

Arithmetic Logic Unit (ALU)

Arithmetic Logic unit (ALU) is a multioperation, combinational logic digital function that performs set of basic arithmetic operation as well as set of logic operations. Arithmetic logic unit (ALU) receives the information from the registers and performs the given operation specified by the control.

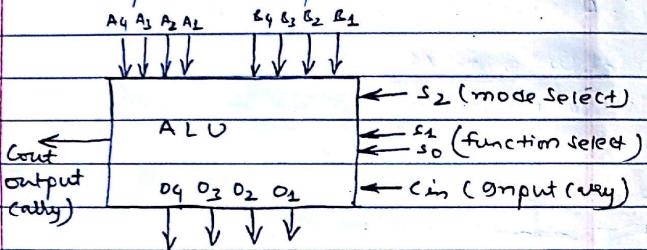


fig: Block output Diagram

The result of the operation is then transferred to a destination Register.

Above figure shows the 9-bit ALU. It has four data inputs 'A' which is combined with the four inputs from B to generate the operation at the output.

Here the mode-select input s_2 distinguishes between Arithmetic

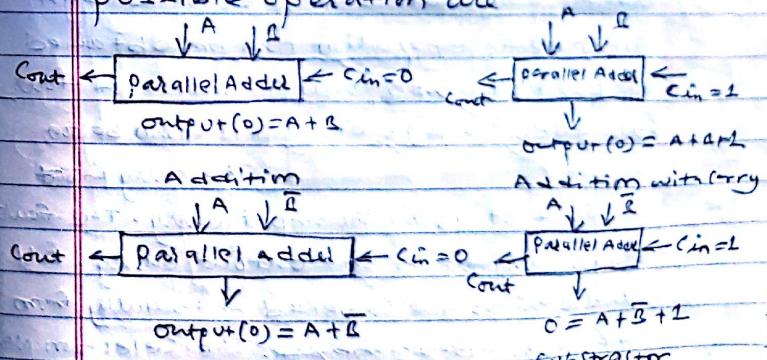
and logic operation. s_1, s_0 input specifies the two function and four logical expression.

There are three stages in ALU Design. They are

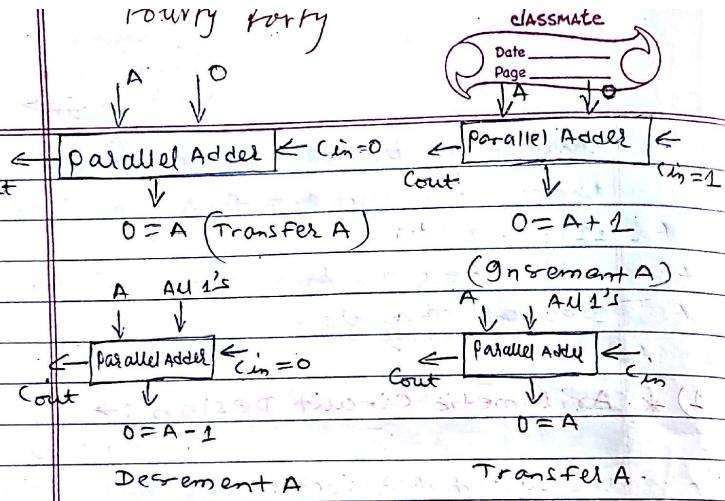
1) * Arithmetic Circuit Design :

In ALU

the Parallel Adder is the basic component of Arithmetic circuit. By controlling the data input to the parallel adder, it is possible to obtain different types of Arithmetic operation. The possible operation are



