Unit: Arithmetic Logic Unit

ALU:

Arithmetic logic unit (ALU) is multifunctional device that can perform both A & L function. ALU is an integral part of central processing unit or CPU of a computer.

An ALU is a multipperational combination logic digital for function. It has a numbers 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 upto 2^k distinct operation. Fig shows the block diagram of 4 bit ALU. The data inputs from A are combined with the 4 inputs from B to generate an operation at the F outputs.

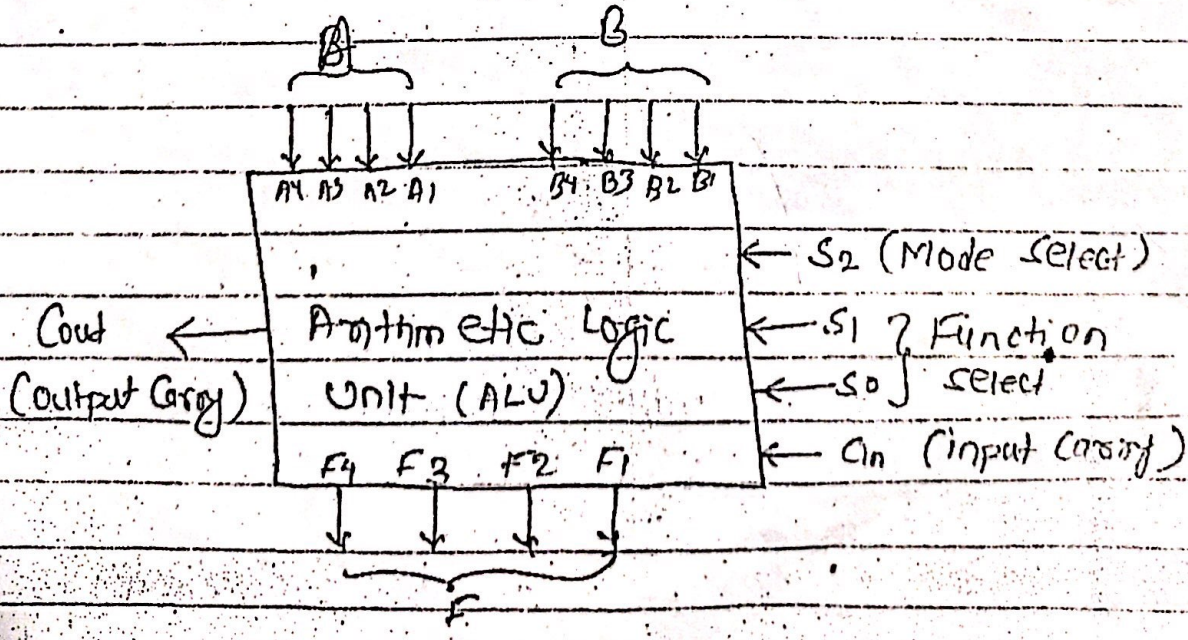
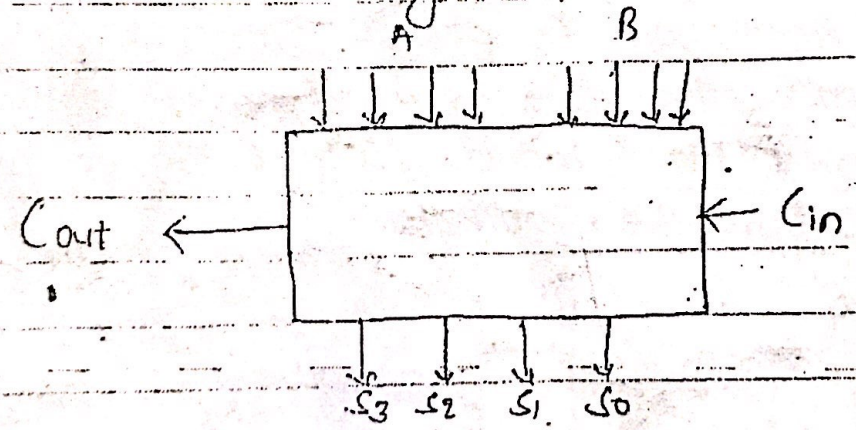


fig: Block Diagram of ALU

The ALU operates on inputs A & B. Operation of the ALU are arithmetic and logic function. S_2 control input line is used to select between arithmetic and logic mode of ALU. Control i/p lines S_1 & S_0 are used to select various functions of ALU like Addition, Subtraction.

Nibble Adder:

Nibble adder is the parallel adder which parallelly adds two 4 bits binary number



The nibble adder is constructed using the Adders (i.e Full Adders) connected in cascade. The nibble adders has the carry in (Cin) and Carry out (Cout). As a result of which it can be used to make 8-bit adders.

