Unit 7: Processor Logic Design

Introduction:

A processor is that part of a digital system or digital computer that implements the operations in the system. It is comprised of a number of registers and the digital functions that implements arithmetic, logic shift, and transfer micro operations. The processor unit, when combined with a control unit that supervises the sequence of micro operations, is called a *central processing unit* or CPU. The number of registers in a processor unit may vary from processor to processor.

The digital function that implements the micro operations on the information stored in a processor registers is commonly known as arithmetic logic unit or ALU. To perform a microoperations, the control routes the source information from the registers and performs a given operation as specified by the control. The result of the operation is then transferred to a destination register.

Processor Organization

The processor part of a computer CPU is sometimes referred as the *data path* of the CPU because the processor forms the paths for the data transfers between the registers in the unit. The various paths are said to be controlled by means of gates that opens the required path and close all others. A processor unit can be designed to fulfill the requirements of a set of data paths for a specific application.

In a well-organized processor unit, the data paths are formed by means of buses and other common lines. The control gates that formulates the given path are essentially multiplexers and decoders whose selection lines specify the required path.

<u>Bus Organization:</u>

When a large number of registers are included in a processor unit, it is most efficient to connect them through common buses or arrange them as a small memory having very fast access time. The registers communicate with each other not only for direct data transfers, but also while performing various microoperations. A bus organization for four processor registers is shown below:

