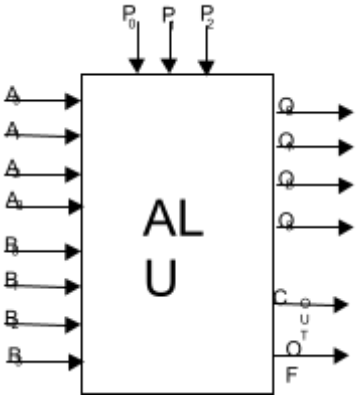
CME4456 Reconfigurable Computing – Homework 1

Due to: 22.03.2022

Cem Hafızoğulları - 2016510034

Write VHDL code to Design an arithmetic-logic circuit with 3-bit opcode variables $P_2P_1P_0$ and two 4-bits data inputs A and B by using full-adder blocks (FA). The circuit generates the following arithmetic, and logic operations. Draw the logic diagram with carry (C_{OUT}) and overflow (OF) outputs. Test your design with using Altera MuxPlus II or Quartus programme)

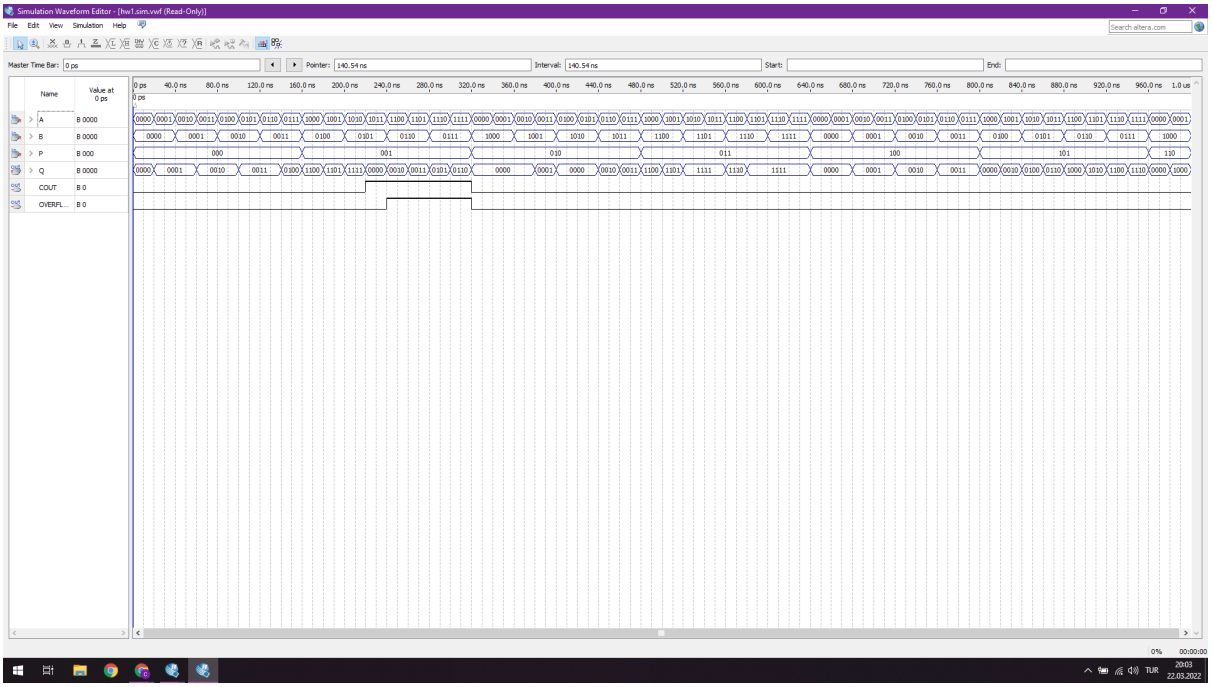


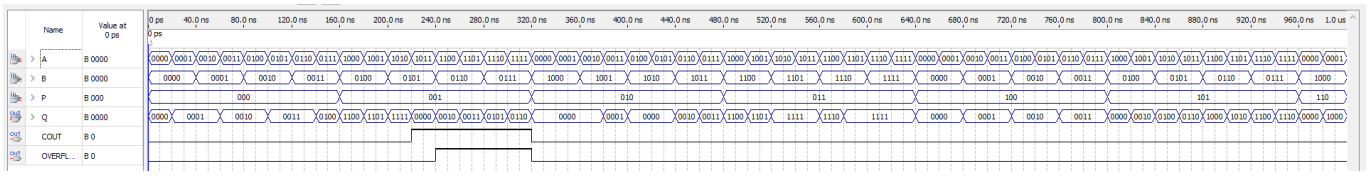
P_2	P_1	P_0	**OPCODE**	OPERATION
0	0	0	SUB	A - B
0	0	1	ADD	A + B
0	1	0	AND	A ∧ B
0	1	1	OR	A ∨ B
1	0	0	ASR	Arithmetic Shift Right A
1	0	1	ASL	Arithmetic Shift Left A
1	1	0	CSR	Circular Shift Right A
1	1	1	CSL	Circular Shift Left A

