JURNAL UPI

ASEAN Journal of Science and Engineering Education

Journal homepage: http://ejournal.upi.edu/index.php/AJSEE/

Design of VHDL for MPPT Incremental Conductance on FPGA with Altera DE2-115 Development Board for Educational Purposes

Virbora Ny[1,], Channareth Srun[1], Phok Chrin[2], Sokchea Am[2], Bunthern Kim[2], Saran Meas[2]*

[1]Faculty of Electronics, National Polytechnic Institute of Cambodia, Cambodia
[2]Energy Technology and Management, Institute of Technology of Cambodia, Cambodia

*Correspondence: E-mail: nareth16npic@gmail.com

ABSTRACT

In this paper, the design of VHDL for MPPT Incremental Conductance on an FPGA with the Altera DE2-115 Development Board is presented for educational purposes. The FPGA is programmed to control the MPPT tracking process by obtaining the maximum power from the solar panel. The Altera DE2-115 Development Board is used for the FPGA system design. The system is designed for individual blocks that are designed specifically to operate each step in the MPPT process. The system consists of five main parts: the solar panel, ADC, FPGA, PWM, and DC-DC Boost Converter. The solar panel provides the DC voltage and current. The ADC converts the solar panel's constantly changing voltage and current to a digital signal. The INC algorithm is started after the FPGA receives the digital signal from the ADC. The output PWM from the MPPT algorithm will drive the DC-DC Boost Converter circuit for the solar power tracking process. The experimental results show that the designed system can track the maximum power point of the solar panel.

ARTICLE INFO

Article History:
Submitted/Received 28 Jan 2023
First Revised 02 Mar 2023
Accepted 24 May 2023
First Available online 26 May 2023
Publication Date 01 Mar 2024

Keyword:
Altera cyclone IV,
Altera DE2-115,
FPGA,
Incremental conductance,
MPPT.

1. INTRODUCTION

Due to the increasing power requirement of a fast-growing population and the power required, the reliance on fossil fuel and natural gas usage badly impacts the environment. Solar energy is a part of clean energy because it harvests light, turns it into electricity, and can transfer power to the grid (Srun *et al.,* 2022). Maximum Power Point Tracking (MPPT) methods and algorithms are implemented to obtain the maximum power possible from the solar panel. Hence, the digital controller is required for monitoring and controlling the MPPT system (Srun *et al.,* 2022; Saran *et al.,*2022). A field programmable gate array (FPGA) is an advanced programmable digital interconnection system on a single silicon chip. FPGAs are designed to be flexible and can be formed into any digital logic designed based on Very-High-Speed Hardware Description Language (VHDL). In this research, the FPGA is programmed to control the MPPT racking process by obtaining the maximum power from the solar panel. The Terasic Altera Cyclone IV EP4CE115F29C7N is used as the FPGA system in this design.

2. METHODS

This study demonstrated the design of VHDL for MPPT incremental conductance on FPGA with Altera DE2-115 development board for educational purposes. We showed step-by-step in designing this process.

3. RESULTS AND DISCUSSION

3.1. System Overview

Figure 1 consists of five main parts of the FPGA MPPT system; the greyed-out block of the MPPT Tracking System is the part described in this design. In the diagram, the solar panel is the DC voltage and current source. The output voltage of a DC-DC boost converter can be regulated by using a voltage controller (Samman *et al*., 2019). Since the Altera Cyclone IV is a complete digital system, the continuously changing voltage and current from the solar panel must go through the Analog to Digital Converter (ADC) process. The digital signal from the ADC was passed to the FPGA, starting the INC algorithm process. The output pulse width modulation (PWM) from the MPPT algorithm will drive the DC-DC boost converter circuit for the solar power tracking process (Br Ginting *et al.,* 2021).

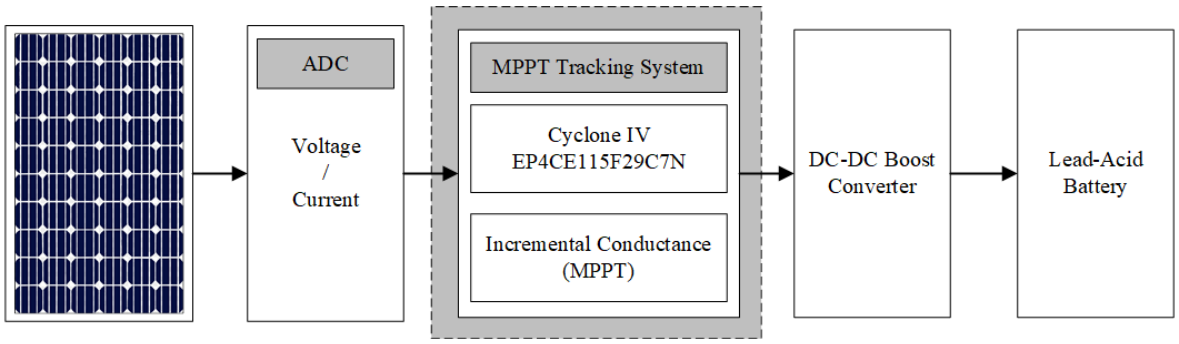

Figure 1. FPGA-based MPPT block system.

3.1.1. Incremental conductance algorithm

Incremental Conductance (INC) is an MPPT algorithm that is based on the voltage and current of the solar panel. The working principle of this algorithm is to obtain solar power at the maximum power point (MPP), as shown in **Figure 2**. This algorithm depends on two parameters, I and V. If dI/dV > I/V, the algorithm will increase the duty cycle to obtain more

DOI: http://dx.doi.org/10.17509/xxxx.xxxx
p- ISSN 2775-6793 e- ISSN 2775-6815

power from the solar panel. During the equivalent of dI/dV = -I/V, the algorithm will maintain the same duty ratio equivalent to the maximum power point reached. Otherwise, if the dI/dV< I/V, the algorithm will reduce the duty ratio to prevent a possible short circuit (Sahu & Dey, 2022; Kabalci *et al.,* 2017).

