Introduction:

An arithmetic logic unit (ALU) is a multi-operation, combinational logic digital function. It can perform a set of basic arithmetic operations. The ALU has a 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 upto 2^K distinct operations.

We have three selection variables (cs) in our experiment which can enable us to perform $2^3 = 8$ distinct operations. The four data inputs from A are combined with the four inputs from B to generate an operation at the F outputs. The selection variable cs2 distinguishes between arithmetic and logical operations while ~~cs0~~ cs1 is used as an input carry in our ALU design.

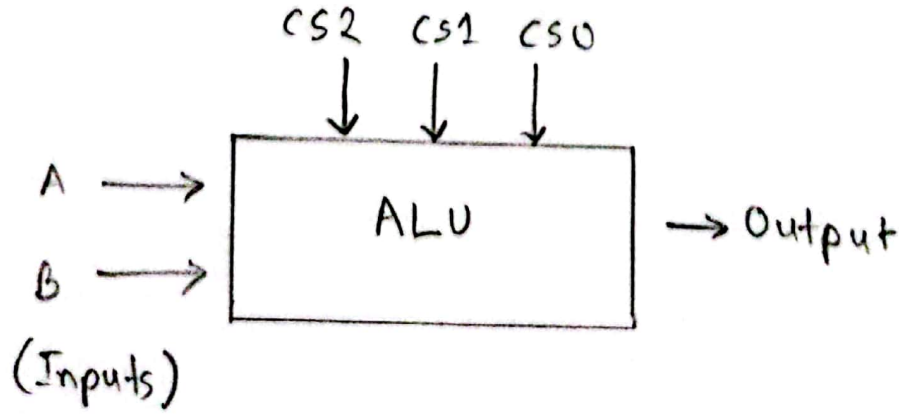
In our ALU design, we use a 4-bit status register. This status register contains 4 status bits that are denoted by C(carry), S(Sign), V(Overflow) and Z(Zero). These status bits change during the arithmetic operations. They indicate the following changes:

- Carry flag (CF): Bit C is set when the output carry of the ALU is 1, otherwise it is 0.
- Sign flag (SF): Bit S is set when the highest order bit of the output of the ALU is 1, it is 0 when the highest order bit is 0.
- Overflow flag (OF): Bit V is set if the XOR of carries C_4 and C_5 is 1, otherwise it is 0.
- Zero flag (ZF): Bit Z is set if the result is 0, otherwise it is 0.

Problem specification:

Design a 4-bit ALU with three selection bits cs2, cs1 and cs0 for performing the following operations:

cs2	cs1(Cin)	cs0	Functions
0	0	0	Subtract with borrow
0	0	1	Transfer A
0	1	0	Subtract
0	1	1	Increment A
1	X	0	AND
1	X	1	OR



Truth table and required k-maps:

1) Truth table for the functions:

S2	S1(Cin)	S0	F	X_i	Y_i	Z_i
0	0	0	A-B-1	A	B′	0
0	0	1	A	A	0	0
0	1	0	A-1	A	B′	1
0	1	1	A+1	A	0	1
1	X	0	AB	A+B′	B′	0
1	X	1	A+B	A+B	0	0

For the arithmetic operations, we take

S1	S0	Y_i
0	0	B_i'
0	1	0
1	0	B_i'
1	1	0

S1	S0	X_i
0	0	A_i
0	1	A_i
1	0	A_i
1	1	A_i

2) Truth table for determining X_i:

For the logical operations ($S2=1$), we have,

S2	S1	S0	X_i	Y_i	Z_i	Obtained F	Required F
1	0	0	A_i	B_i'	0	$A_i \oplus B_i'$	$A_i B_i$
1	0	1	A_i	0	0	A_i	$A_i + B_i$
1	1	0	A_i	B_i'	0	$A_i \oplus B_i'$	$A_i B_i$
1	1	1	A_i	0	0	A_i	$A_i + B_i$

For $S2S1S0 = 101$ and 111, we require

$X_i = A_i + B_i$ so that.

$X_i \oplus Y_i = A_i + B_i$

For $S2S1S0 = 100$ and 110,

$$A_i B_i = (A_i + P) \oplus B_i'$$

$$= (A_i + P)' B_i' + (A_i + P) B_i$$

$$= A_i' P' B_i' + A_i B_i + P B_i = A_i B_i + A_i B_i' P' + B_i P.$$

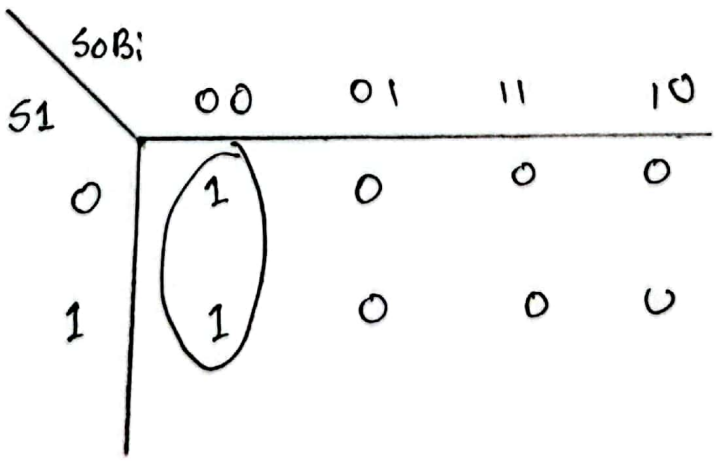
For $P = B_i'$, we have $X_i \oplus Y_i = A_i B_i$.

