Project: Data-Path

Course: Computer Architecture

Dr. Khalid Elshafey

Participance:

1- Sherif Gamal Abdelatif (45)
2- Abdelrahman Farouk Sayed (52)
3- Elsayed Khalid Elsayed Elrefaey (32)
4- Hossam Shaaban Sayed (38)
5- Attia Sayed Attia (60)
6- Ahmed Sedek Ali (10)

Objective:

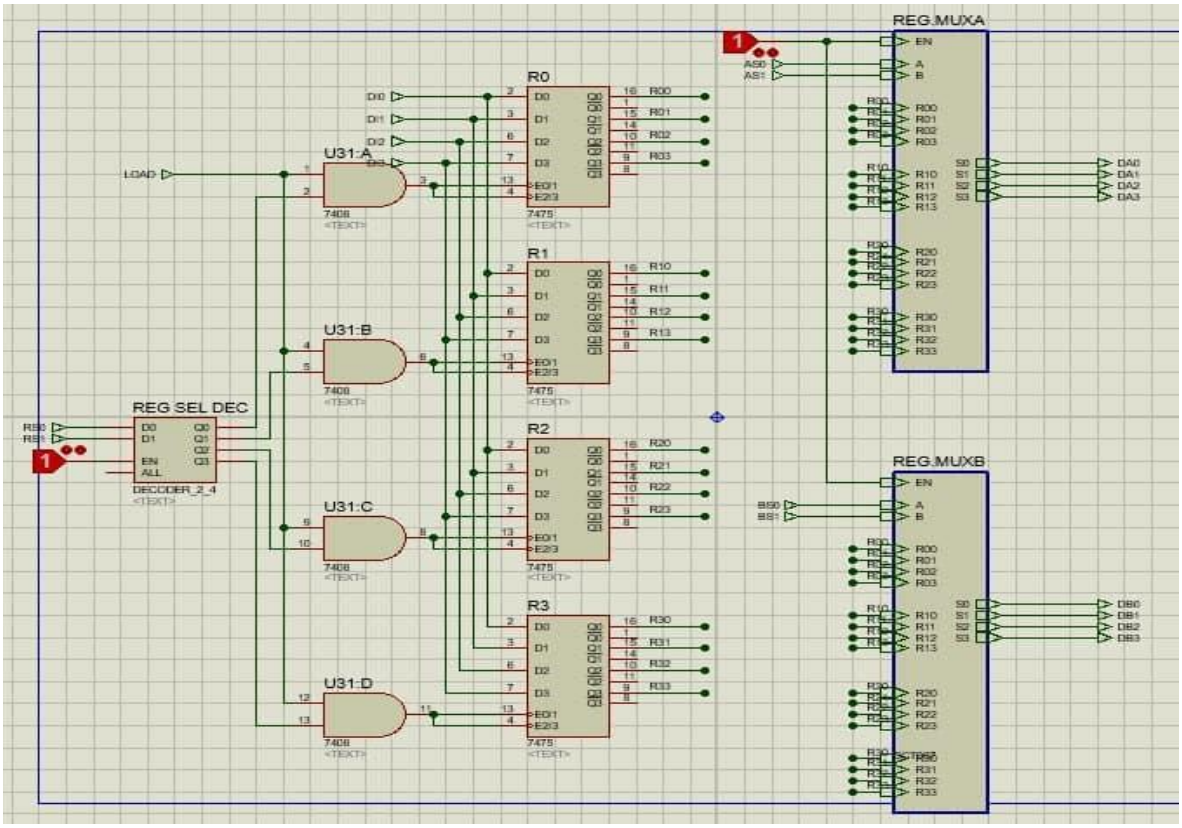
Building a data-path which includes the following parts:

- Register File
- ALU

Tools: Proteus simulator

Design: -

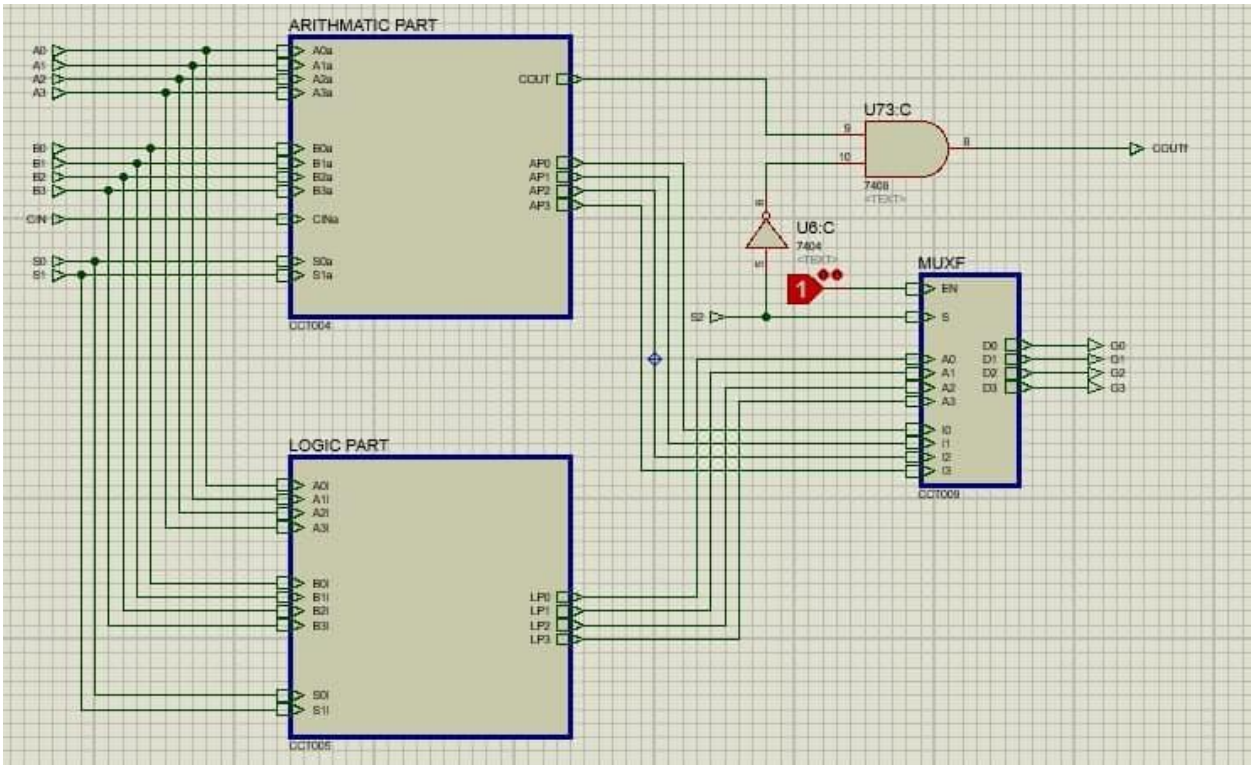
1- Register file design:



The register file contains:

- Reg Decoder (2x4).
- Four (4-bit) Registers.
- Two (4x1) registers' Multiplexers; one for bus A and the other is for bus B.

2- ALU: -



The ALU has: two (4-bit) inputs, Carry input, and Three select lines.

