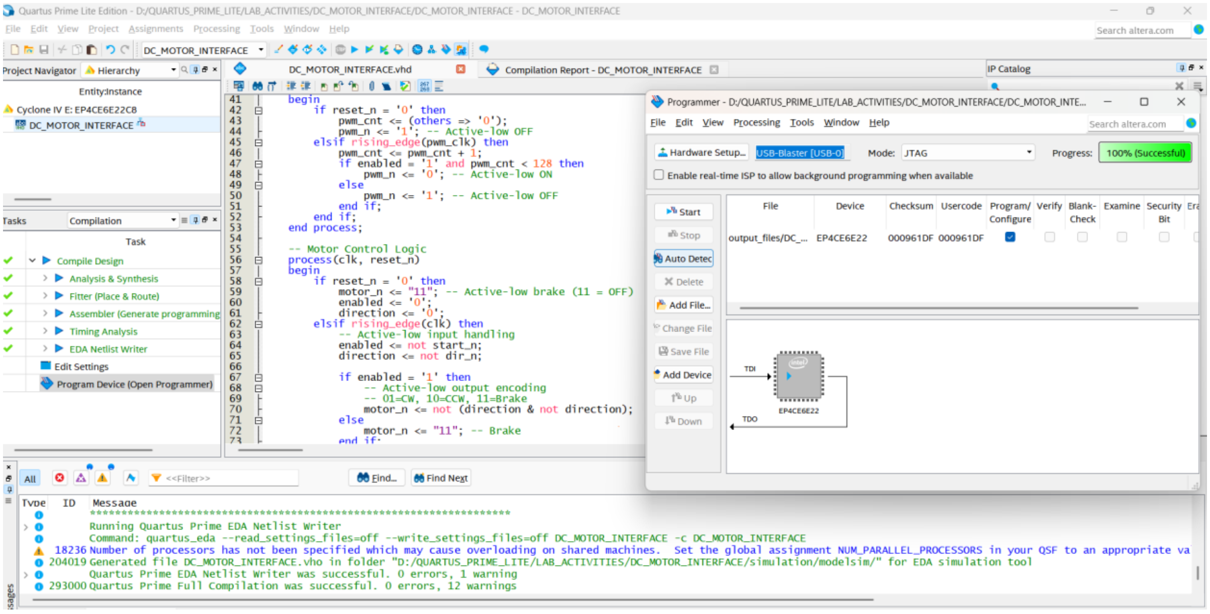
PLAZA, ELMO L.
BSCPE 3A

18. DC MOTOR INTERFACE

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity DC_MOTOR_INTERFACE    is
Port (
   -- Active-low control inputs
   clk      : in  STD_LOGIC;   -- PIN_23 (50MHz)
   reset_n  : in  STD_LOGIC;   -- PIN_25 (RESET button)
   start_n  : in  STD_LOGIC;   -- PIN_88 (KEY1)
   dir_n    : in  STD_LOGIC;   -- PIN_89 (KEY2)

   -- Active-low outputs
   pwm_n    : out STD_LOGIC;   -- PIN_84 (led4)
   motor_n  : out STD_LOGIC_VECTOR(1 downto 0); -- PIN_85,86 (led3,led2)

   -- Active-low status LED
   stat_led_n : out STD_LOGIC  -- PIN_87 (led1)
);
end DC_MOTOR_INTERFACE;

architecture Behavioral of DC_MOTOR_INTERFACE is
   signal pwm_cnt    : unsigned(7 downto 0) := (others => '0');
   signal clk_div    : unsigned(18 downto 0) := (others => '0');
   signal pwm_clk    : STD_LOGIC := '0';
   signal enabled    : STD_LOGIC := '0';
   signal direction  : STD_LOGIC := '0';
begin

   -- Clock divider (50MHz → 95Hz)
   process(clk)
   begin
      if rising_edge(clk) then
         clk_div <= clk_div + 1;
         pwm_clk <= clk_div(18);
      end if;
   end process;