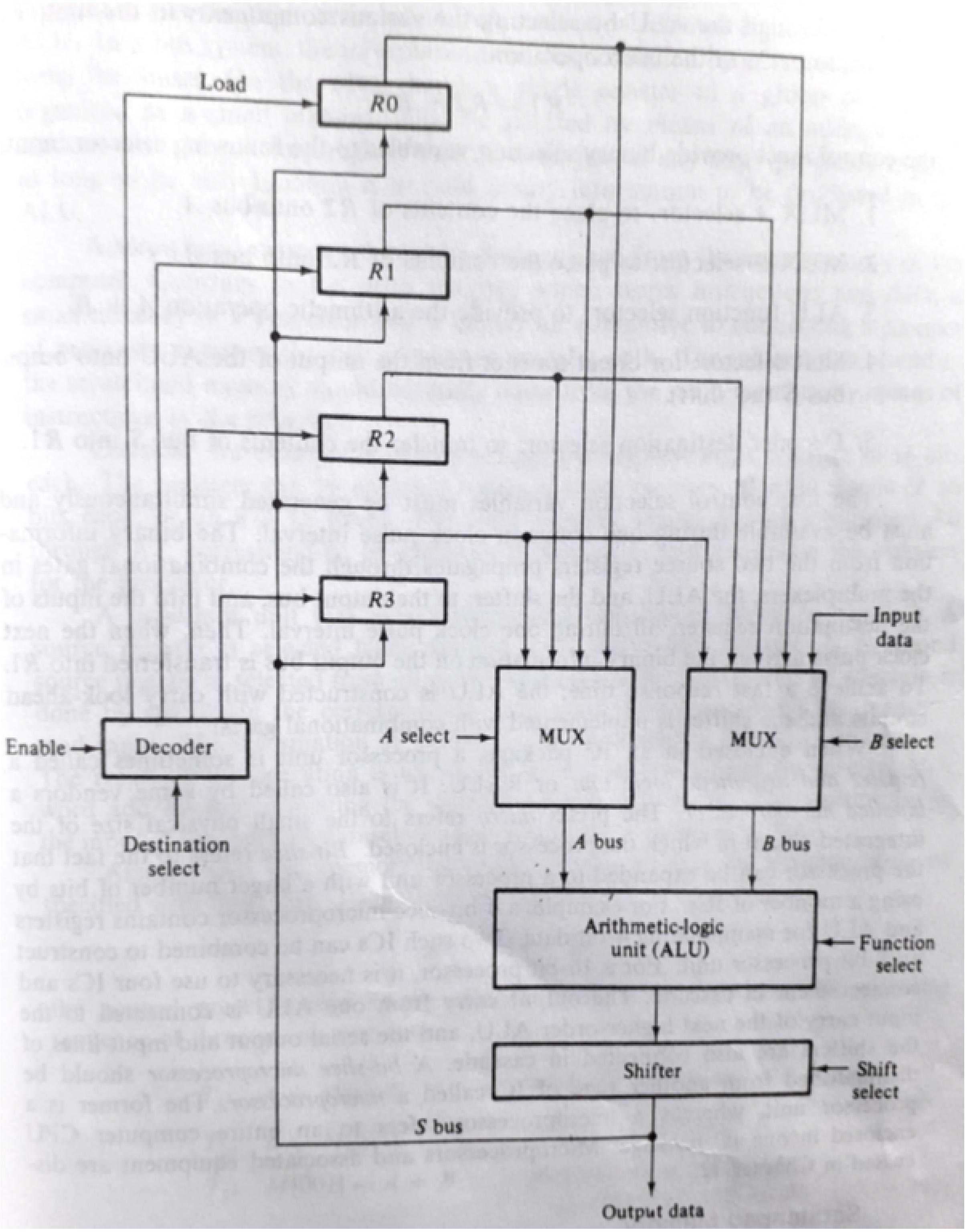


Fig: Processor registers and ALU connected through common buses

In the above figure, each register is connected to two multiplexers (MUX) to form input buses A and B. The selection lines of each multiplexer selects one register for the particular bus. The A and B buses are applied to common arithmetic logic unit (ALU). The function selected in the ALU determines the particular operation that is to be performed. The shift microoperations are performed in the shifter. The result of the microoperation goes through the output bus S into the inputs of all the registers. The destination register that receivers the information from the output bus is selected by a decoder. When enabled, this decoder activates one of the register load inputs to provide a transfer path between the data on the S bus and the inputs of the selected destination register.

A processor unit may have more than four registers. The construction of a bus organized processor with more registers requires larger multiplexers and decoders.

Arithmetic Logic Unit

An arithmetic logic unit (ALU) is a digital circuit used to perform arithmetic and logic operations. It represents the fundamental building block of the central processing unit (CPU) of a computer. Modern CPUs contain very powerful and complex ALUs. In addition to ALUs, modern CPUs contain a control unit (CU).

An arithmetic logic unit is a multioperation, combinational logic digital function. It can perform a set of basic arithmetic operations and a set of logic operations. The ALU has number of selection lines to select a particular operation in the unit. The selection lines are decoded within the ALU so that *K* selection variables can specify 2^K separate operations. The figure below shoes the block diagram of a 4-bit ALU. The four data inputs from A are combined with the four inputs from B to generate an operation at the F outputs.

