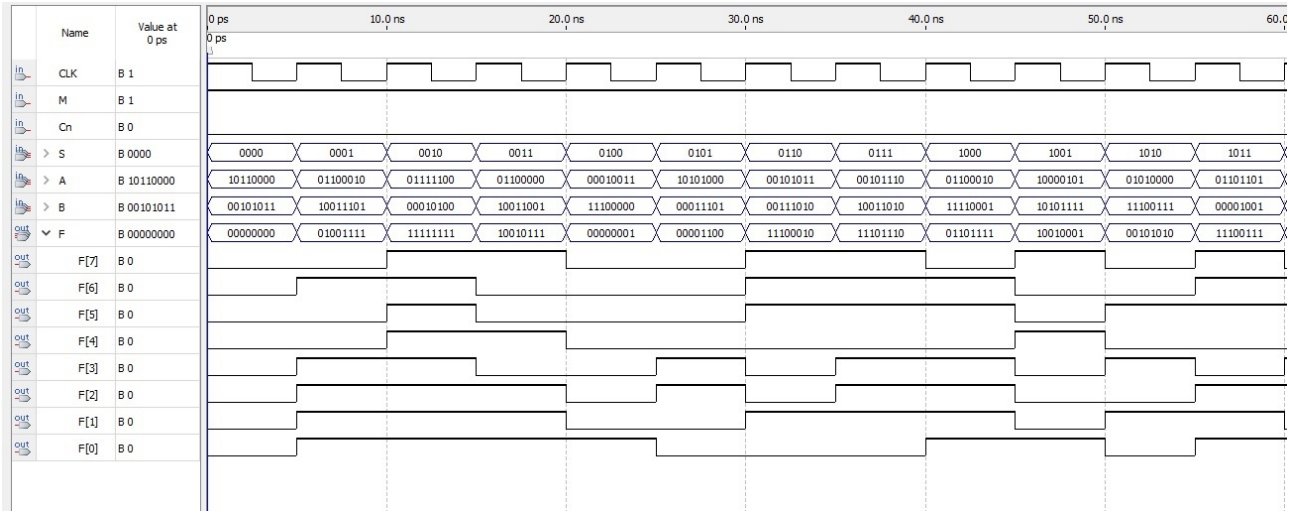
Code for 8-bit ALU:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity alu is
	port(
		A, B : in std_logic_vector(7 downto 0);
		CLK, M, Cn : in std_logic;
		S : in std_logic_vector(3 downto 0);
		F : out std_logic_vector(7 downto 0)
	);
end alu;

architecture behavior of alu is
begin
	process(A, B, CLK, M, Cn, S)
	begin
		if(CLK'event and CLK = '1') then
			if(M='1') then			-- Logic Functions	-- Cn is ignored.
				case S is
					when "0000" => F <= not (A);
					when "0001" => F <= A nand B;
					when "0010" => F <= (not A) or B;
					when "0011" => F <= "00000001";
					when "0100" => F <= A nor B;
					when "0101" => F <= not B;
					when "0110" => F <= A xnor B;
					when "0111" => F <= A or (not(B));
					when "1000" => F <= not (A) and B;
					when "1001" => F <= A xor B;
					when "1010" => F <= B;
					when "1011" => F <= A or B;
					when "1100" => F <= "00000000";
					when "1101" => F <= A and (not(B));
					when "1110" => F <= A and B;
					when "1111" => F <= A;
				end case;
```

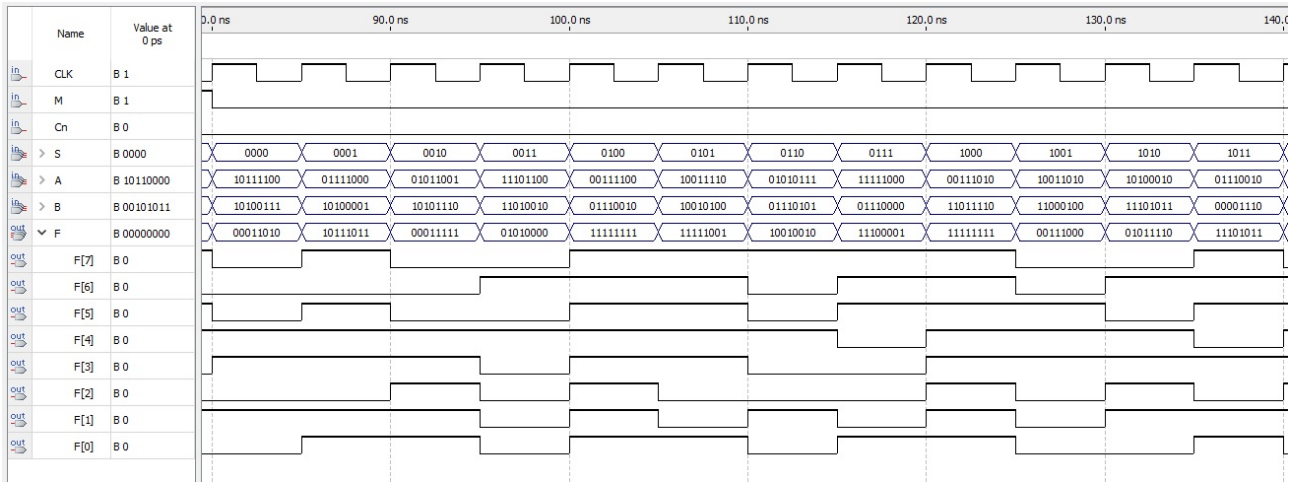
```
                else                    -- M = '0'      -- Arithmetic Functions.