Id $A = A_3 A_2 A_1 A_0$

$B = B_3 B_2 B_1 B_0$

$S = S_3 S_2 S_1 S_0$

$C = C_3$

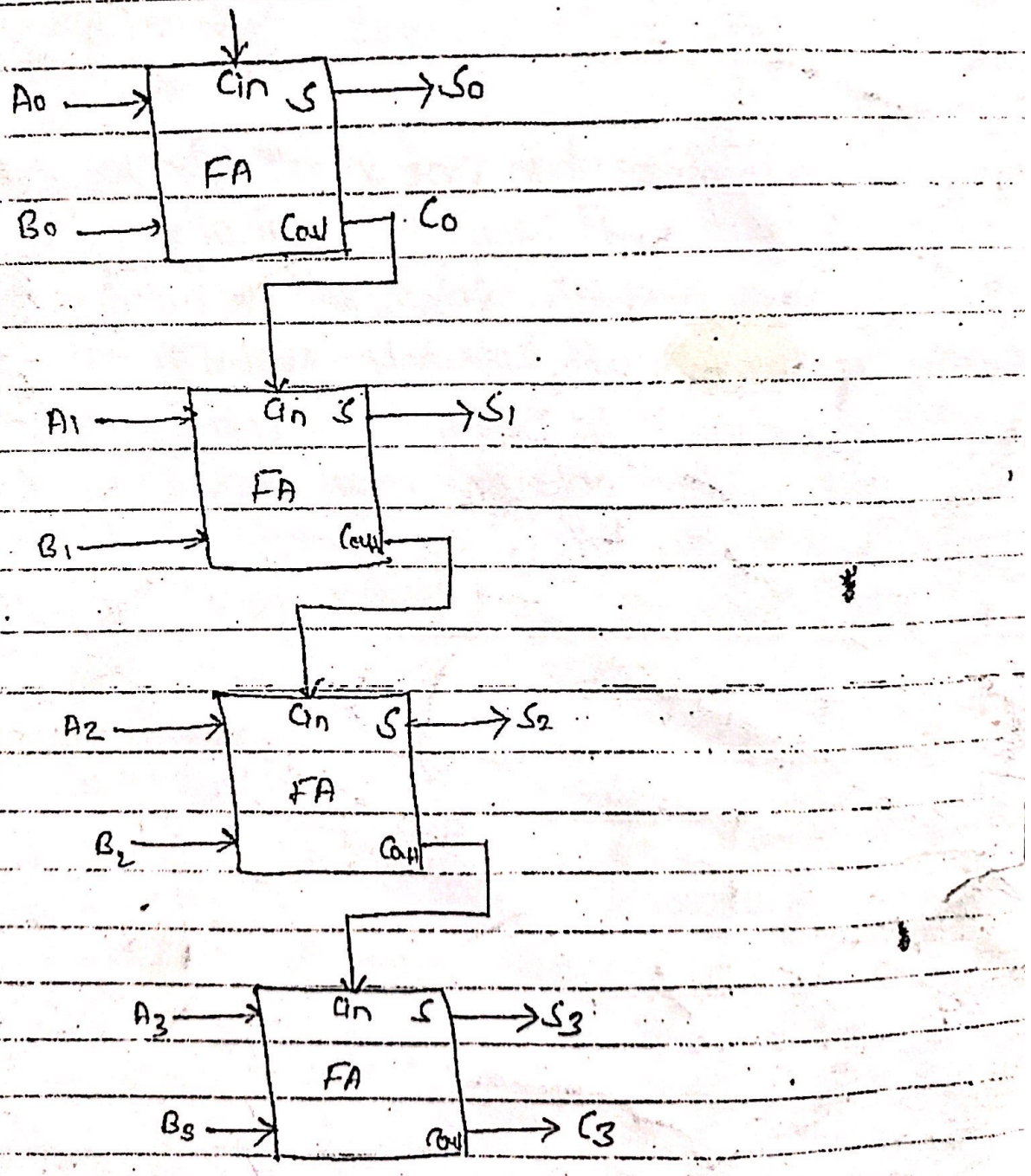


fig: ~~4 bit~~ Nibble Adder

Adder - Subtractor unit :-

→ In Digital ckt, A binary adder - subtractor is capable of both the addition & subtraction of binary no. in one ckt itself.