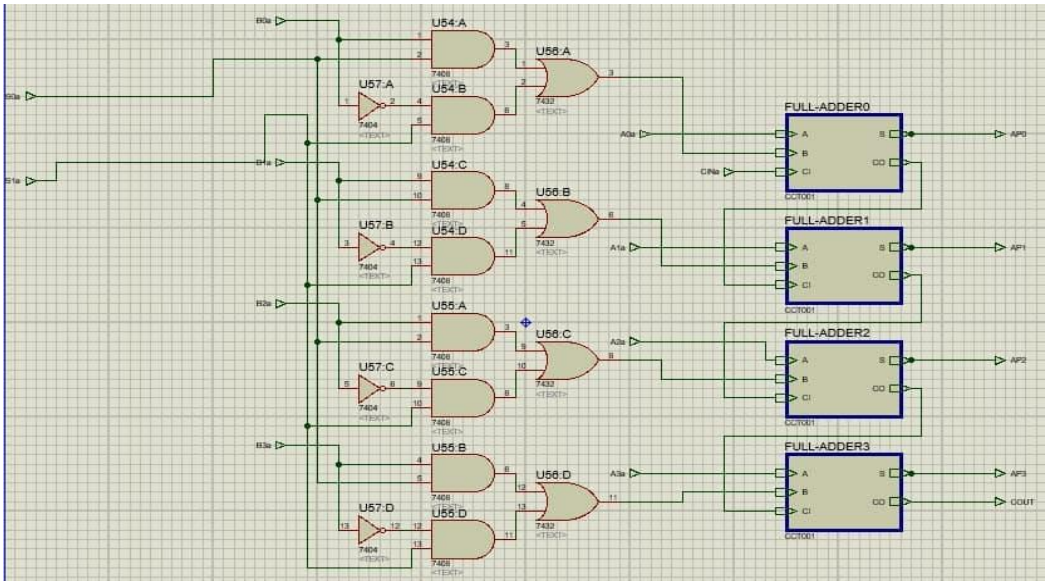
Following is the functional table of the ALU:

FUNCTION TABLE

	S2	S1	S0	CIN	OPERATION
Arithmetic	0	0	0	0	$G = A$
	0	0	0	1	$G = A + 1$
	0	0	1	0	$G = A + B$
	0	0	1	1	$G = A + B + 1$
	0	1	0	0	$G = A + \overline{B}$
	0	1	0	1	$G = A + \overline{B} + 1$
	0	1	1	0	$G = A - 1$
	0	1	1	1	$G = A$
Logic	1	0	0	X	$G = A \wedge B$
	1	0	1	X	$G = A \vee B$
	1	1	0	X	$G = A \oplus B$
	1	1	1	X	$G = \overline{A}$

ALU consists of two main parts {Arithmetic, Logic}

- Arithmetic Part: -



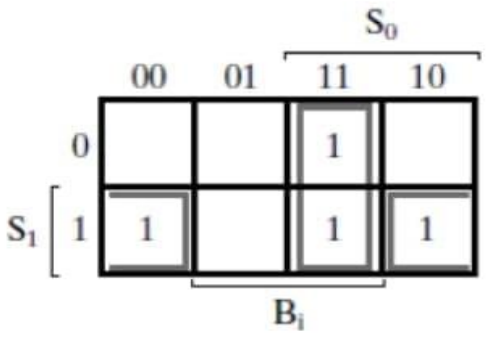
As shown in the previous image, the arithmetic Part consists of, one Full adder for each bit of the input; and the B input logic which is derived from the following table:

TABLE 1
Function Table for Arithmetic Circuit

Select		Input	$G = (A \mid Y \mid C_{in})$	
S_1	S_0	Y	$C_{in} = 0$	$C_{in} = 1$
0	0	all 0s	$G = A$ (transfer)	$G = A + 1$ (increment)
0	1	B	$G = A + B$ (add)	$G = A + B + 1$
1	0	$\overline{B}$	$G = A + \overline{B}$	$G = A + \overline{B} + 1$ (subtract)
1	1	all 1s	$G = A - 1$ (decrement)	$G = A$ (transfer)

