

# EE224 Course Project Report

**By Srisht Singh (170070056)**

**Atharv Pawar (170070003)**

**Rishabh Ramteke (170070046)**

**Vishwas Bharti (170070060)**

Finite State Machine :

## ALU :

```
port ( A, B: in std_logic_vector(15 downto 0);
      C: out std_logic_vector(15 downto 0);
      carry, iszero: out std_logic;
      S: in std_logic_vector(1 downto 0);
      Z: out std_logic);
signal S1: std_logic_vector(15 downto 0);
signal S0: std_logic_vector(15 downto 0);
signal sum: std_logic_vector(16 downto 0);
signal eq: std_logic_vector(15 downto 0);
signal temp: std_logic_vector(15 downto 0);
```

A & B are 16 bit inputs. C is 16 bit output. 'Carry' is carry of when the ALU performs addition.  
S is 2 bit input.

If S=00, then ALU should perform addition

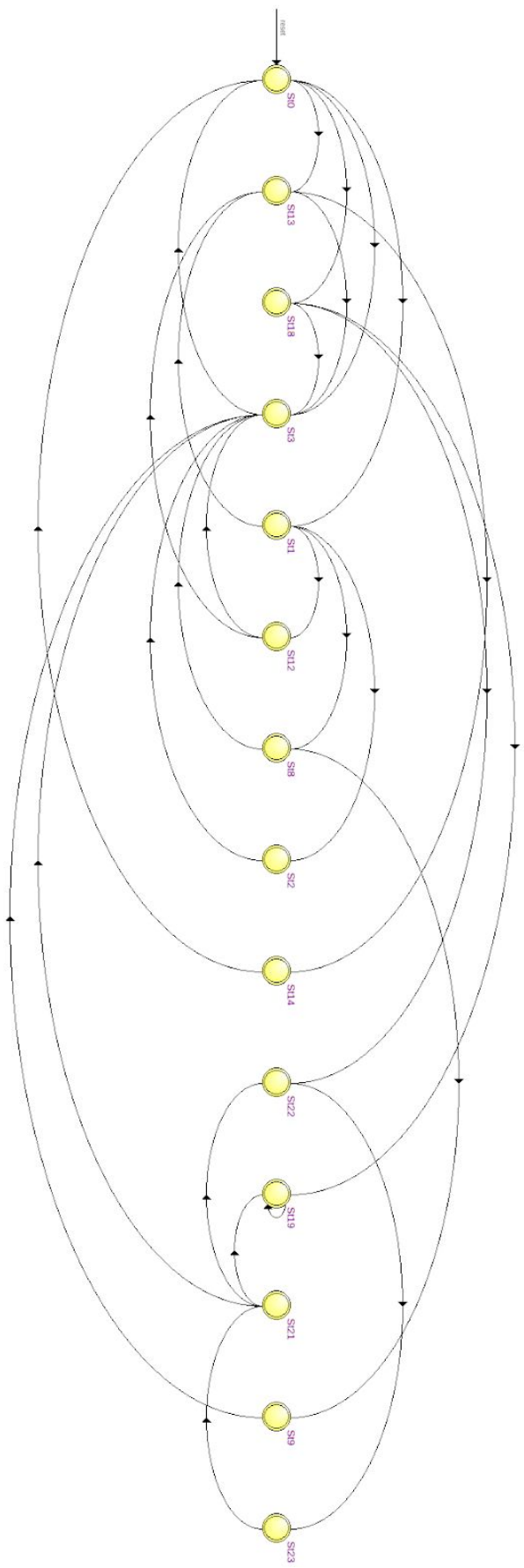
If S=01, then ALU should perform subtraction

If S=10, then ALU should perform as a NAND gate.

'iszero' is a signal indicates when the result C is equal to zero.

eq is output of (A xnor B). We have used this signal to check equality of A and B.

'Z' is 1 when A and B are equal.



## register\_file :

```
port ( A1, A2: in std_logic_vector(2 downto 0);  
       D1, D2: out std_logic_vector(15 downto 0);  
       A3: in std_logic_vector(2 downto 0);  
       D3: in std_logic_vector(15 downto 0);  
       en, clk: in std_logic);  
signal reg_file: rfarray := (others => "0000000000000000");
```

A1,A2 are 3 bit inputs and D1,D2 are 16 bit outputs.

A1, A2 are address and when we provide them as input, we get the data D1,D2 respectively, that these addresses stored, as output.

Thus, in this case we are using register\_file to extract the data that is stored at a particular address.

A3 is 3 bit input and D3 is 16 bit input. In this case, we have to store D3 data at an address A3.