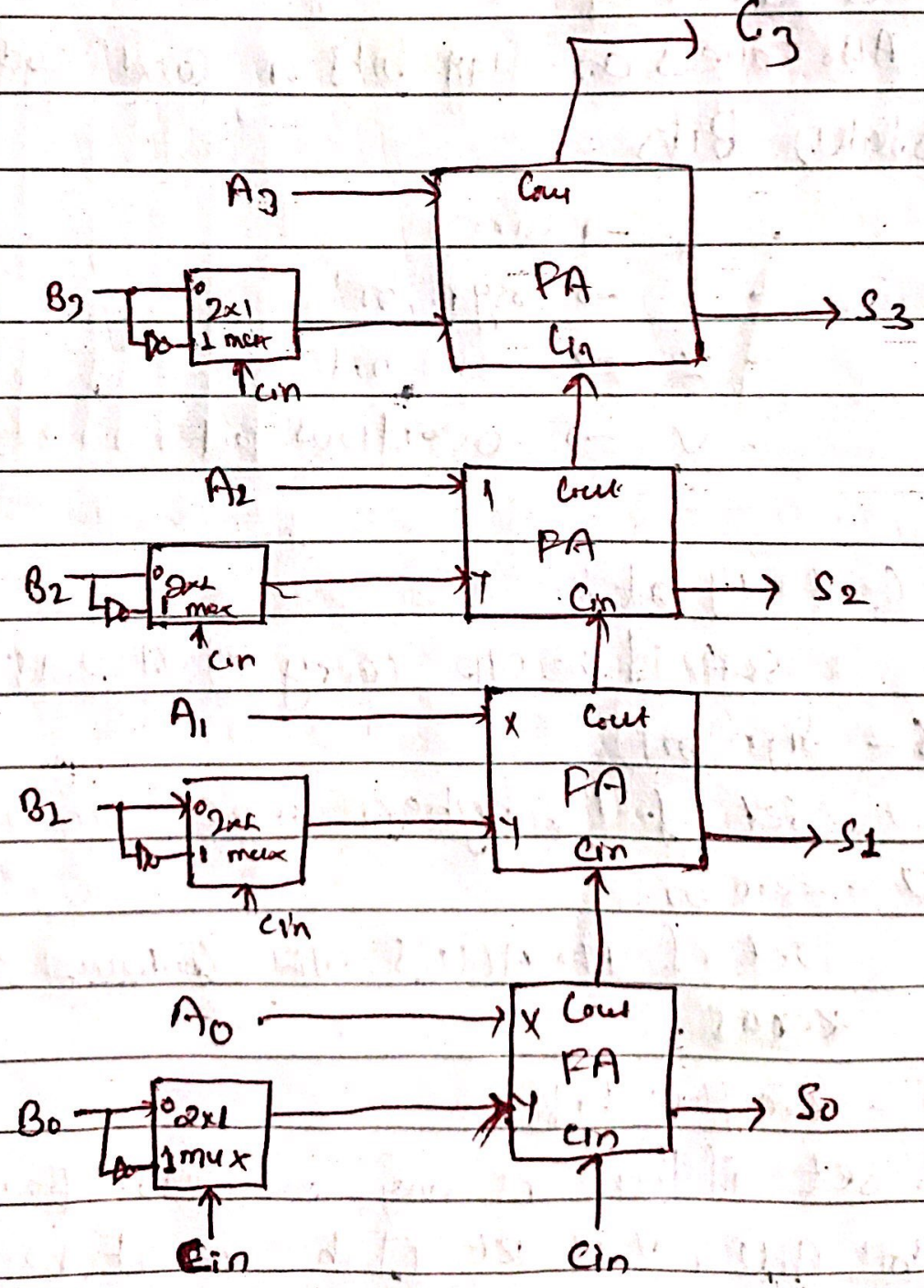
→ The operation is performed depending on the binary values the control signals holds.

- It is one of the components of the ALU.

4 Adder / Subtractor unit

$A+B = A+B+0$

$A-B = A+\bar{B}+1$



when $C_{in} = 1$ for subtract

for Add

9-7 STATUS REGISTER

The relative magnitudes of two numbers may be determined by subtracting one number from the other and then checking certain bit conditions in the resultant difference. If the two numbers are unsigned, the bit conditions of interest are the output carry and a possible zero result. If the two numbers include a sign bit in the highest-order position, the bit conditions of interest are the sign of the result, a zero indication, and an overflow condition. It is sometimes convenient to supplement the ALU with a status register where these status-bit conditions are stored for further analysis. Status-bit conditions are sometimes called *condition-code* bits or *flag* bits.

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.

The status bits can be checked after an ALU operation to determine certain relationships that exist between the values of A and B. If bit V is set after the addition of two signed numbers, it indicates an overflow condition. If Z is set after an exclusive-OR operation, it indicates that $A = B$. This is so because $x \oplus x = 0$, and the exclusive-OR of two equal operands gives an all-0's result which sets the Z bit. A single bit in A can be checked to determine if it is 0 or 1 by masking all bits except the bit in question and then checking the Z status bit. For example, let $A = 101x1100$, where x is the bit to be checked. The AND operation of A with $B = 00010000$ produces a result $000x0000$. If $x = 0$, the Z status bit is set, but if $x = 1$, the Z bit is cleared since the result is not