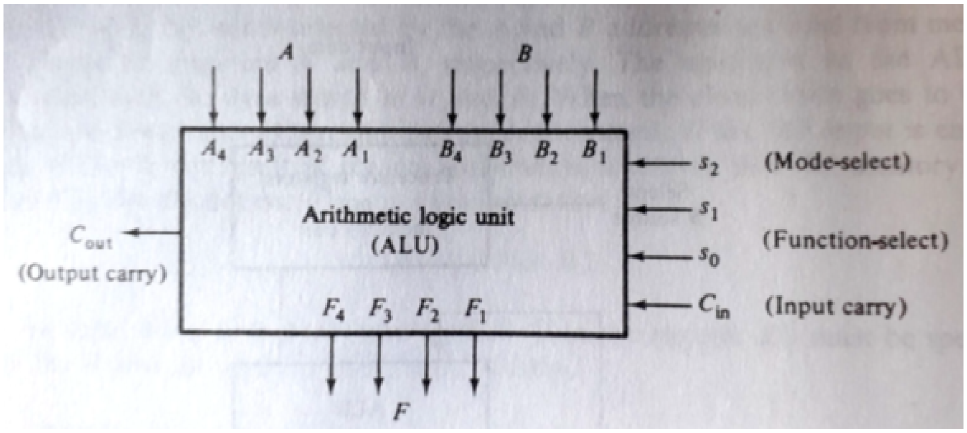
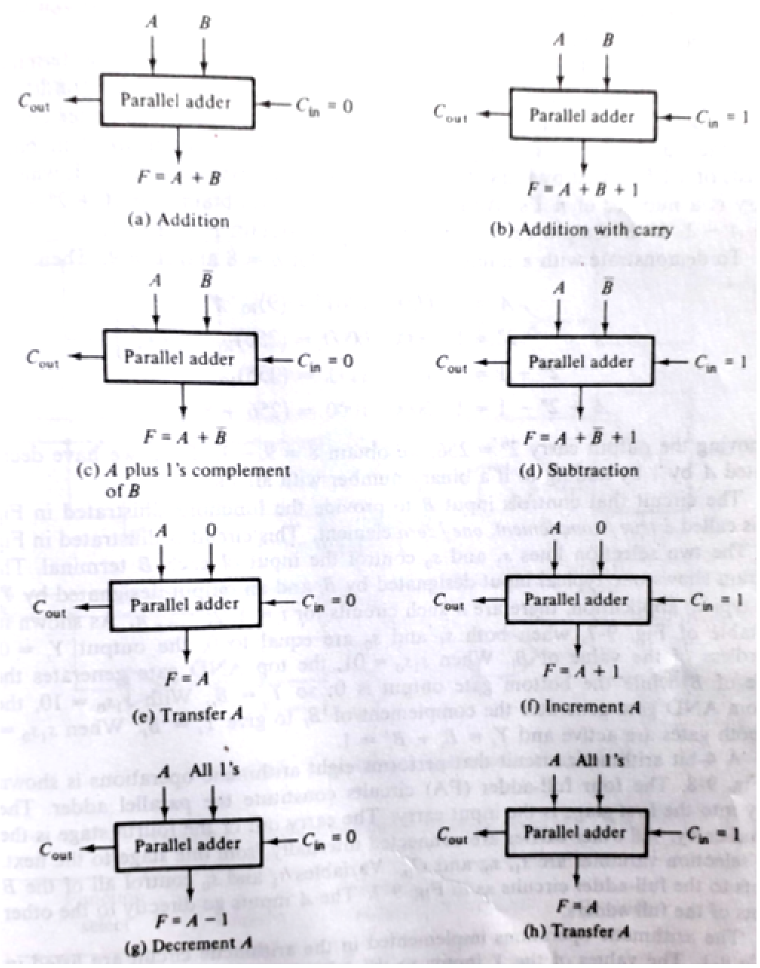


Fig: Block diagram of 4 bit ALU

Design of Arithmetic Circuit

The basic component of the arithmetic section of an ALU is a parallel adder. A parallel adder is constructed with a number of full-adder circuits connected in cascade. By controlling the data inputs to the parallel adder, it is possible to obtain different types of arithmetic operations.



(a) Addition

(b) Addition with carry

(c) A plus 1's complement of B

(d) Subtraction

(e) Transfer A

(f) Increment A

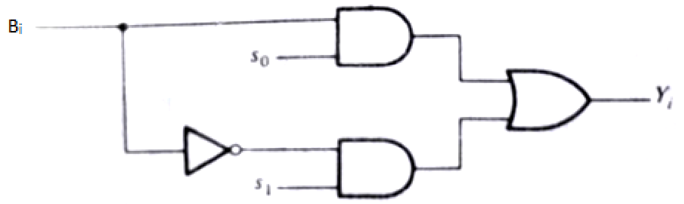
(g) Decrement A

(h) Transfer A

Fig: Eight different operations of ALU

The number is bits in the parallel adder may be of any value. The input carry C_{in} goes to the full-adder circuit in the least significant bit position. The output carry C_{out} comes from the full-adder in the most significant bit position.