Fourty forty

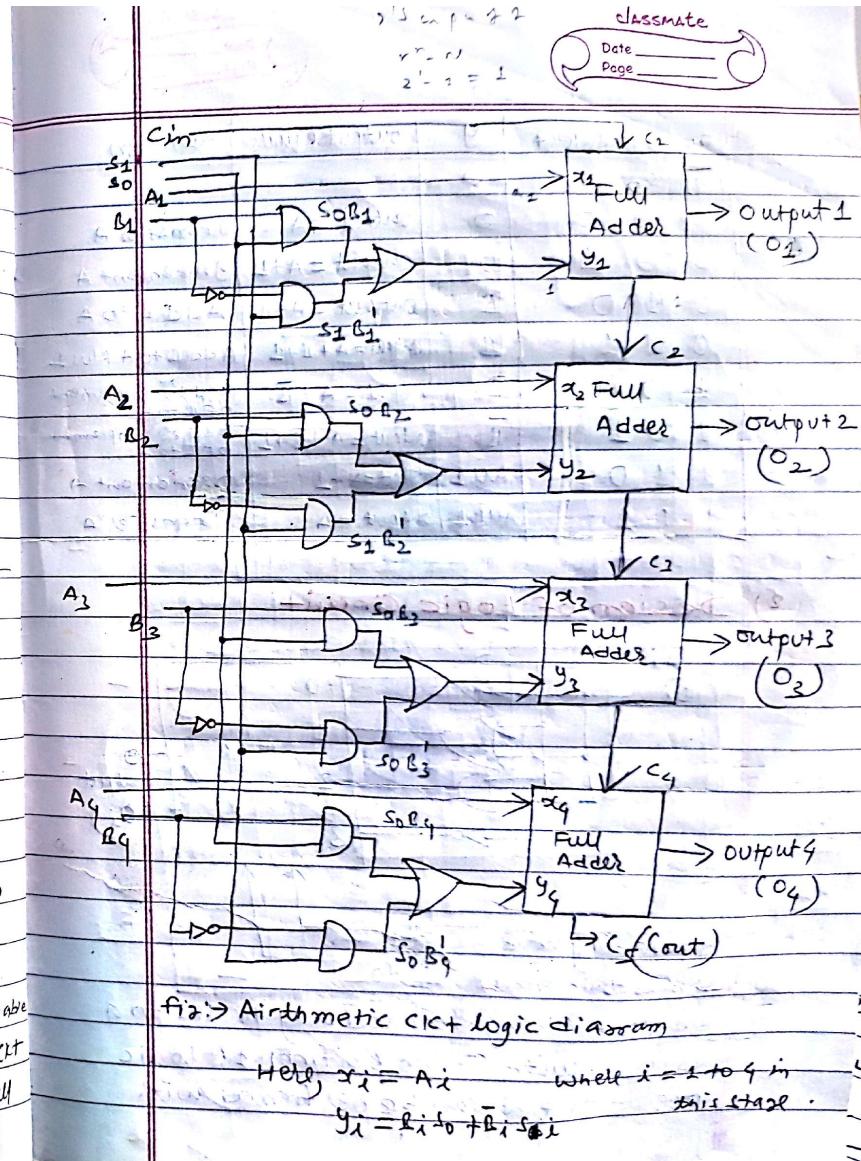


Above arithmetic operation can be obtained by controlling one set of input to parallel adder.

Therefore 4-bit arithmetic ckt that performs eight Arithmetic operation can be obtained by parallel Adder which consist of four Full Adder (F.A) ckt which is shown below.

The carry into

the first stage is the input carry. The fourth stage carry out is the output carry. All other carries are connected internally from one stage to the next. The selection variable s_0 and s_1 control all of 8 inputs to Full Adder (F.A). The input A goes directly to other input of the full Adder.