   -- PWM Generator (active-low output)
   process(pwm_clk, reset_n)
   begin
      if reset_n = '0' then
         pwm_cnt <= (others => '0');
```

```
            pwm_n <= '1'; -- Active-low OFF
        elsif rising_edge(pwm_clk) then
            pwm_cnt <= pwm_cnt + 1;
            if enabled = '1' and pwm_cnt < 128 then
                pwm_n <= '0'; -- Active-low ON
            else
                pwm_n <= '1'; -- Active-low OFF
            end if;
        end if;
    end process;

    -- Motor Control Logic
    process(clk, reset_n)
    begin
        if reset_n = '0' then
            motor_n <= "11"; -- Active-low brake (11 = OFF)
            enabled <= '0';
            direction <= '0';
        elsif rising_edge(clk) then
            -- Active-low input handling
            enabled <= not start_n;
            direction <= not dir_n;

            if enabled = '1' then
                -- Active-low output encoding
                -- 01=CW, 10=CCW, 11=Brake
                motor_n <= not (direction & not direction);
            else
                motor_n <= "11"; -- Brake
            end if;
        end if;
    end process;

    -- Active-low status LED (ON when enabled)
    stat_led_n <= not enabled;

end Behavioral;
```



Quartus Prime Lite Edition - D:/QUARTUS_PRIME_LITE/LAB_ACTIVITIES/DC_MOTOR_INTERFACE/DC_MOTOR_INTERFACE - DC_MOTOR_INTERFACE
File Edit View Project Assignments Processing Tools Window Help
Search altera.com
DC_MOTOR_INTERFACE
Project Navigator
Hierarchy
Entity:Instance
Cyclone IV E: EP4CE6E22C8
DC_MOTOR_INTERFACE
DC_MOTOR_INTERFACE.vhd
Compilation Report - DC_MOTOR_INTERFACE
IP Catalog
Tasks
Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming
Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
All
<<Filter>>
Find...
Find Next
Programmer - D:/QUARTUS_PRIME_LITE/LAB_ACTIVITIES/DC_MOTOR_INTERFACE/DC_MOTOR_INTE...
File Edit View Processing Tools Window Help
Search altera.com
Hardware Setup...
Start
Stop
Auto Detec
Delete
Add File...
Change File
Save File
Add Device
Up
Down
TDI
TDO
EP4CE6E22

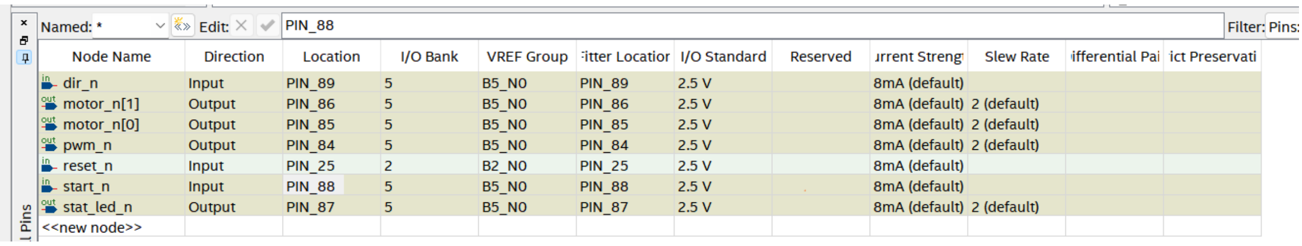
```
        begin
            if reset_n = '0' then
                pwm_cnt <= (others => '0');
                pwm_n <= '1'; -- Active-low OFF
            elsif rising_edge(pwm_clk) then
                pwm_cnt <= pwm_cnt + 1;
                if enabled = '1' and pwm_cnt < 128 then
                    pwm_n <= '0'; -- Active-low ON
                else
                    pwm_n <= '1'; -- Active-low OFF
                end if;
            end if;
        end process;