A 4-bit arithmetic circuit that performs eight arithmetic operations is shown in figure below. The four full adder(FA) circuits forms the parallel adder. The carry in the first stage is the input carry. The carry out of the fourth stage is the output carry. All other carries are connected internally from one stage to the next. The selection variables are s_1, s_0, and $C_{in.}$ Variables s1 and s0 control all of the B inputs to the full adder circuit as shown in figure. The A input go directly to the other inputs of the full adders.



S_1	S_0	Y_i
0	0	0
0	1	B_i
1	0	B_i'

1	1	1

Figure: True/Complement, one/zero circuit.

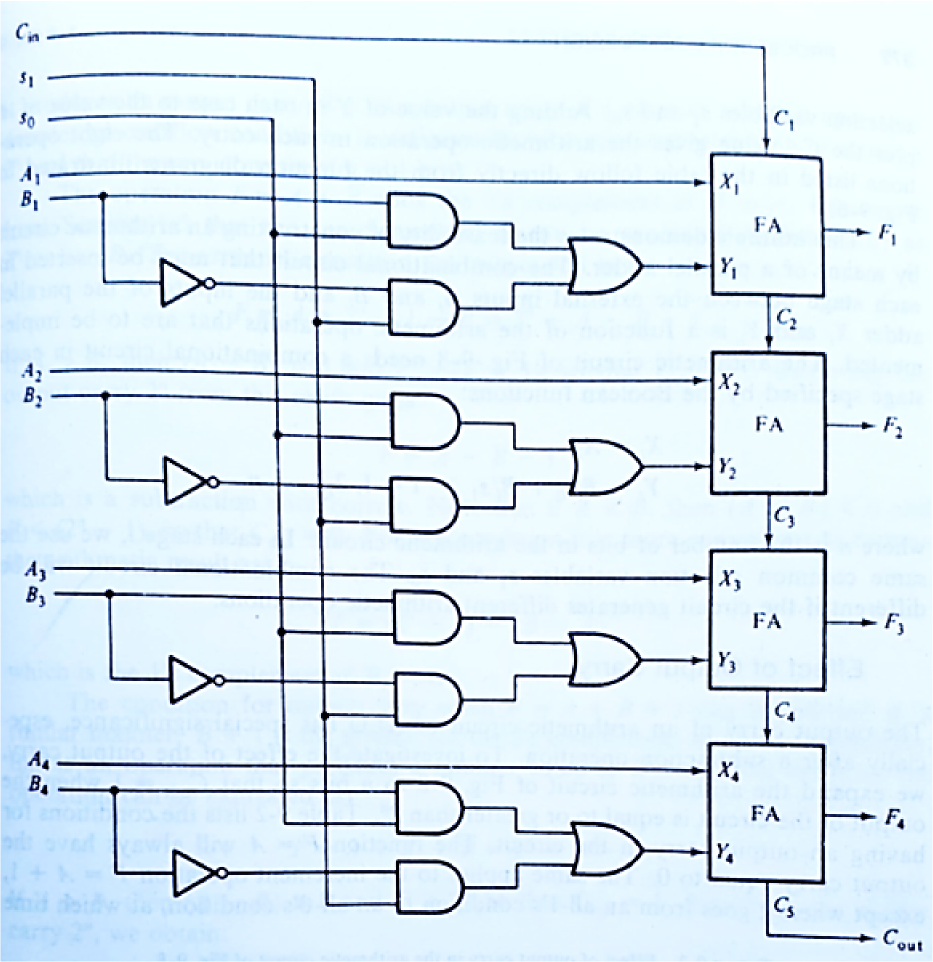
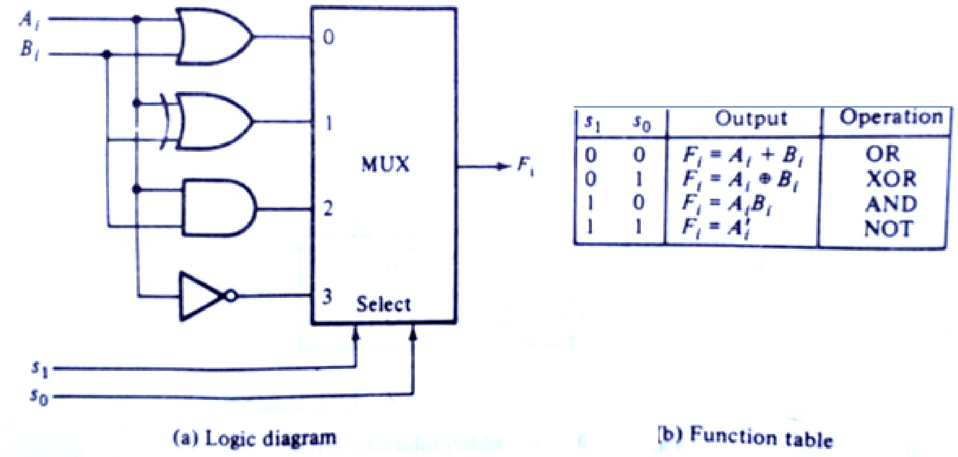