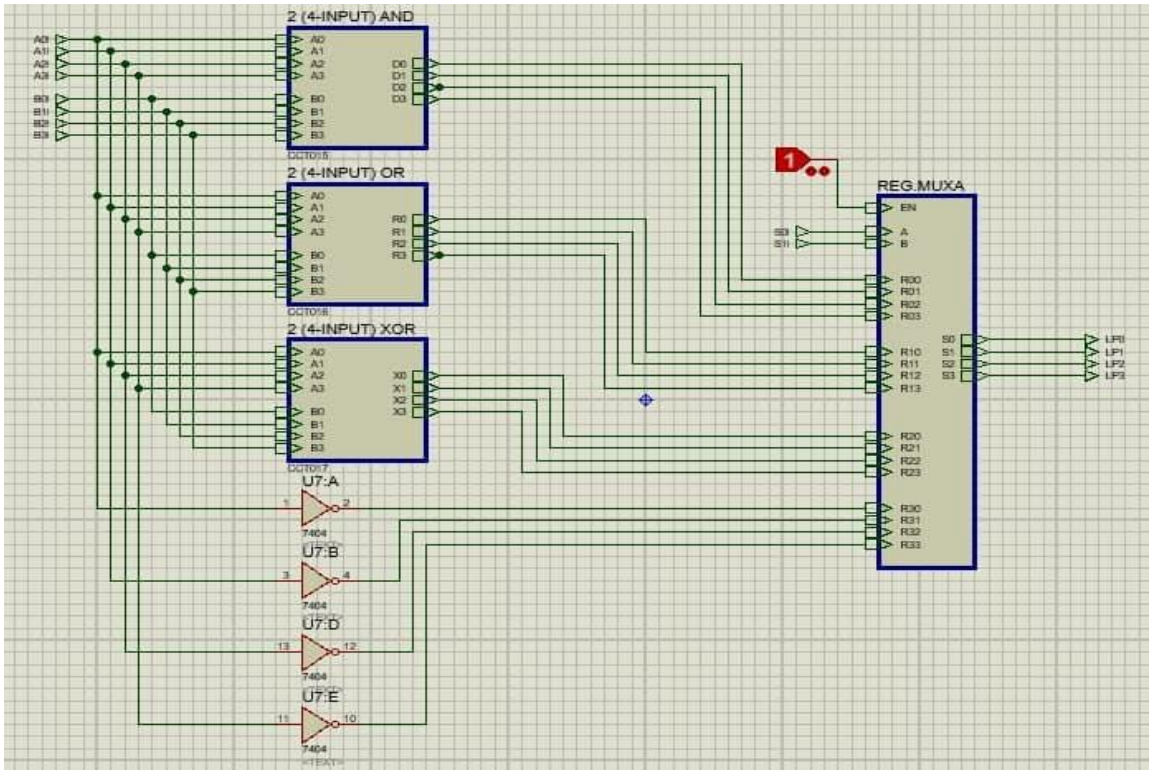
Inputs			Output
S_1	S_0	B_i	Y_i
0	0	0	0 $Y_i = 0$
0	0	1	0
0	1	0	0 $Y_i = B_i$
0	1	1	1
1	0	0	1 $Y_i = \overline{B}_i$
1	0	1	0
1	1	0	1 $Y_i = 1$
1	1	1	1

(a) Truth table



(b) Map simplification:
$Y_i = B_i S_0 + \overline{B}_i S_1$

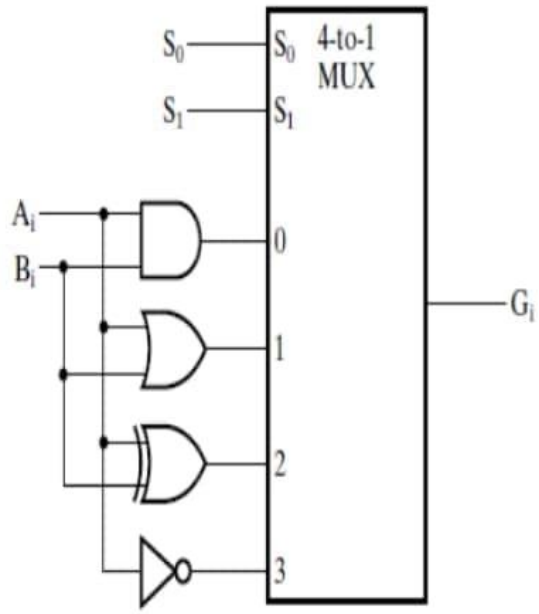
- Logic Part: -



As shown in the previous image, the Logic Part consists of:

- Two (4-bit) input AND
- Two (4-bit) input OR
- Two (4-bit) input XOR
- Four NOT gates
- (4x1) Registers' Multiplexer

The Logic part circuit is derived from the following table:



(a) Logic diagram

S_1	S_0	Output	Operation
0	0	$G = A \wedge B$	AND
0	1	$G = A \vee B$	OR
1	0	$G = A \oplus B$	XOR
1	1	$G = \overline{A}$	NOT