        -- Motor Control Logic
        process(clk, reset_n)
        begin
            if reset_n = '0' then
                motor_n <= "11"; -- Active-low brake (11 = OFF)
                enabled <= '0';
                direction <= '0';
            elsif rising_edge(clk) then
                -- Active-low input handling
                enabled <= not start_n;
                direction <= not dir_n;

                if enabled = '1' then
                    -- Active-low output encoding
                    -- 01=CW, 10=CCW, 11=Brake
                    motor_n <= not (direction & not direction);
                else
                    motor_n <= "11"; -- Brake
                end if;
```

USB-Blaster [USB-0]   Mode: JTAG   Progress: 100% (Successful)

□ Enable real-time ISP to allow background programming when available

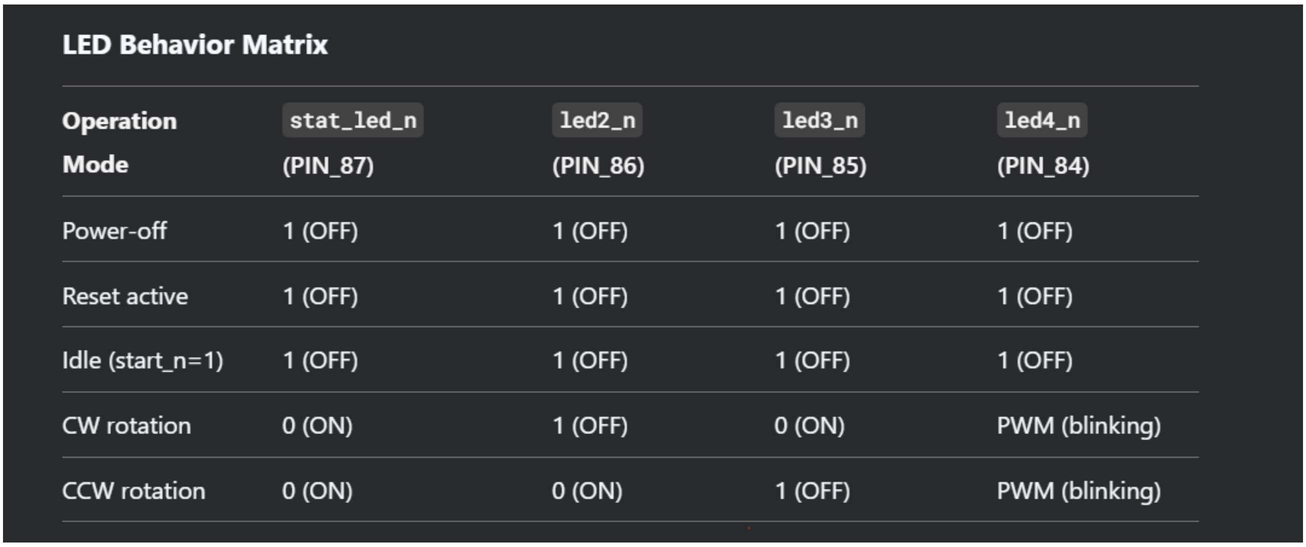
File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Era
output_files/DC_...	EP4CE6E22	000961DF	000961DF	☑	□	□	□	□	□