Figure: Logic diagram of arithmetic circuit.

Design of Logic Circuit

The logic microoperation manipulate the bits of the operands separately and treat each bit as a binary variable. All logic operations can be obtained by means of AND, OR and NOT (complement) operations. The simplest and most straightforward way to design a logic circuit is shown in figure below:



s_1	s_0	Output	Operation
0	0	$F_i = A_i + B_i$	OR
0	1	$F_i = A_i \oplus B_i$	XOR
1	0	$F_i = A_i B_i$	AND
1	1	$F_i = A_i'$	NOT

(a) Logic diagram

(b) Function table

Fig: One stage of logic circuit.

The above diagram shows one typical stage designed by subscript *i* . The circuit must be repeated *n* times for an *n*-bit logic circuit.

- ★ The four gates generate the four logic operations OR, XOR, AND, and NOT.
- ★ The two selection variables in the multiplexer select one of the gates for the output.
- ★ The function table lists the output logic generated as a function of the two selection variables.

Design of Arithmetic Logic Unit

ALU can be design with eight arithmetic operations and four logic operations. Three selection variable S_2, S_1 and S_0 select eight different operations, and the input carry C_{in} is used to select four additional arithmetic operations.

- With S_2=0, selection variable S_1 and S_0 together with C_{in} will select the eight arithmetic operations.
- With S_2=1, variable S_1 and S_0 will select the four logic operations OR, XOR, AND and NOT.

We can design one stage of the ALU and then duplicate it for the number of stages required. There are six inputs to each stage: A_i, B_i, C_i, S_2, S_1 and S_0. There are two outputs in each stage: Fi and the carry out C_{i+1}.

The steps involved in the design of an ALU are:

1. Design the arithmetic section independent of the logic section
2. Determine the logic operations obtained from the arithmetic circuit in step 1, assuming that the input carries to all stages are 0.
3. Modify the arithmetic circuit to obtain the required logic operations.