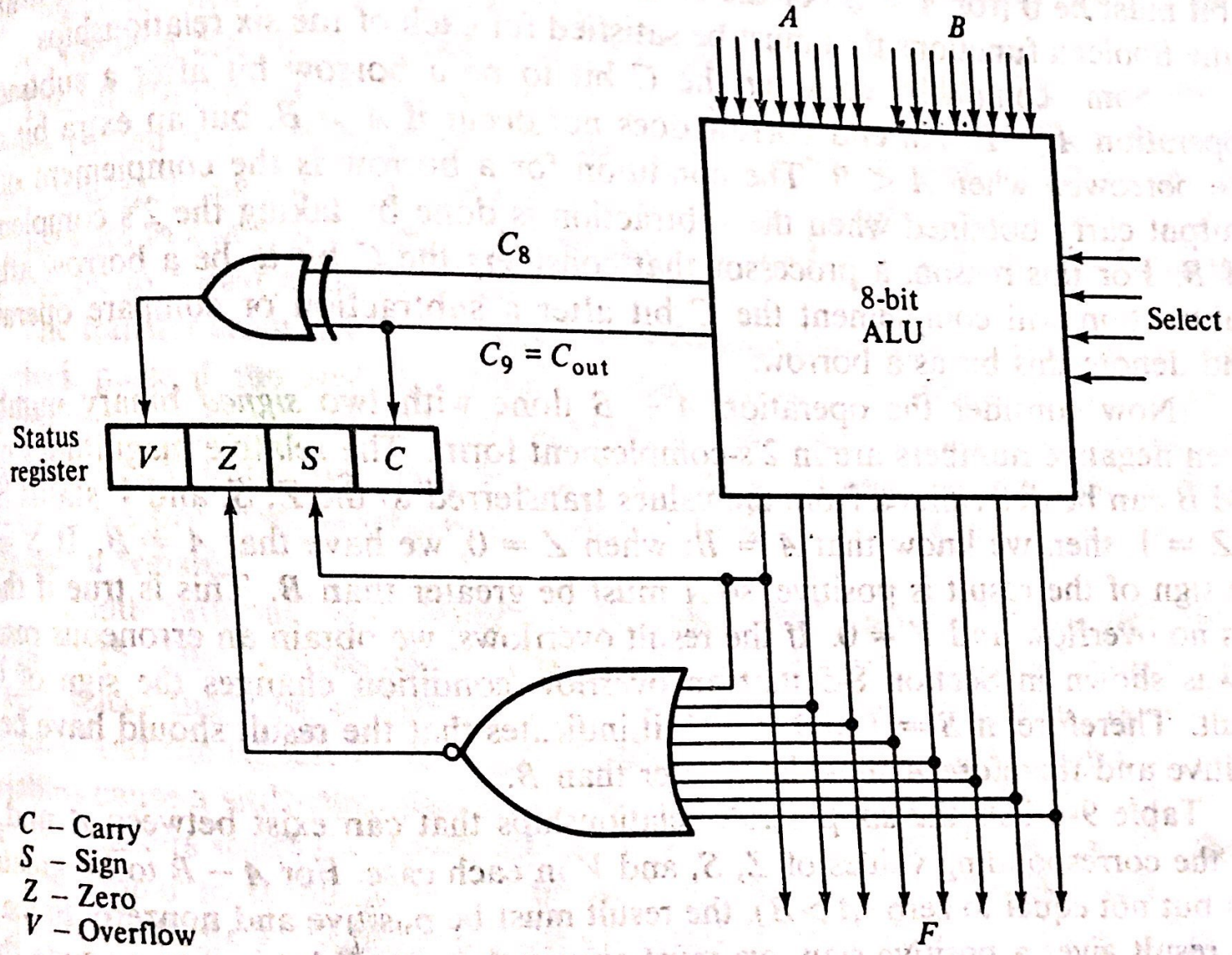


Figure 9-14 Setting bits in a status register

Design a shifter:-

- The shift unit attached to a processor transfers the o/p of the ALU onto the o/p 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 micro-operation commonly not available in an ALU.

- It is a bidirectional shift register with parallel load. The information from the ALU can be transferred to the register in parallel & then shifted to the right or left.

- The transfer from a source register to a destination register can be done with one clock pulse if the shifter is implemented with a combinational ckt.