(b) Function table

Steps: -

1- Connect Bus A, to the first input of the ALU
2- Connect Bus B, to the first input of MUX B which selects either of Bus B or Constant input (which is connected to the second input of the MUX B), to be the second input of the ALU.
3- Connect the output of the ALU, to the first input of MUX D, which selects either the

ALU output, or the Data-in (which is coming from the input probes), to be the input of the Register file

---

Code: -

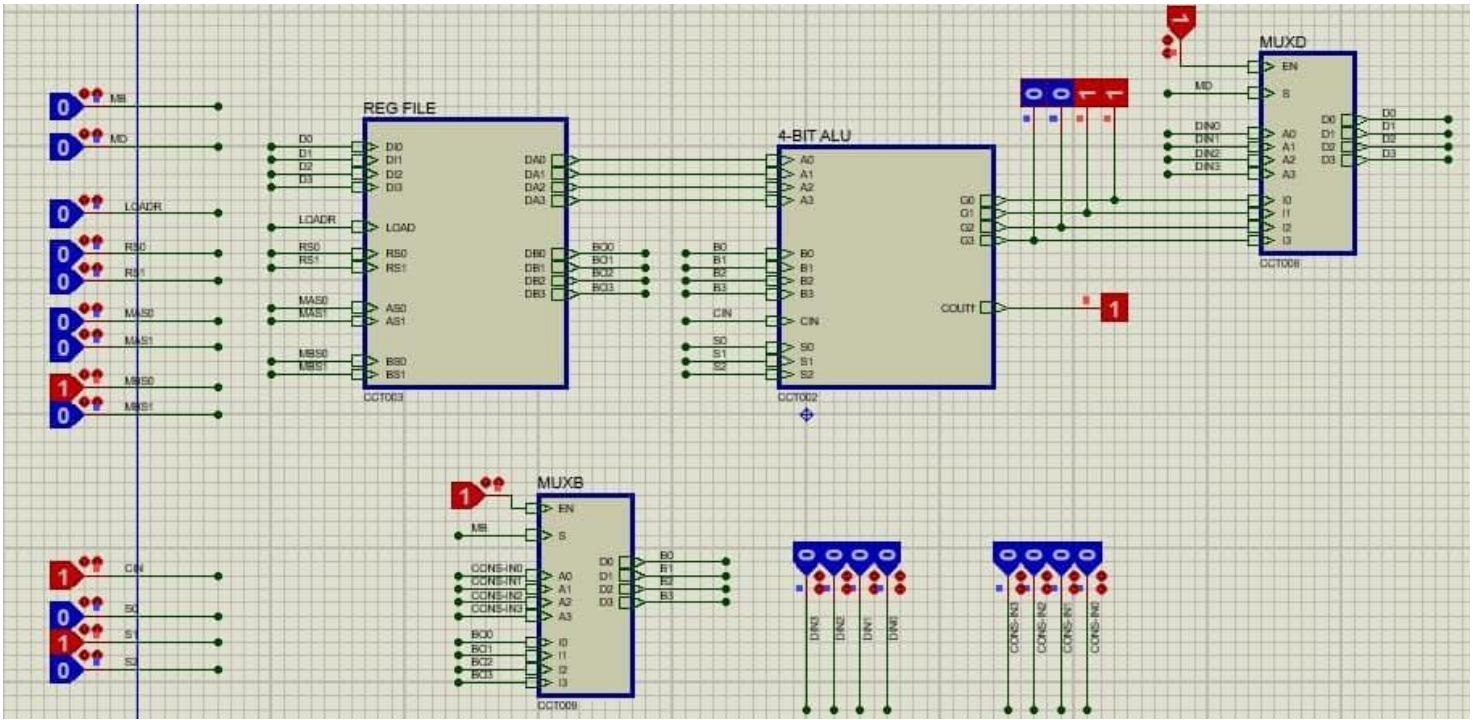
$R0 \leftarrow 8$

$R1 \leftarrow 5$

$R2 \leftarrow R0 - R1$

$R3 \leftarrow R0 \oplus R2$

- *Subtraction Result*: (8 - 5)



- *XOR Result*: (8 ⊕ (8 - 5))