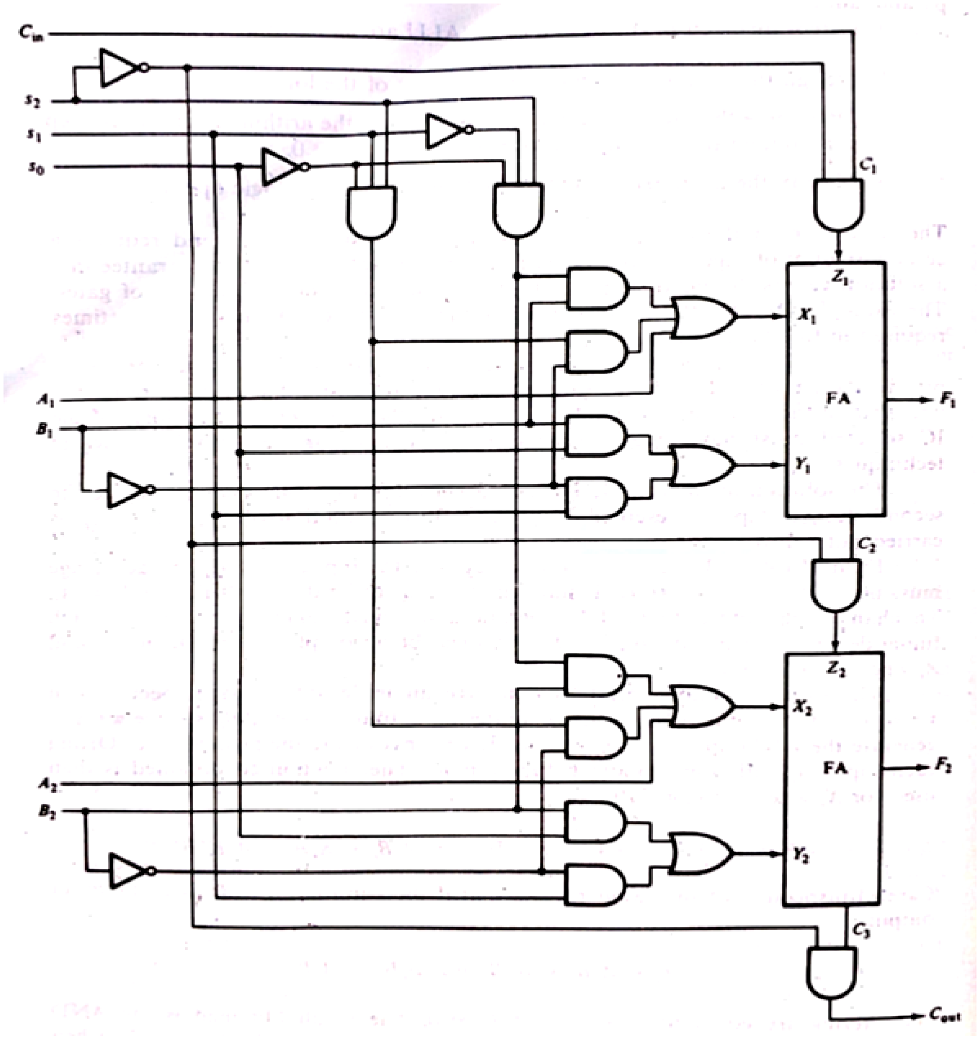
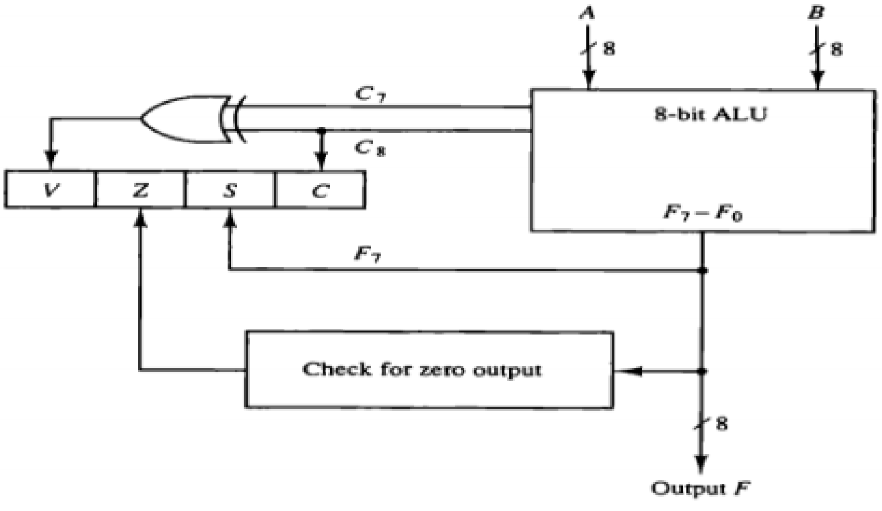


Fig: Logic diagram of ALU

Status Register

The purpose of the Status Register is to hold information about the most recently performed ALU operation, control the enabling and disabling of interrupts and set the CPU operating mode. STATUS register is an 8-bit reserved memory location which contains 8 different flags (each flag takes one bit with 0 or 1 value). These flags are used for keeping different Status of the execution and selection of other execution related things like:

- Finding the result of a mathematical operation- whether the result is zero or negative or positive
- Selecting the variable addressing method



Status register bits.