fix → Arithmetic ckt logic diagram

$$\text{Here, } x_i = A_i \quad \text{where } i=1 \text{ to } 4 \text{ in this stage.}$$

$$y_i = \bar{B}_i \bar{s}_0 + \bar{B}_i s_0$$

Function select	y	output equals	Function
$s_1 s_0 c_i$			
0 0 0	0	output = A	Transfer A.
0 0 1	0	output = A + 1	Increment A.
0 1 0	1	output = A + 1	Add 1 to A.
0 1 1	1	output = A + 1 + 1	Add 1 to A plus 1
1 0 0	1	output = A + \bar{B}	Add 1's complement of B to A.
1 0 1	1	output = A + $\bar{B} + 1$	Add 2's complement of B to A.
1 1 0	All 1	output = A - 1	Decrement A
1 1 1	All 1	output = A	Transfer A

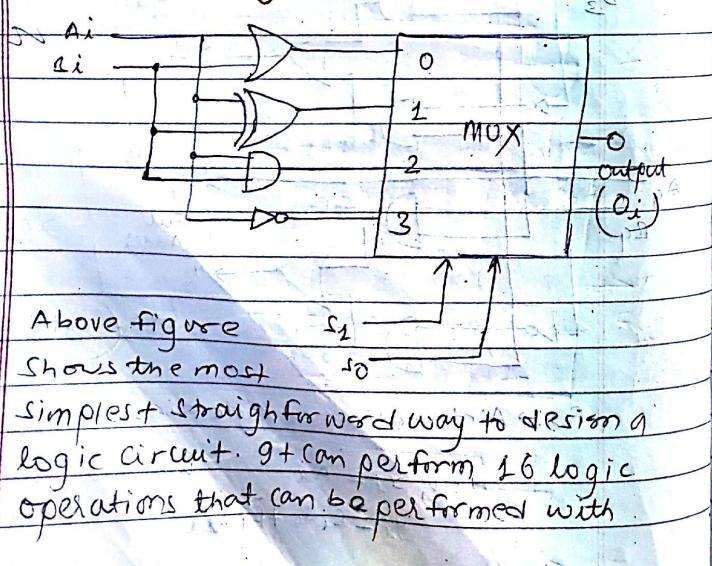
two binary variable g_i 's can be generated in the circuit and selected by means of four selection lines.

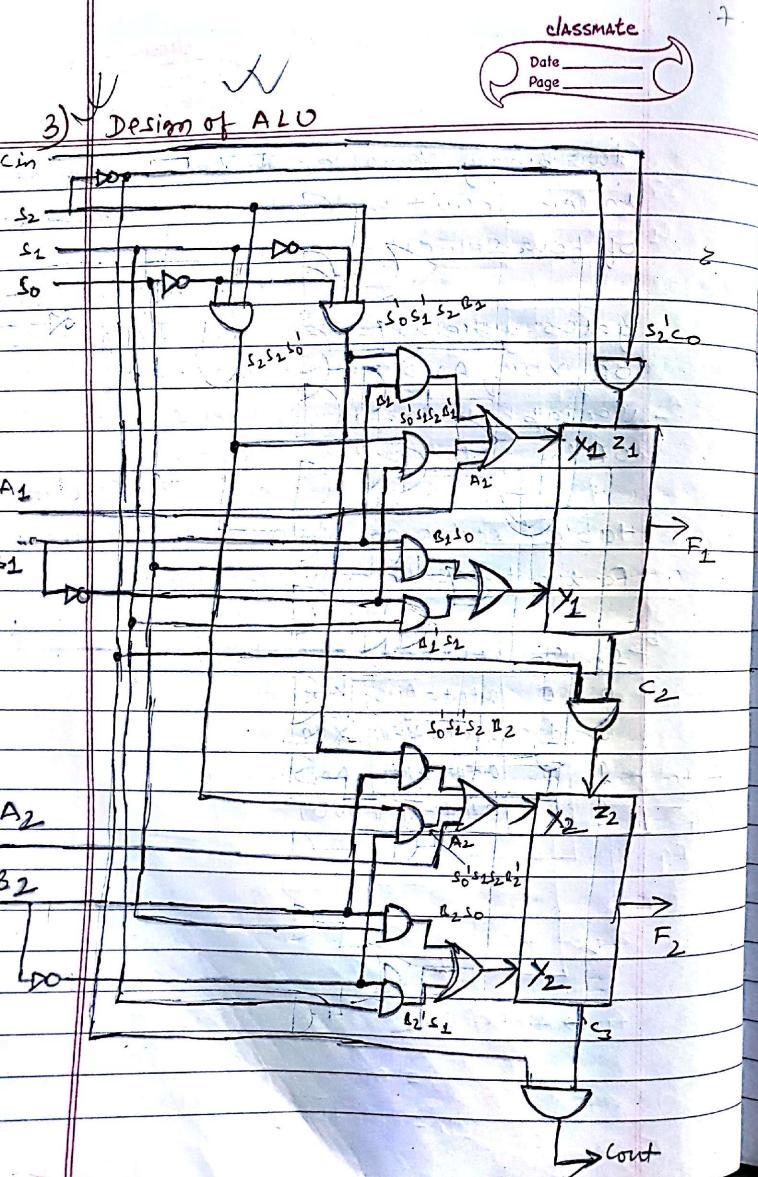
Here the four gates generates four logic operations OR, XOR, AND and NOT. The two selection variable in the multiplexer selects one of the gates for the output.

