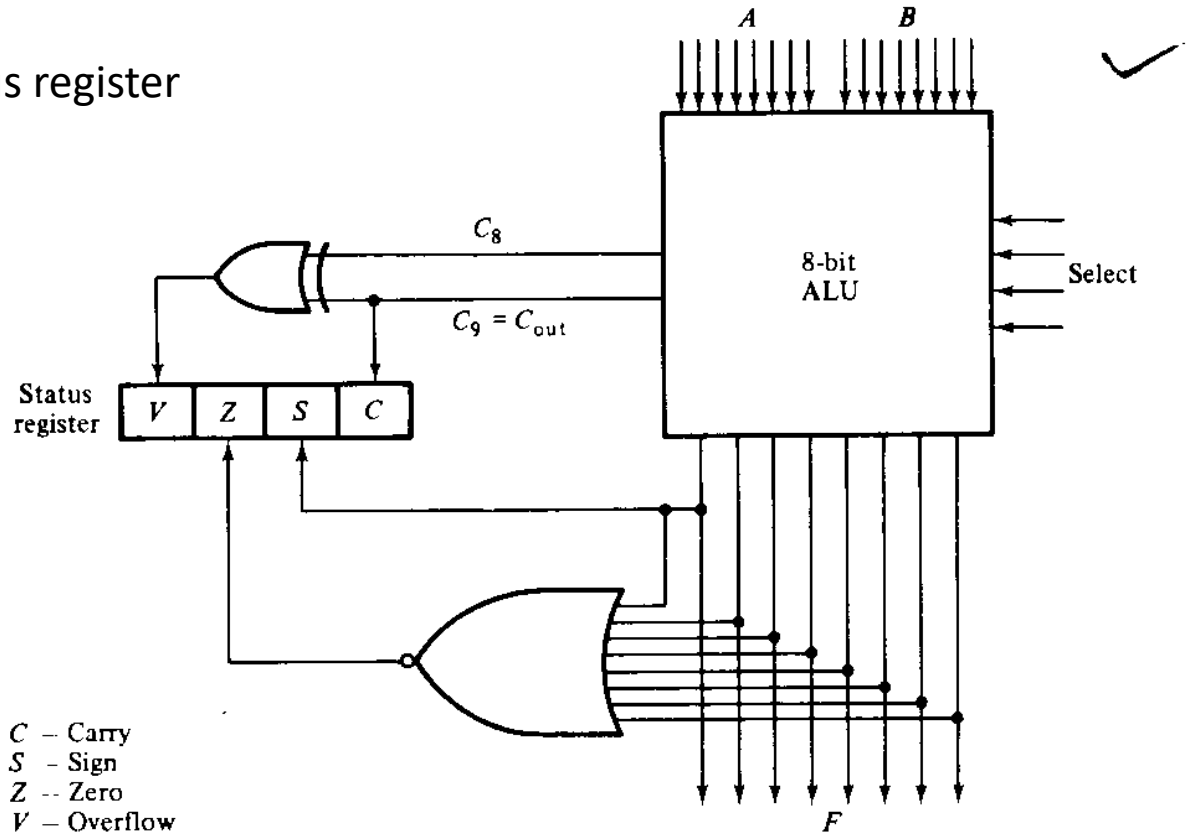


Fig. 9-14

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.

Status bits after the subtraction of unsigned Numbers(A-B)

Relation	Condition of status bits	Boolean function
$A > B$	$C = 1$ and $Z = 0$	CZ'
$A \geqslant B$	$C = 1$	C
$A < B$	$C = 0$	C'
$A \leqslant B$	$C = 0$ or $Z = 1$	$C' + Z$
$A = B$	$Z = 1$	Z
$A \neq B$	$Z = 0$	Z'