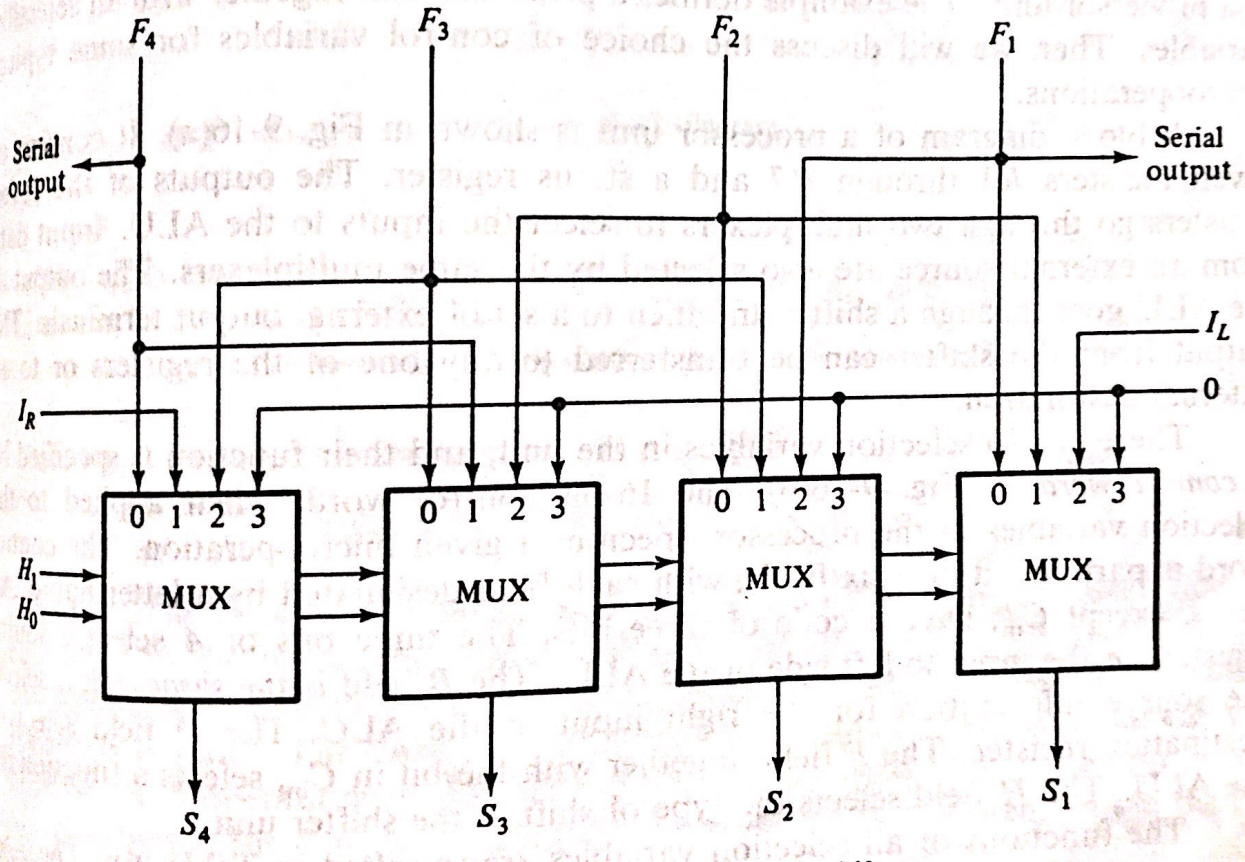


Figure 9-15 4-bit combinational-logic shifter

TABLE 9-7 Function table for shifter

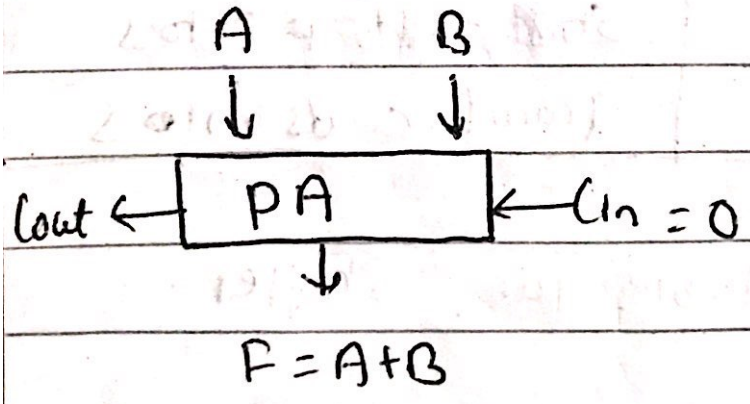
H_1	H_0	Operation	Function
0	0	$S \leftarrow F$	Transfer F to S (no shift)
0	1	$S \leftarrow \text{shr } F$	Shift-right F into S
1	0	$S \leftarrow \text{shl } F$	Shift-left F into S
1	1	$S \leftarrow 0$	Transfer 0's into S

Design of arithmetic logic unit

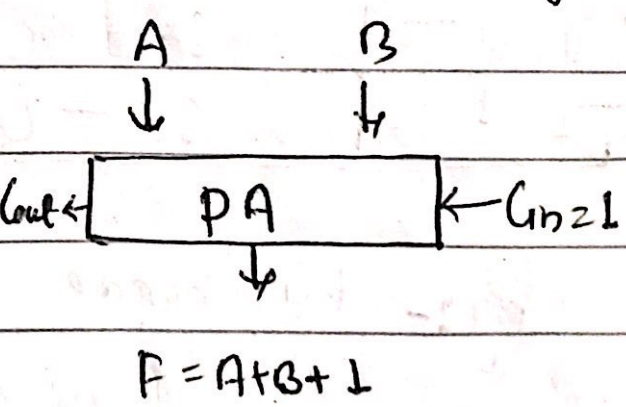
- The basic component of the arithmetic section of an ALU is a parallel adder.