The function table lists the generated output logic for the two selection variable.

s_1	s_0	output	operation
0	0	output = $A + \bar{B}_i$	OR
0	1	output = $A_i \oplus \bar{B}_i$	XOR
1	0	output = $A_i \bar{B}_i$	AND
1	1	output = \bar{A}_i	NOT

Design of Logic Circuit





Digital logic & Computer

Design Pg No. → 380

Fig. → 9-12

Airthmetic logic unit has eight Airthmetic operation and four logic operation.

Three selection variable S_2, S_1, S_0 select eight different operation. with $S_2 = 0$, selection variable S_1, S_0 together with Cin .

For $S_2 = 1$, variable S_2, S_0 will select the four logic operation OR, XOR, AND and NOT.

above figure shows only two stages of Airthmetic logic unit and can be extended further as well.

The input to each full adder is expressed by Boolean function

$$X_i = A_i + S_2 S_1 S_0 B_i + S_2 S_1 S_0' B_i'$$

$$Y_i = S_0 L_i + S_1 B_i$$

$$Z_i = S_2' C_i$$

When $S_2 = 0$, the three function will be reduced to

$$X_i = A_i, Y_i = S_0 L_i + S_1 B_i, Z_i = C_i$$

are the function of Airthmetic circuit

The logic operation are generated when $S_2 = 1$.

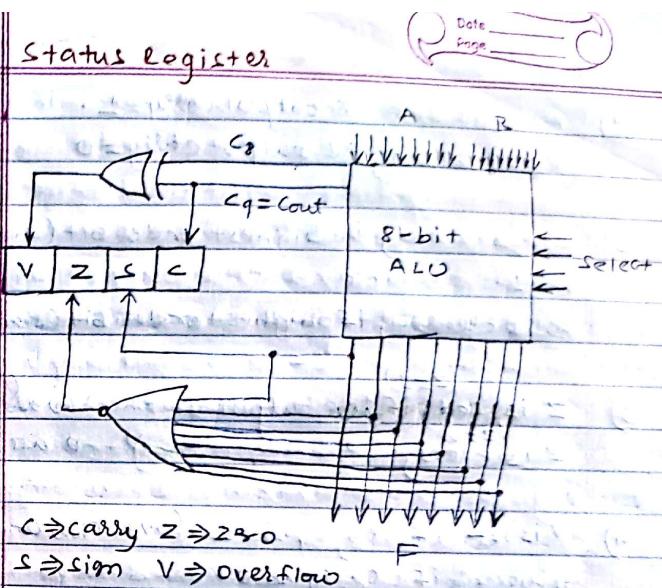
For $S_2 S_1 S_0 = 101$ or 110 , the function reduced to $X_i = A_i, Y_i = S_0 L_i + S_1 B_i, C_i = 0$

therefore 12 operation

generated in ALU are shown below

Selection	Output	Function
$s_2 s_1$	$s_0 \text{ cin}$	
0 0 0	$F = A$	Transfer A
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

* Status register



Above figure shows the block diagram of the 8-bit ALU with a 4-bit status register. The four bit status bits are C, S, Z, V.