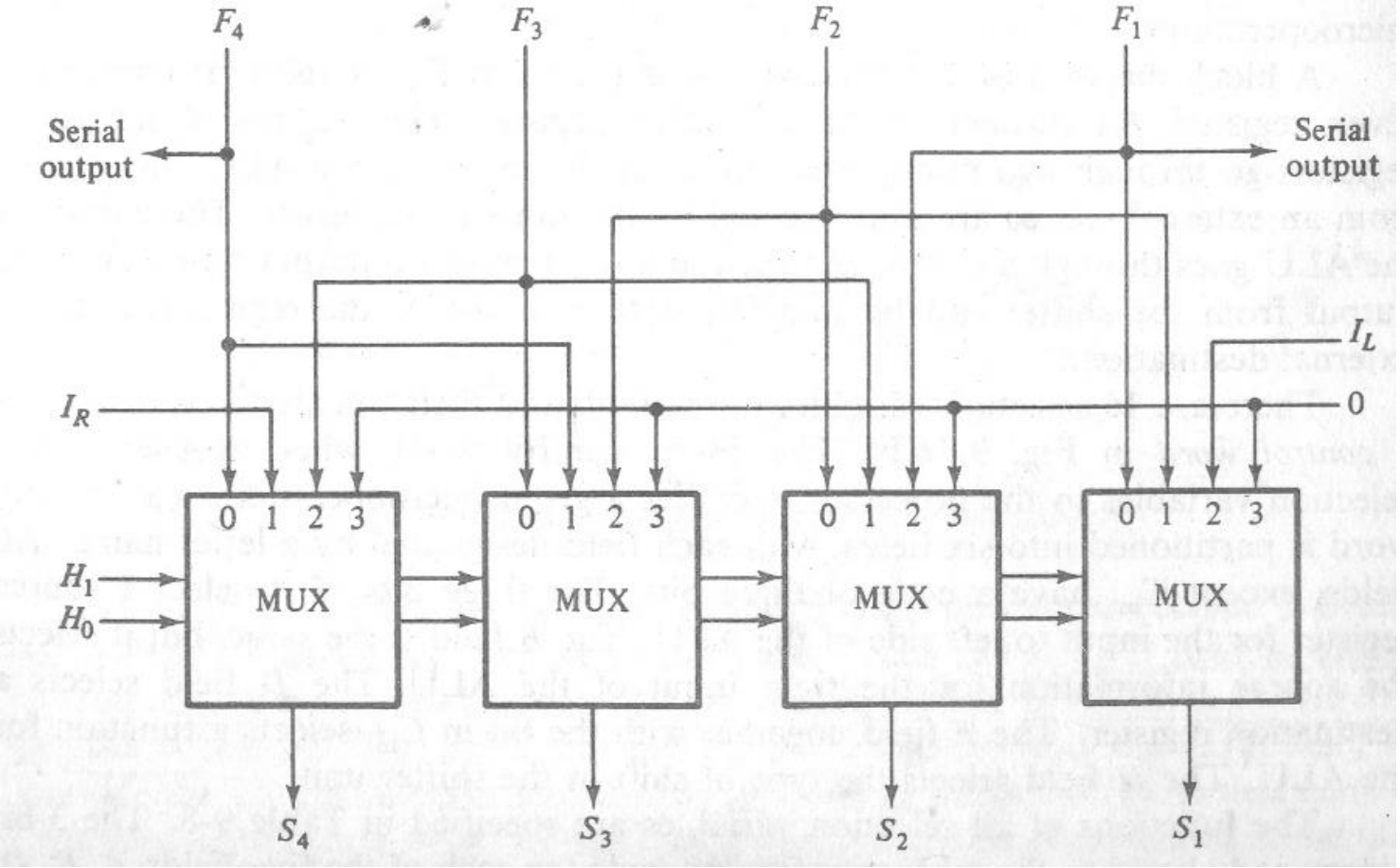
Status bits after the subtraction (A-B) of two signed binary numbers when negative numbers are in 2's complement form

Relation	Condition of status bits	Boolean function
$A > B$	$Z = 0$ and ($S = 0, V = 0$ or $S = 1, V = 1$)	$Z'(S \odot V)$
$A \geqslant B$	$S = 0, V = 0$ or $S = 1, V = 1$	$S \odot V$
$A < B$	$S = 1, V = 0$ or $S = 0, V = 1$	$S \oplus V$
$A \leqslant B$	$S = 1, V = 0$ or $S = 0, V = 1$ or $Z = 1$	$(S \oplus V) + Z$
$A = B$	$Z = 1$	Z
$A \neq B$	$Z = 0$	Z'

Design of Shifter

The shift unit attached to a processor transfers 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. Provision is sometimes made for no transfer from the ALU to the output bus. The shifter provides the shift microoperations commonly not available in an ALU.

Design of Shifter



AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

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 L or CLC R is used, I_L and I_R must be connected to Carry (C).
CLC R – Circulate Left with Carry
CRC R – Circulate Right with Carry

9-9 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 demonstrate 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.

The functions of all selection variables are specified in Table 9-8. The 3-bit binary code listed in the table specifies the code for each of the five fields A, B, D, F, and H. The register selected by A, B, and D is the one whose decimal number is equivalent to the binary number in the code. When the A or B field is 000, the corresponding multiplexer selects the input data. When $D = 000$, no destination register is selected. The three bits in the F field, together with the input carry C_{in}, provide the 12 operations of the ALU as specified in Table 9-4. Note that there are two possibilities for $F = A$. In one case the carry bit C is cleared, and in the other case it is set to 1 (see Table 9-2).