- A parallel adder is constructed with a no. of full adder ckts connected in cascade.

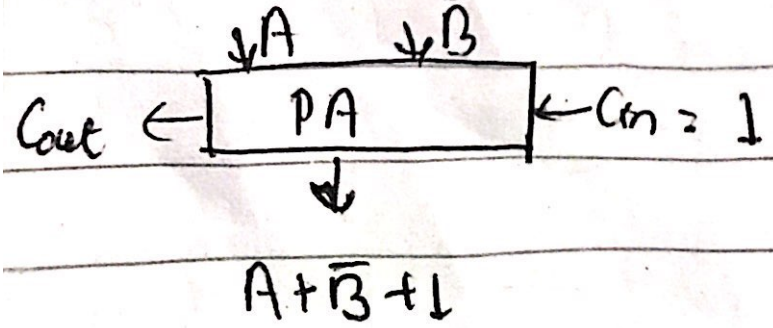
(a) Addition:-



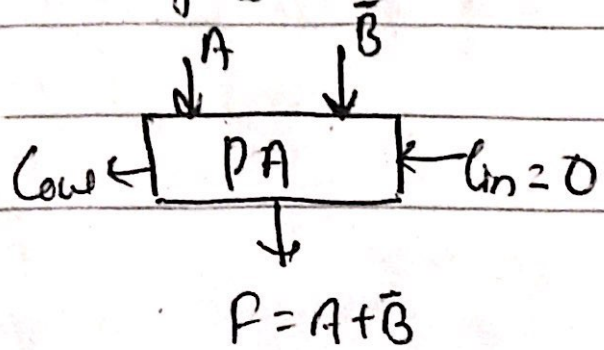
(b) Addition with carry



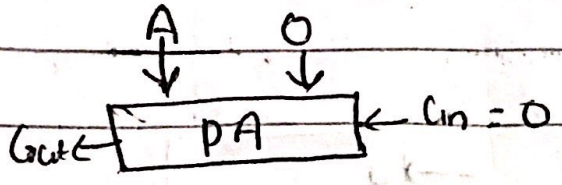
(c) subtraction



(d) A plus 1's complement of B.

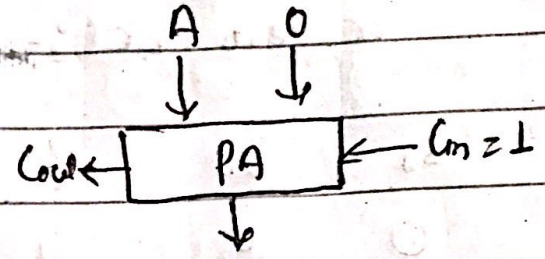


(e) Transfer A



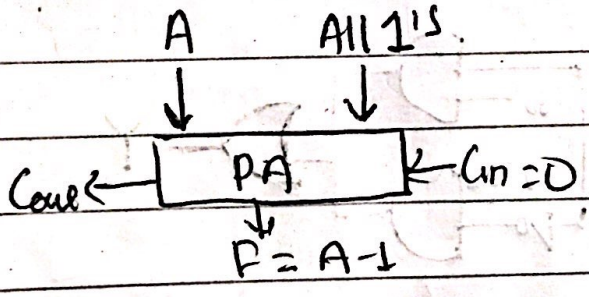
F = A

(f) Increment A



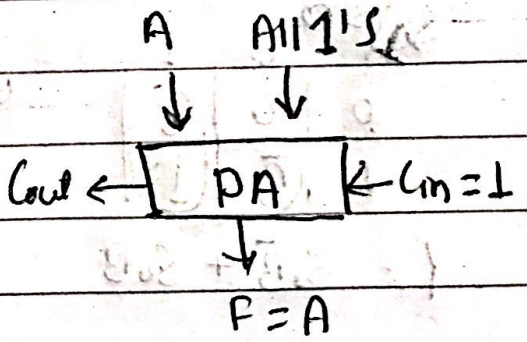
F = A+1

(g) Decrement A:



F = A-1

(h) Transfer A



F = A

Note!-

(1) A - 1

A - $\bar{1}$ + 1 { A-B = A-$\bar{B}$+1 }

A - $\overline{0001}$ + 1

A - 1110 + 1

A - 1111

(2)