Status bits are added to ALU to identify the status of bit condition. Status register determines the magnitude of two given numbers. As numbers are signed and unsigned.

For unsigned number

through two multiplexers to select the input to the ALU.

Input data from an external source are also selected by same multiplexer. The output of the ALU goes through the shifter and then to a set of external output terminal.

The output from the shifter can be transferred to any one of the register or to external destination.

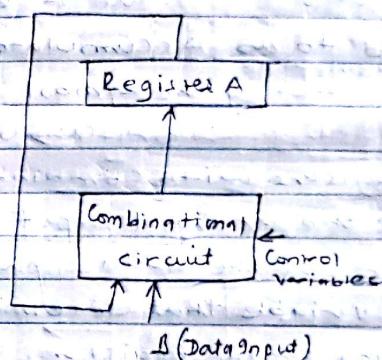
There are 16 selection variables in the unit and their function is specified by the control word. The 16 bit control word when applied to a selection variable in the processor specifies a given microoperation.

The three bits of A select a source Register for the input to the left side of the ALU. The 3-bits of B select the source information for the right i/p of the ALU.

The D-fields select a destination register. The F-field together with Cin bit select the function

for the ALU and H-field selects the type of shift in the shifter unit.

* Design of Accumulator:



Accumulator is a multipurpose register which performs microoperation and distinguishes one register to another.

The above figure shows the block diagram of Accumulator. Here Accumulator is a bidirectional shift Register with a parallel load connected to ALU. An Accumulator ckt is designed by sequential ckt techniques instead of using a Combinational ckt ALU.

15

classmate
Date _____
Page _____

From the figure we see that the combinational CKT replaces the ALU with register A which is the part of overall sequential CKT.

Here 'A' register is referred to as Accumulator Register sometimes and is denoted by the symbol 'AC' also.
The Accumulator is the Registers with combinational CKT. Here the external input to the accumulator all the data input from I and control variable that determines the microoperation for the Register. The microoperation produced by Accumulator is

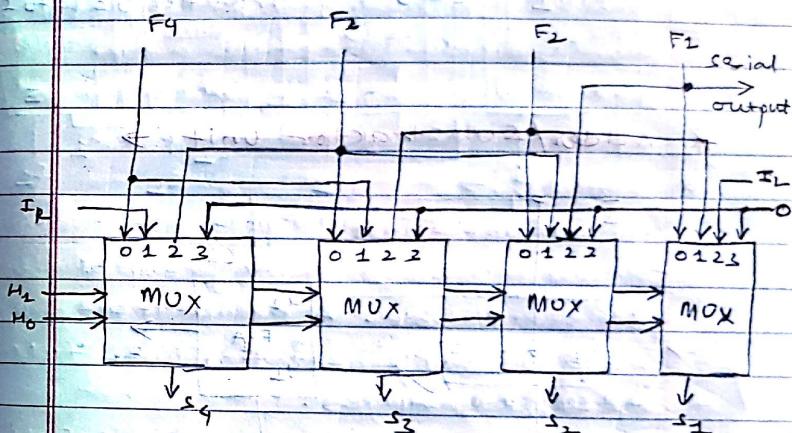
Control Variable	Microoperation	Name
P ₁	A $\leftarrow A + B$	Add
P ₂	A $\leftarrow 0$	clear
P ₃	A $\leftarrow \bar{A}$	Complement
P ₄	A $\leftarrow A \cdot B$	AND
P ₅	A $\leftarrow A \vee B$	OR
P ₆	A $\leftarrow A \oplus B$	EX-OR
P ₇	A $\leftarrow \text{Shr } A$	Shift Right
P ₈	A $\leftarrow \text{Shl } A$	Shift Left

16

classmate
Date _____
Page _____

P₉ A $\leftarrow A + 1$ Increment
I.F(A=0) ? (Z=1) Check for zero.