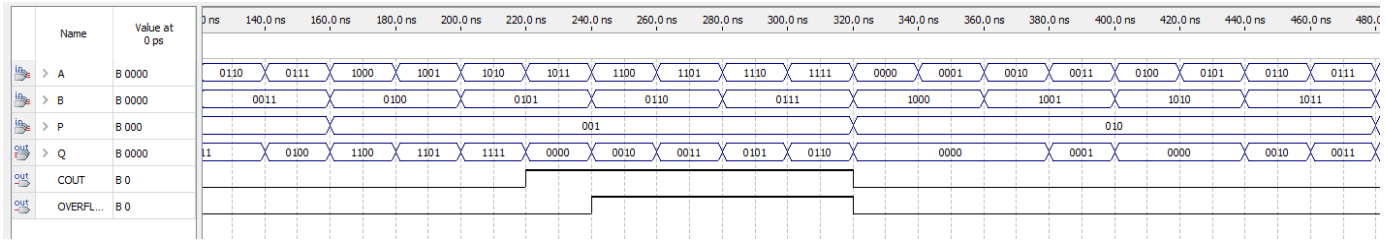
```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use ieee.numeric_std.all;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity ALU is
	port(
		A, B: in STD_LOGIC_VECTOR(3 downto 0);
		Q: out STD_LOGIC_VECTOR(3 downto 0);
		P: in STD_LOGIC_VECTOR(2 downto 0);
		COUT, OVERFLOW: out STD_LOGIC
	);
```

```
end ALU;

architecture ALU_a of ALU is
signal temp: STD_LOGIC_VECTOR(4 downto 0);
        begin
PROCESS(A, B, P)
        BEGIN
        COUT <= '0';
        OVERFLOW <= '0';
                CASE P IS
                        WHEN "000" =>
                                if (B > A) then
                                        OVERFLOW <= '1';
                                else
                                        OVERFLOW <= '0';
                                end if;
                                Q <= A - B;
                        WHEN "001" =>
                                temp <= std_logic_vector((unsigned("0" & A) + unsigned(B)));
                                Q <= temp(3 downto 0);
                                COUT <= temp(4);
                                if (temp > 16) then
                                        OVERFLOW <= '1';
                                end if;
                        WHEN "010" =>
                                Q <= A and B;
                        WHEN "011" =>
                                Q <= A or B;
                        WHEN "100" =>
                                Q <= std_logic_vector(shift_right(unsigned(A), 1));
                        WHEN "101" =>
                                Q <= std_logic_vector(shift_left(unsigned(A), 1));
                        WHEN "110" =>
                                Q <= std_logic_vector(rotate_right(unsigned(A),1));
                        WHEN "111" =>
                                Q <= std_logic_vector(rotate_left(unsigned(A),1));
                END CASE;
END PROCESS;
end ALU_a;
```

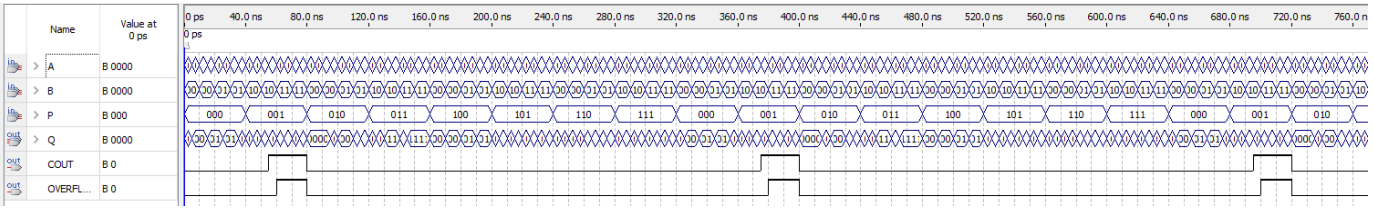
WAVEFORM:



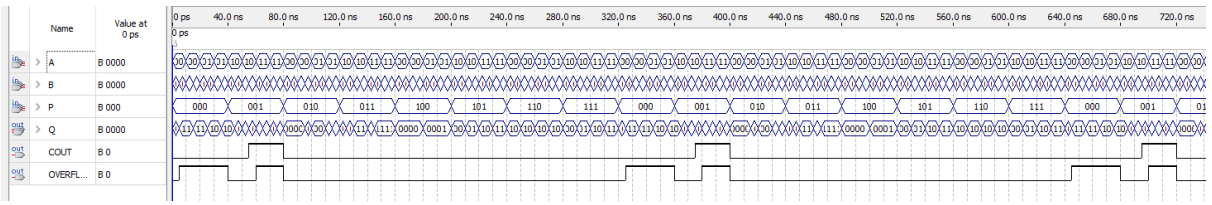




Waveform with increment by 1;
A : 5 ns
B : 10 ns
P : 40 ns



Waveform with increment by 1;
A : 10 ns
B : 5 ns
P : 40 ns


Name
Value at 0 ps
0 ps
40.0 ns
80.0 ns
120.0 ns
160.0 ns
200.0 ns
240.0 ns
280.0 ns
320.0 ns
360.0 ns
400.0 ns
440.0 ns
480.0 ns
520.0 ns
560.0 ns
600.0 ns
640.0 ns
680.0 ns
720.0 ns
A
B 0000
B
B 0000
P
B 000
Q
B 0000
COUT
B 0
OVERFL...
B 0