```
Type  ID     Message
             *********************************************************************
             Running Quartus Prime EDA Netlist Writer
             Command: quartus_eda --read_settings_files=off --write_settings_files=off DC_MOTOR_INTERFACE -c DC_MOTOR_INTERFACE
      18236  Number of processors has not been specified which may cause overloading on shared machines.  Set the global assignment NUM_PARALLEL_PROCESSORS in your QSF to an appropriate va
      204019 Generated file DC_MOTOR_INTERFACE.vho in folder "D:/QUARTUS_PRIME_LITE/LAB_ACTIVITIES/DC_MOTOR_INTERFACE/simulation/modelsim/" for EDA simulation tool
             Quartus Prime EDA Netlist Writer was successful. 0 errors, 1 warning
      293000 Quartus Prime Full Compilation was successful. 0 errors, 12 warnings
```



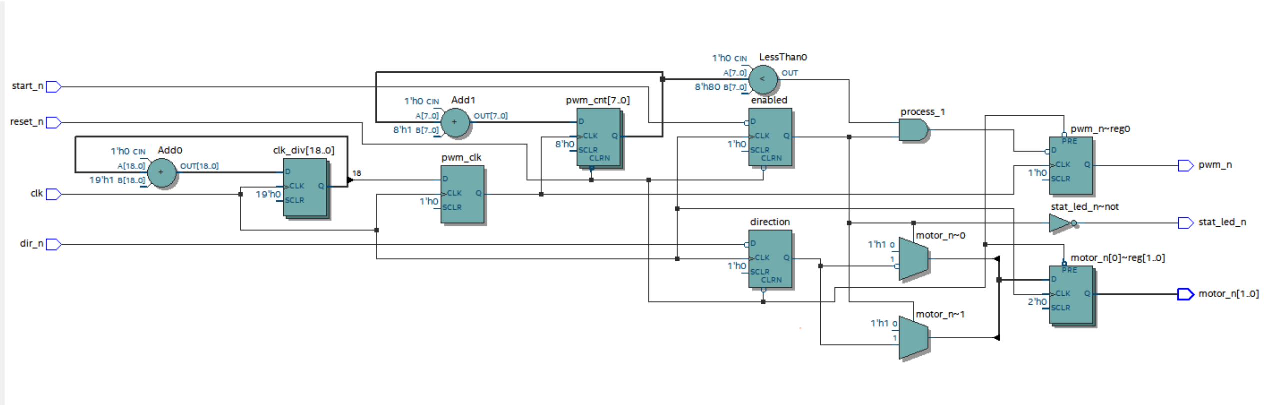
Named: * Edit: PIN_88 Filter: Pins:

Node Name	Direction	Location	I/O Bank	VREF Group	itter Locatior	I/O Standard	Reserved	urrent Streng	Slew Rate	ifferential Pai	ict Preservati
dir_n	Input	PIN_89	5	B5_N0	PIN_89	2.5 V		8mA (default)			
motor_n[1]	Output	PIN_86	5	B5_N0	PIN_86	2.5 V		8mA (default)	2 (default)		
motor_n[0]	Output	PIN_85	5	B5_N0	PIN_85	2.5 V		8mA (default)	2 (default)		
pwm_n	Output	PIN_84	5	B5_N0	PIN_84	2.5 V		8mA (default)	2 (default)		
reset_n	Input	PIN_25	2	B2_N0	PIN_25	2.5 V		8mA (default)			
start_n	Input	PIN_88	5	B5_N0	PIN_88	2.5 V		8mA (default)			
stat_led_n	Output	PIN_87	5	B5_N0	PIN_87	2.5 V		8mA (default)	2 (default)		
<<new node>>											



LED Behavior Matrix

Operation Mode	stat_led_n (PIN_87)	led2_n (PIN_86)	led3_n (PIN_85)	led4_n (PIN_84)
Power-off	1 (OFF)	1 (OFF)	1 (OFF)	1 (OFF)
Reset active	1 (OFF)	1 (OFF)	1 (OFF)	1 (OFF)
Idle (start_n=1)	1 (OFF)	1 (OFF)	1 (OFF)	1 (OFF)
CW rotation	0 (ON)	1 (OFF)	0 (ON)	PWM (blinking)
CCW rotation	0 (ON)	0 (ON)	1 (OFF)	PWM (blinking)



start_n
reset_n
clk
dir_n
1'h0 CIN
Add0
A[18..0]
19'h1 B[18..0]
OUT[18..0]
clk_div[18..0]
19'h0
18
1'h0 CIN
Add1
A[7..0]
8'h1 B[7..0]
OUT[7..0]
pwm_clk
1'h0
pwm_cnt[7..0]
8'h0
1'h0 CIN
LessThan0
A[7..0]
8'h80 B[7..0]
OUT
enabled
1'h0
direction
1'h0
process_1
pwm_n~reg0
1'h0
pwm_n
stat_led_n~not
stat_led_n
motor_n~0
1'h1
motor_n~1
1'h1
motor_n[0]~reg[1..0]
2'h0
motor_n[1..0]
D
CLK
Q
SCLR
CLRN
PRE