* Design of shifter



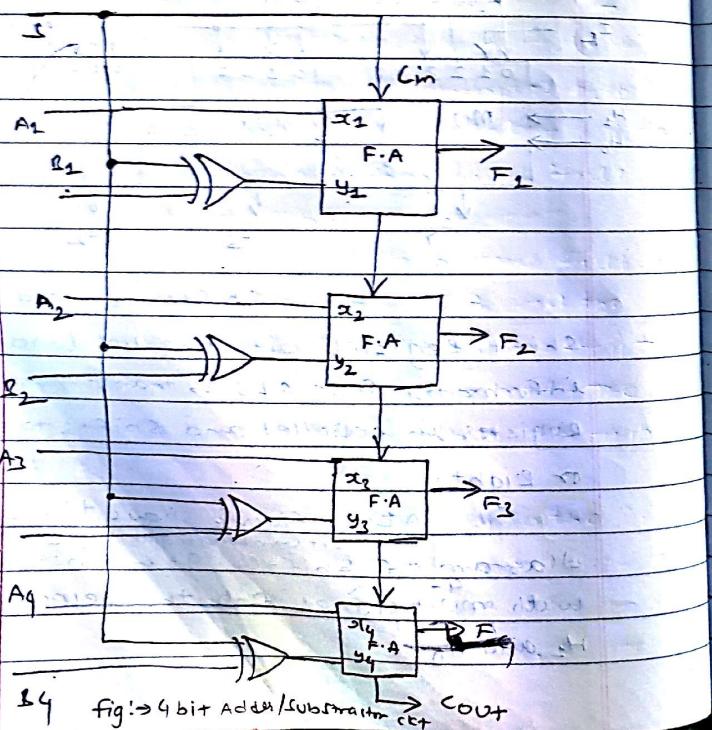
shifter is a bidirectional shift register with parallel load. The information from ALU is transferred to the register in parallel and shifts to the left or right.

Above figure shows the block diagram of shifter. It is constructed with multiplexer and two selecting variables H₂ and H₃.

The operation of Shifter is

H_2	H_0	operation	function
0	0	$S \leftarrow F$	Transfer F to S (no shift)
0	1	$S \leftarrow \text{sh}r F$	Shift right F into S.
1	0	$S \leftarrow \text{sh}l F$	Shift left F into S.
1	1	$S \leftarrow 0$	Transfer 0's into S.

Adder/Subtractor Unit :



Above figure shows 4-bit Adder/Subtractor circuit. Here input b_i is passed through to the Full Adder. The selection variable goes to the input of each gate and also to the input carry of the parallel Adder. Here,

when $s=0$, $Cin=0$
then $x_i = A_i$, $y_i = b_i$ which then performs addition ($A_i + B_i$)

For $s=1$, $Cin=1$

Here $x_i = A_i$

$y_i = b_i \oplus s = b_i$ which is then introduced to the Full Adder with input carry 1. This performs subtraction ($A_i - B_i$)