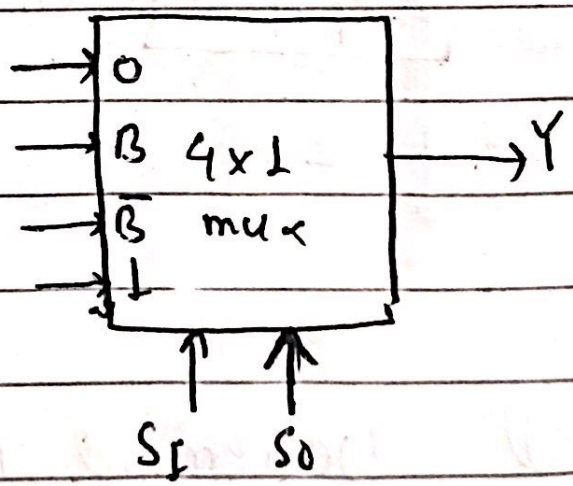
A + 1111

+1

11 0000

Function table for True, complement, one or zero cct :-

S_1	S_0	Y
0	0	0
0	1	B
1	0	$\bar{B}$
1	1	1



S_1 \ S_0	0	1
0	0	B
1	$\bar{B}$	1

$Y = S_1\bar{B} + S_0B$

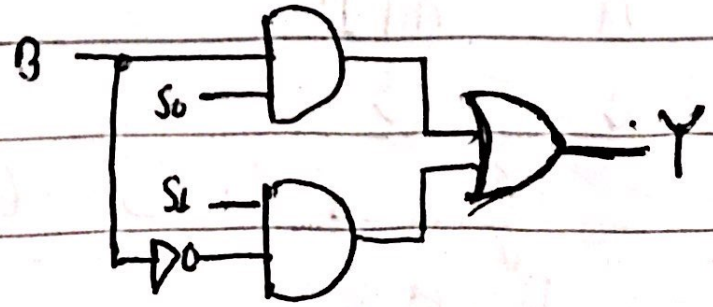


TABLE 9-1 Function table for the arithmetic circuit of Fig. 9-8

Function select			Y equals	Output equals	Function
S_1	S_0	C_{in}			
0	0	0	0	$F = A$	Transfer A
0	0	1	0	$F = A + 1$	Increment A
0	1	0	B	$F = A + B$	Add B to A
0	1	1	B	$F = A + B + 1$	Add B to A plus 1
1	0	0	$\overline{B}$	$F = A + \overline{B}$	Add 1's complement of B to A
1	0	1	$\overline{B}$	$F = A + \overline{B} + 1$	Add 2's complement of B to A
1	1	0	All 1's	$F = A - 1$	Decrement A
1	1	1	All 1's	$F = A$	Transfer A