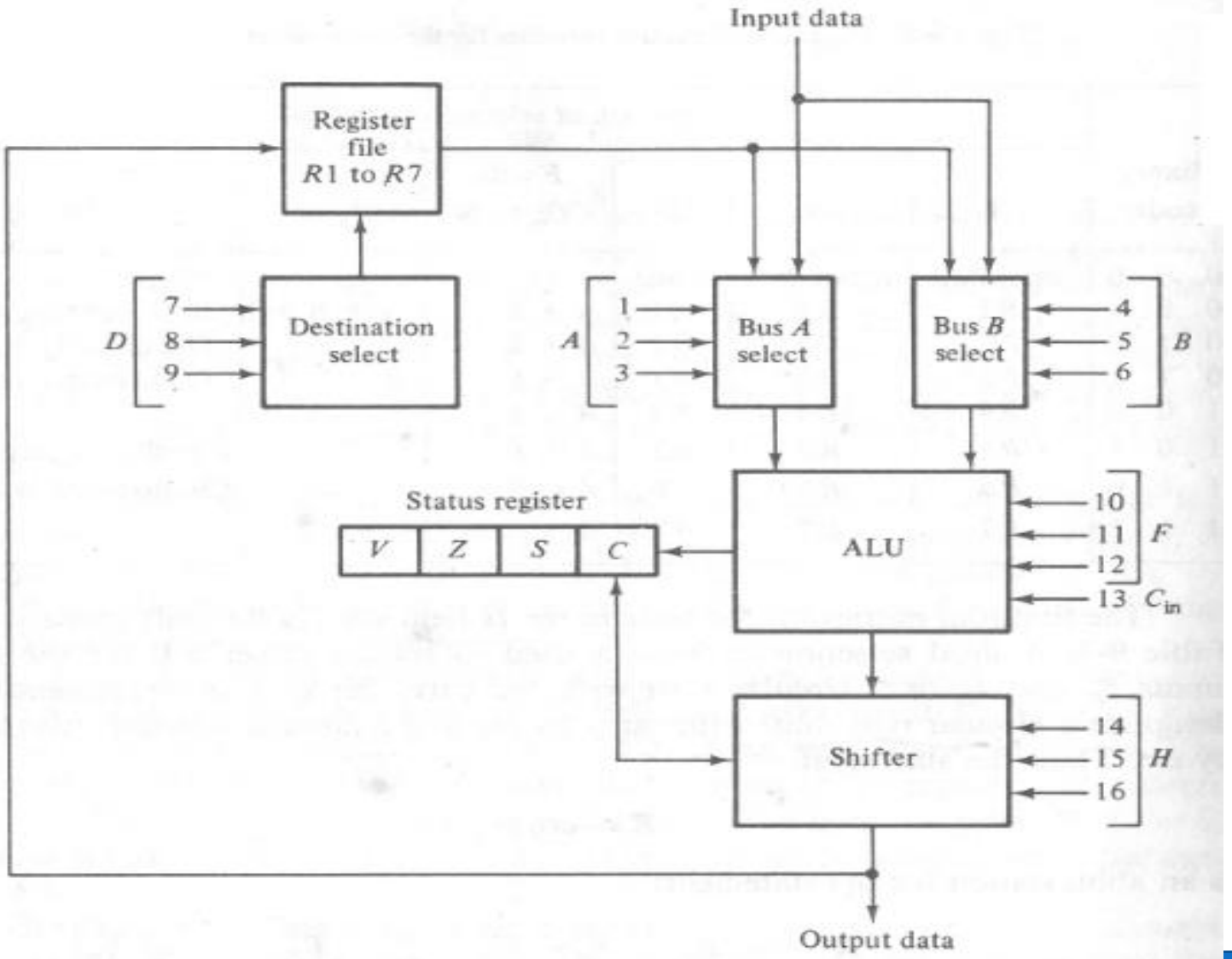
TABLE 9-8 Functions of control variables for the processor of Fig. 9-16

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	$R7$	$R7$	$R7$	$\overline{A}$	—	—

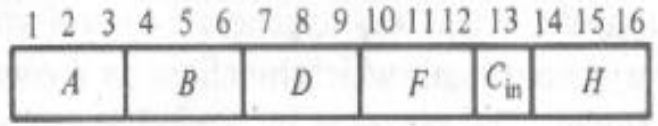
TABLE 9-4 Function table for the ALU of Fig. 9-13

Selection				Output	Function
s_2	s_1	s_0	C_{in}		
0	0	0	0	$F = A$	Transfer A
0	0	0	1	$F = A + 1$	Increment A
0	0	1	0	$F = A + B$	Addition
0	0	1	1	$F = A + B + 1$	Add with carry
0	1	0	0	$F = A - B - 1$	Subtract with borrow
0	1	0	1	$F = A - B$	Subtraction
0	1	1	0	$F = A - 1$	Decrement A
0	1	1	1	$F = A$	Transfer A
1	0	0	X	$F = A \vee B$	OR
1	0	1	X	$F = A \oplus B$	XOR
1	1	0	X	$F = A \wedge B$	AND
1	1	1	X	$F = \bar{A}$	Complement A

Processor Unit with Control Variable



(a) Block diagram



(b) Control word

Fig. 9-16(a)

Examples of Microoperations

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	$R7$	$R7$	$R7$	$\overline{A}$	—	—

TABLE 9-9 Examples of microoperations f

Microoperation	Control word: 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 $R2$

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 bot ALU input A and B.

Next…

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