- ★ C-Carry Flag
- ★ S- Sign Flag
- ★ Z- Zero Flag
- ★ V- Overflow Flag

Fig: Setting bits in a status register

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 with a shift, or it may shift the information to right or left. The shifter provides the shift microoperation commonly not available in an ALU. Shifter is constructed with a bidirectional shift registers with parallel load. The information from the ALU can be transferred to the register in parallel and then shifted to the right or left. In this configuration, a clock pulse is needed for the transfer to the shift register and another clock pulse is needed for the shift. These two pulses are in addition to the pulse required to transfer the information from the shift register to a destination register.

A combinational logic shifter can be constructed with multiplexers as shown in figure below:

- The two selection variables, H_1 and $H_{0,}$ applied to all four multiplexers selects the type of operation in the shifter.
- With H_1H_0= 00, no shift is executed and the signal from F goes directly to the S lines.
- The next two selection variables cause a shift-right operation and shift-left operation.
- When H_1H_0=11, the multiplexers select the input attached to 0 and as a consequence the S outputs are equal to 0, blocking the transfer of information from the ALU to the output bus.

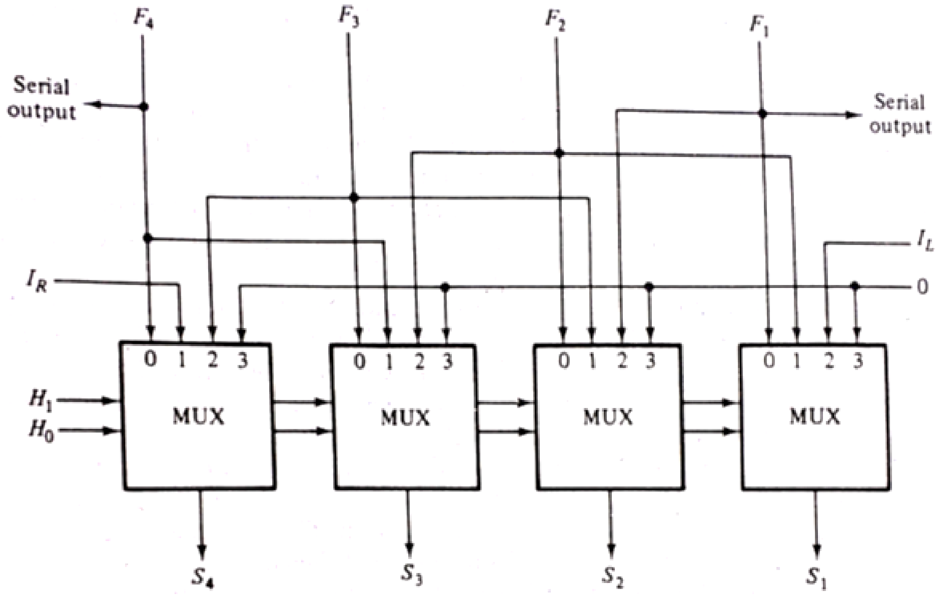


Fig: 4-bit combinational logic shifter

Table describing the operation of the shifter.

H1	H0	Operation	Function
0	0	S <- F	Transfer F to S (No Shift)
0	1	S<-shr F	Shift- Right F into S
1	0	S<- shl F	Shift- Left into S
1	1	S <- 0	Transfer 0's into S

Microoperations

In computer central processing units, **micro-operations** (also known as micro-ops) are the functional or atomic, operations of a processor. These are low level instructions used in some designs to implement complex machine instructions. They generally perform operations on data stored in one or more registers. They transfer data between registers or between external buses of the CPU, also performs arithmetic and logical operations on registers.

- End of Unit 7 -