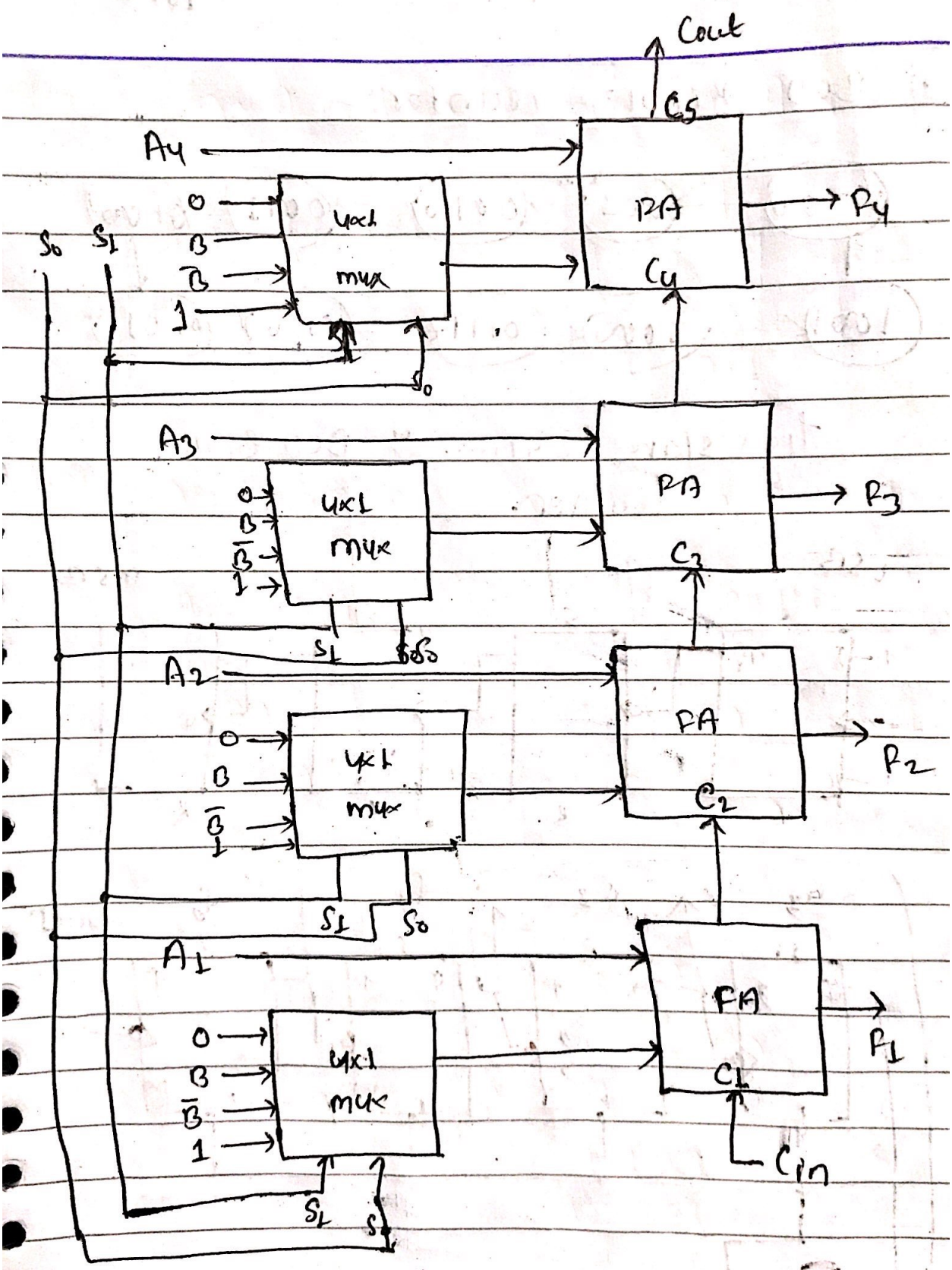


fig: Logic diagram of Arithmetic circuit.

Design of Accumulator:-

- Some processor units distinguish one register from all others & call it an accumulator register.

- The block diagram of an accumulator that forms a sequential ckt is shown in below

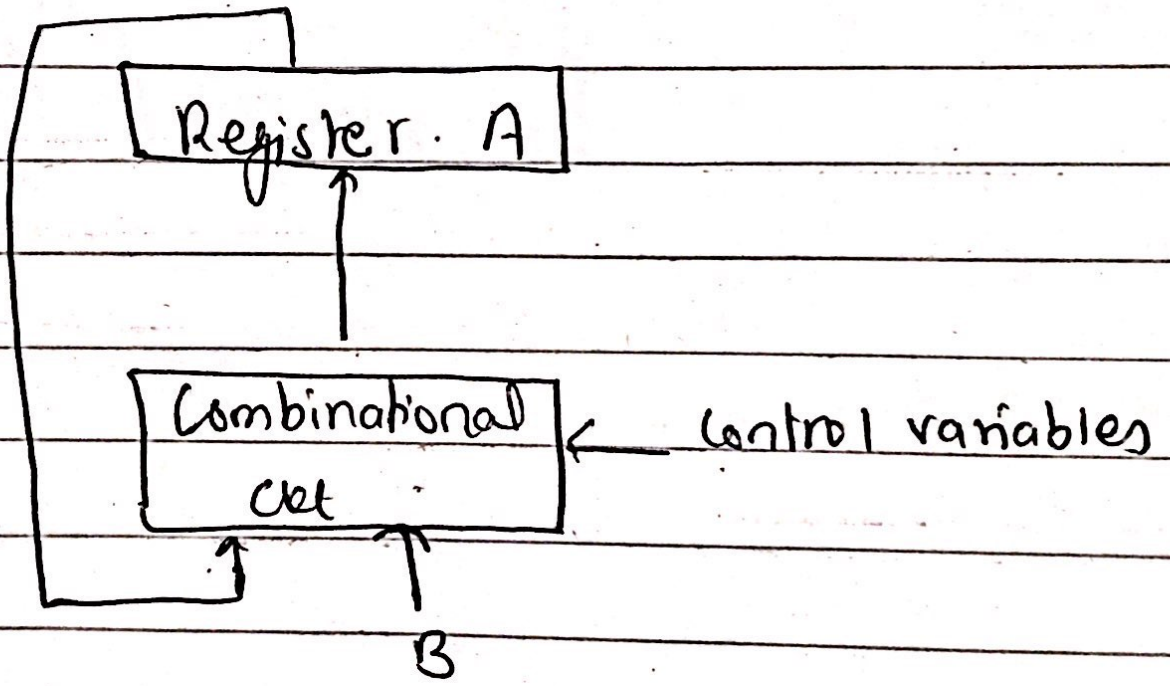


Fig: Block diagram of accumulator.

→ Accumulator register is essentially bi-directional shift register with parallel load which is connected to an ALU.

→ In above figure, 'A' register & the associated combinational circuit constitute a sequential ckt.

→ The combinational ckt replaces the ALU but cannot be separated from the register, since it is only the combinational ckt part of a sequential ckt.

→ The 'A' register is reffered to as the accumulator register & is sometimes denoted by the symbol AC.

- Here, accumulator ~~register~~ refers to both the 'A' register & its associated combinational circuit.

→ The external i/p's to the accumulator are the data i/p's from B & the control variables that determine the microoperations for the register.

- The next state of register 'A' is a fn of its present state & of the external i/p's.