                        if Cn = '0' then                -- SET 1
                                case S is
                                        when "0000" => F <= A - 1;
                                        when "0001" => F <= (A and B) - 1;
                                        when "0010" => F <= (A and not(B)) - 1;
                                        when "0011" => F <= -"00000001";
                                        when "0100" => F <= A + (A or not(B));
                                        when "0101" => F <= A and B + (A or not(B));
                                        when "0110" => F <= A - B - 1;
                                        when "0111" => F <= A or not(B);
                                        when "1000" => F <= A + (A or B);
                                        when "1001" => F <= A + B;
                                        when "1010" => F <= (A and not(B)) + (A or B);
                                        when "1011" => F <= A or B;
                                        when "1100" => F <= A;
                                        when "1101" => F <= (A and B) + A;
                                        when "1110" => F <= (A and not(B)) + A;
                                        when "1111" => F <= A;
                                end case;
                        else            -- Cn = '1'     -- SET 2
                                case S is
                                        when "0000" => F <= A;
                                        when "0001" => F <= A and B;
                                        when "0010" => F <= A and not(B);
                                        when "0011" => F <= "00000000";
                                        when "0100" => F <= A + (A or not(B)) + 1;
                                        when "0101" => F <= A and B + (A or not(B)) + 1;
                                        when "0110" => F <= (A or not(B)) + 1;
                                        when "0111" => F <= A - B;
                                        when "1000" => F <= A + (A or B) + 1;
                                        when "1001" => F <= A + B + 1;
                                        when "1010" => F <= (A and not(B)) + (A or B) + 1;
                                        when "1011" => F <= (A or B) + 1;
                                        when "1100" => F <= A + 1;
                                        when "1101" => F <= (A and B) + A + 1;
                                        when "1110" => F <= (A and not(B)) + A + 1;
                                        when "1111" => F <= A + 1;
                                end case;
                        end if;
                end if;
        end if;
    end process;
end behavior;
```

RTL Diagram for 8-bit ALU:

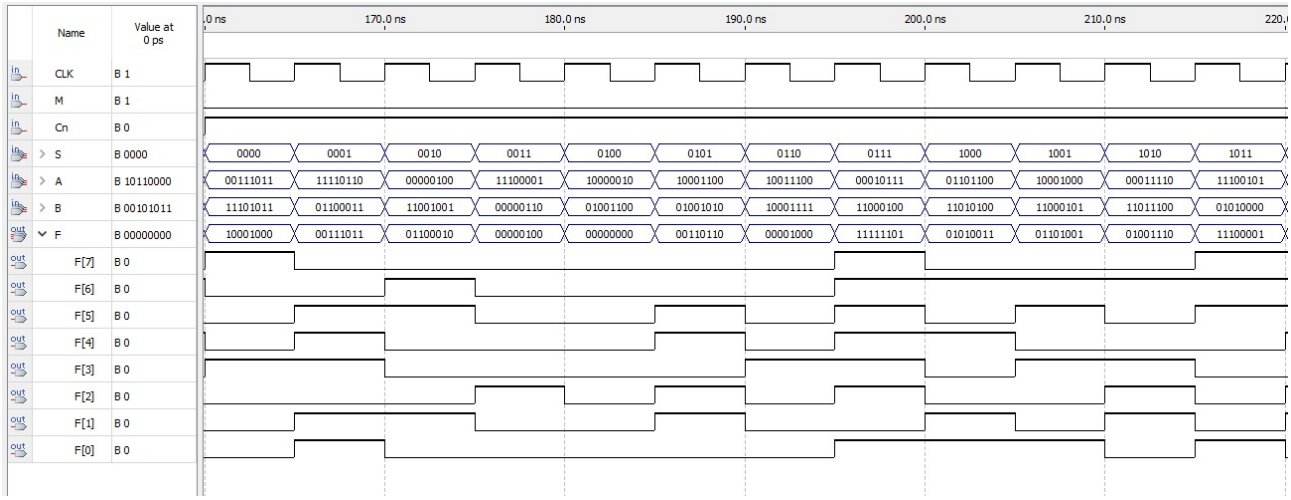
Waveform for 8-bit ALU with M=1:



Waveform for 8-bit ALU with M=0 and Cn=0:



Waveform for 8-bit ALU with M=0 and Cn=1:



Remarks:

The output was tallied with the truth table and was found to be correct. Hence the experiment is successful.