$\therefore$ we require $P = B_i'$ for $S_2 S_1 S_0 = 100$ and 110.

3) Truth table for determining Y_i:

S1	S0	B_i	Y_i
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

K-map:



$Y_i = S_0' B_i'$

Combining the arithmetic and logical operations,

$$X_i = A_i + S_2 S_0' (A_i + B_i') + S_2 S_0 (A_i + B_i)$$

$$= A_i + A_i S_2 S_0' + A_i S_2 S_0 + B_i' S_2 S_0' + B_i S_2 S_0.$$

$$= A_i (1 + S_2 S_0' + S_2 S_0) + B_i' S_2 S_0' + B_i S_2 S_0.$$

$$= A_i + S_2 (B_i' S_0' + S_0 B_i)$$

$$= A_i + S_2 (S_0 \odot B_i)$$

4) Truth table for determining Z_i:

S2	S1	S0	Z_i
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

K-map:

S_2 \ S_1S_0	00	01	11	10
0	0	0	1	1
1	0	0	0	0

$Z_i = S_2'S_1$

Block diagrams:

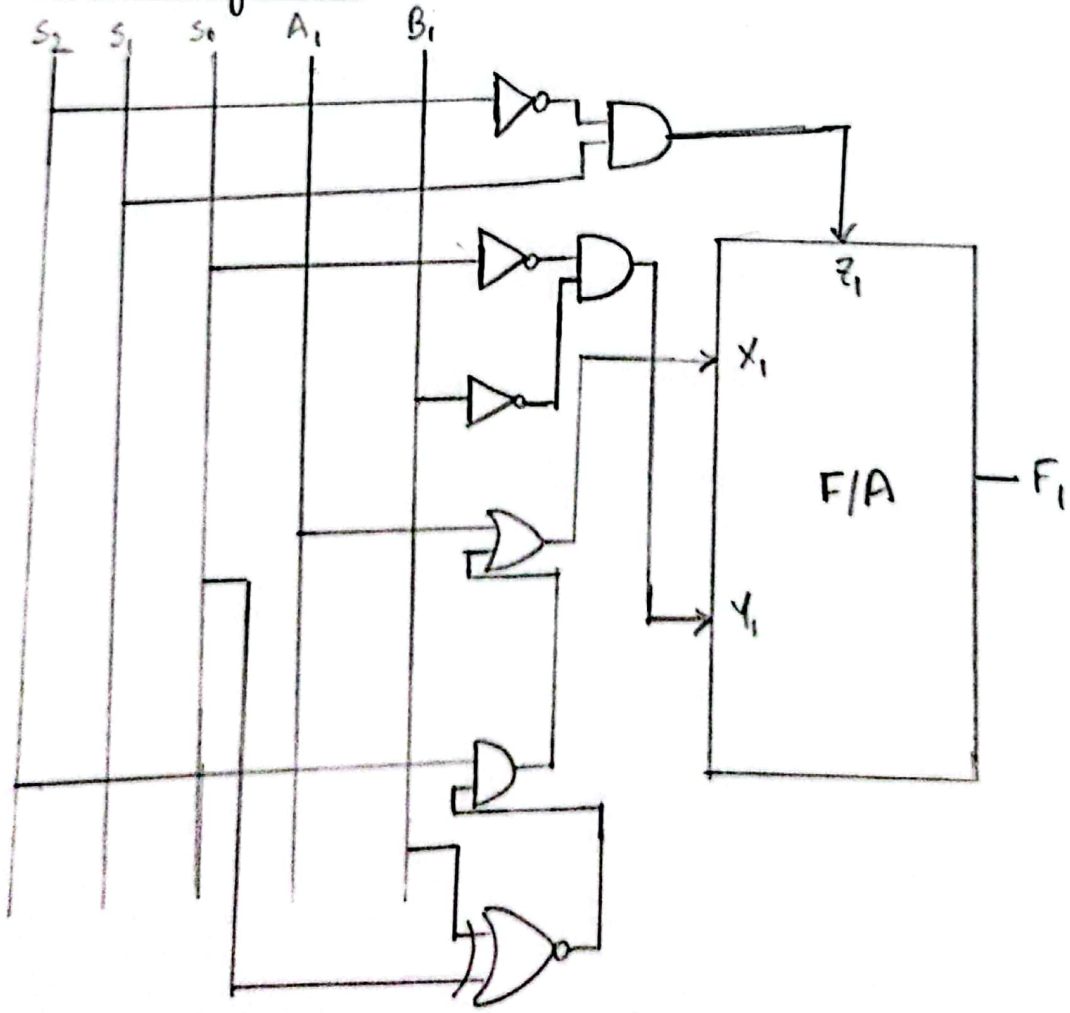


Fig: Logic diagram of ALU

ii) Status register:

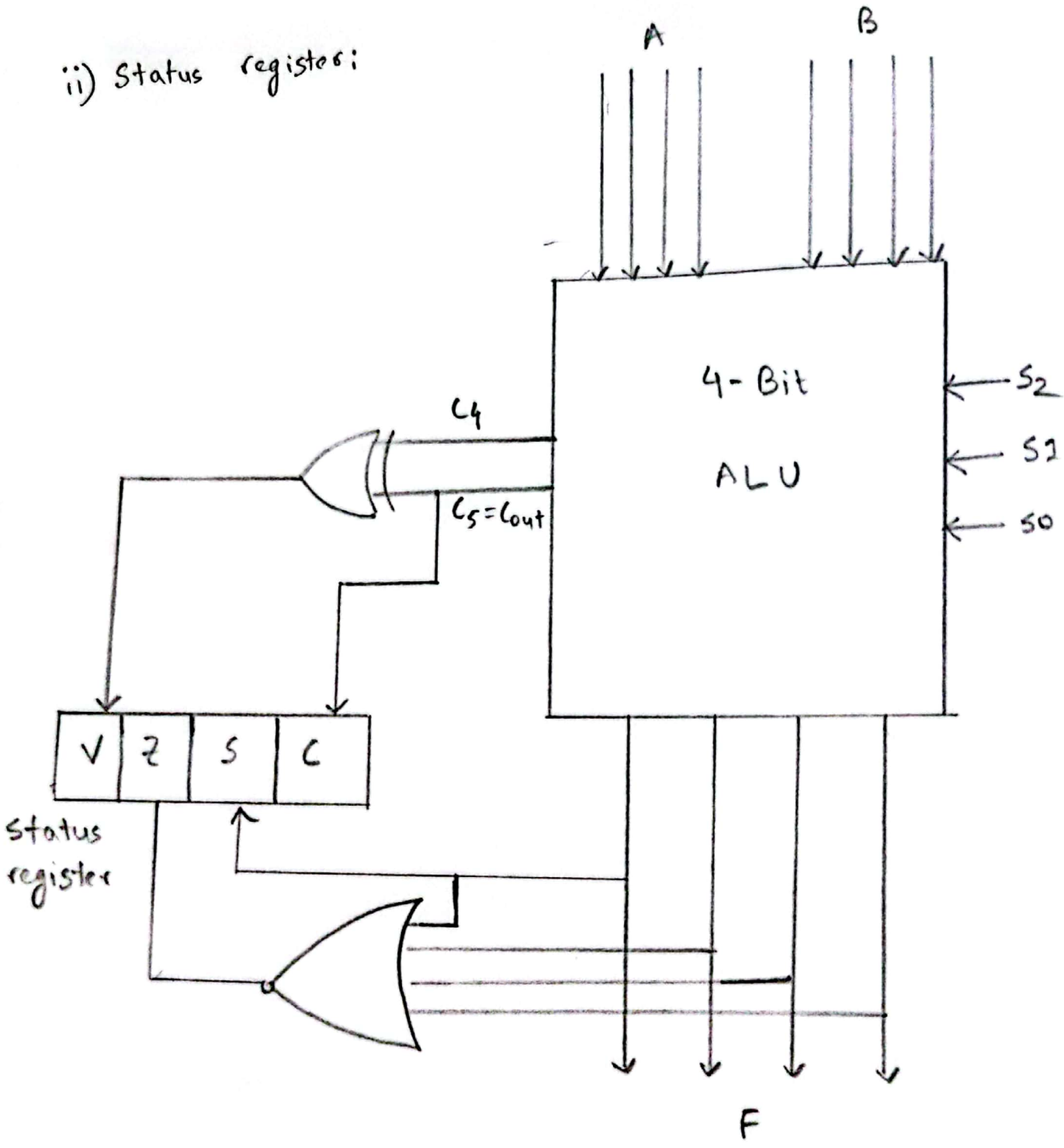


Fig: Block diagram of a status register.

Required ICs:

IC	Quantity

Discussion:

In this experiment, we were given the task to implement an ALU that can perform four arithmetic operations and two logical operations. We implemented the ALU in such a way that it can perform both arithmetic and logic operations in a single circuit, instead of requiring two different circuits. The number of ICs were kept as minimal as possible. Logisim - Win-2.7.1 simulation software was used to simulate the circuit.