* Nibble Adder

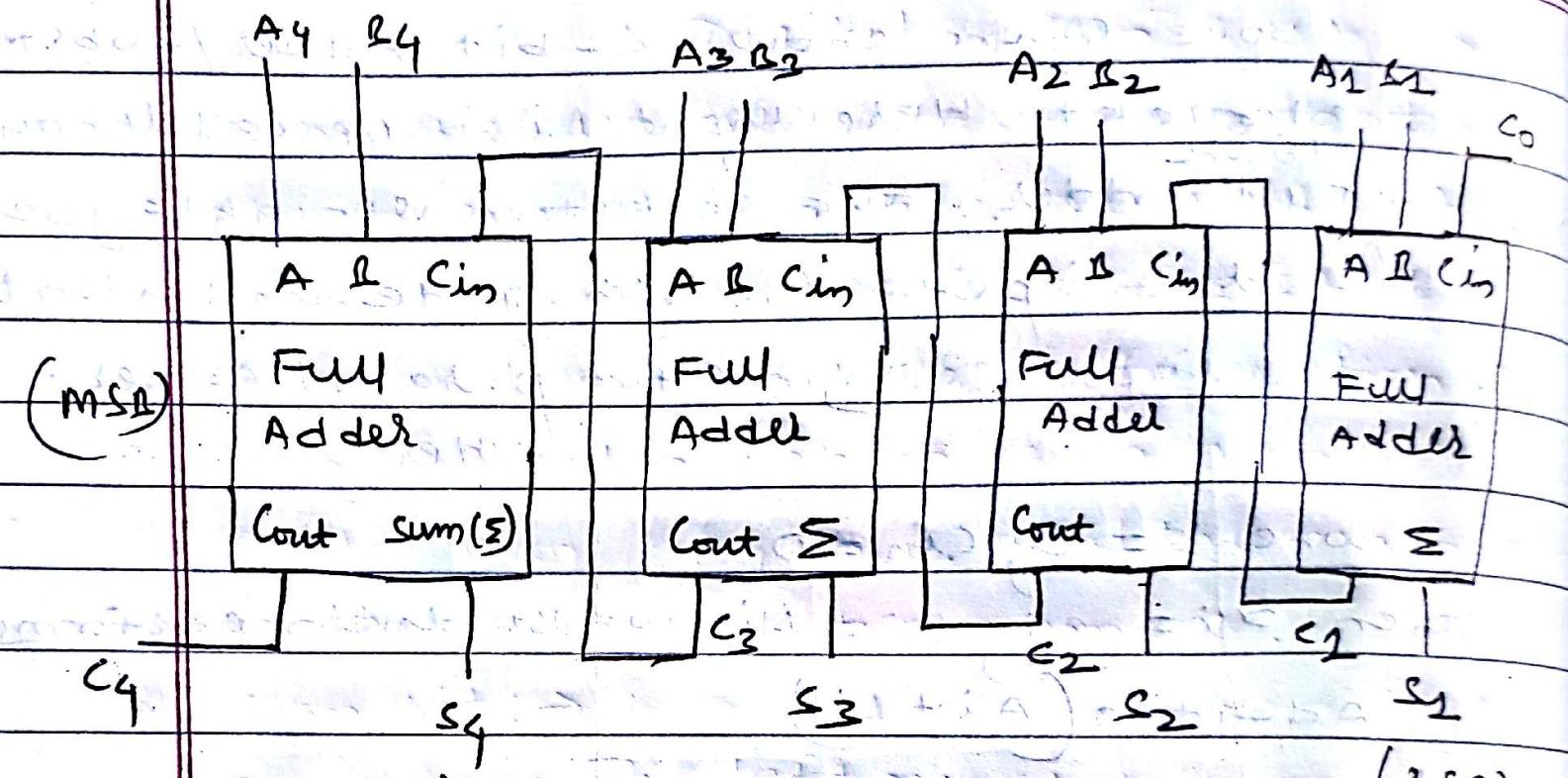


fig: → Nibble Adder (2's R)

A group of four bit is a Nibble. A basic 4-bit parallel Adder is implementation with full Adder (F.A) stages.

It has four full Adders. The Augend bits of A and the addend bits of B denoted by subscript number $(1, 2, 3, 4)$ from right to left. The carries are connected in a chain through the Full Adder.

The input carry to the Adder is C_0 and output carry is C_4 and generates the required sum bit ' Σ '.