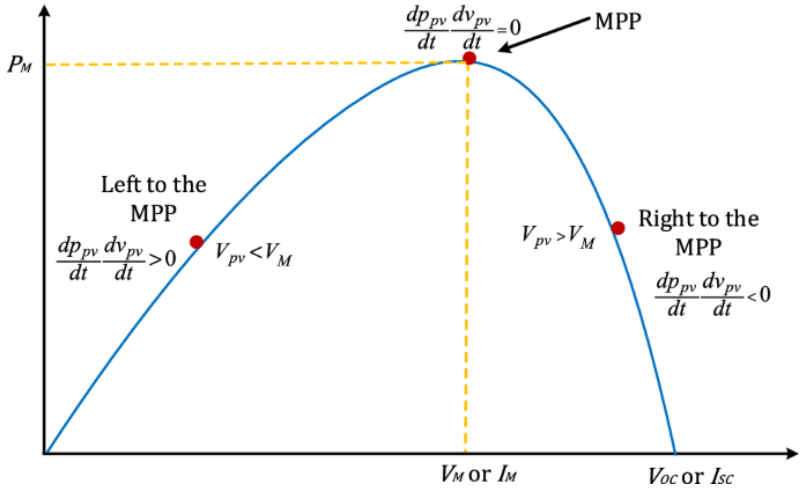

Figure 2. Solar panel maximum power point graph.

3.1.2. DC-DC boost converter

The boost converter is a DC-DC converter topology that converts the lower input voltage to a higher output voltage. **Figure 3** shows the circuit diagram of a boost converter circuit containing important components such as an inductor, MOSFET, diode, and capacitor. To generate a higher output voltage first, the energy is stored in the inductor while the switch is closed. The stored energy will later be released to the load, increasing the output voltage (Al Husaeni & Hadianto, 2022).

As shown in **Figure 3,** when the duty cycle is 100%, the L_i and the MOSFET are connected directly, shorting the positive to the negative terminal. In this condition, a huge current passed through the inductor and MOSFET as a result of the damaged MOSFET and burned circuit. As mentioned, the design of MPPT must set a specific maximum duty cycle to prevent this condition from happening (Riza, 2021).

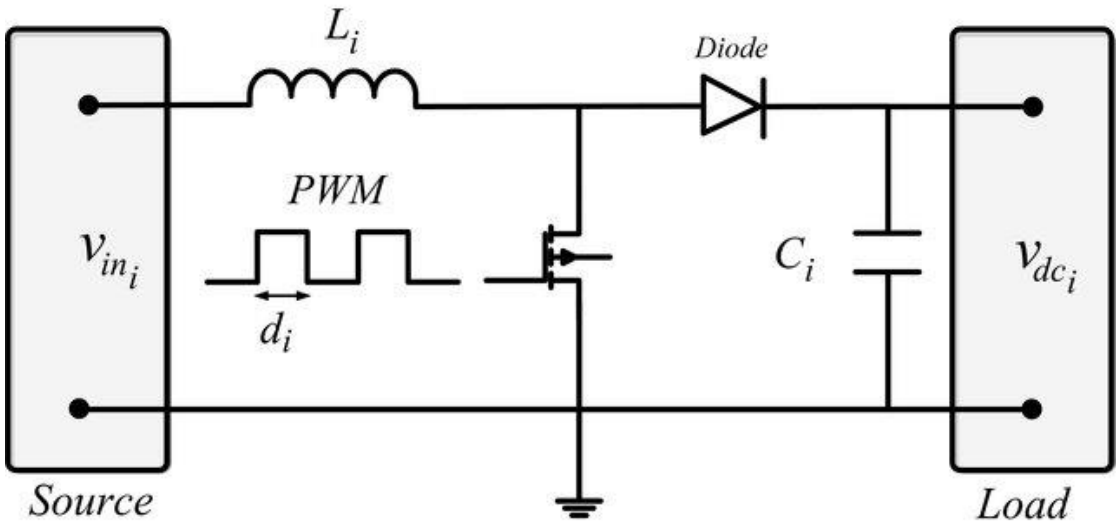

Figure 3. The boost converter schematic.

DOI: http://dx.doi.org/10.17509/xxxx.xxxx
p- ISSN 2775-6793 e- ISSN 2775-6815

3.2. System Design, Simulation, and Implementation

3.2.1. Design VHDL block diagram files

To simplify the design of the MPPT tracking system on the FPGA, there are five blocks designed separately as a Block Diagram File (BDF), including signal delay, delta calculation, division calculation, incremental conductance MPPT, and PWM generation block. For design purposes, each block is connected directly to the same clock source. The delay of each pipeline design is embedded in each block of the process. Each block accepts an 8-bit signal and is operated as a signed value ranging from -128 to 127 and an unsigned value ranging from 0 to 255 in decimal value. The process is done every 3 clock cycles.

3.2.1.1. Signal delay block

Figure 4 shows the signal delay block accepts 8-bit data input and outputs two 8-bit data: the O [7..0] is real-time output, and the O_DLY [7..0] is the output data after 3 clock cycles of delay. The output signal of the Signal Delay Block simulation is a Vector Waveform File (.vwf), as shown in **Figure 5**. The O_DLY [7..0] signal is passed after three falling edges of the clock cycle following the O[7..0] signal.

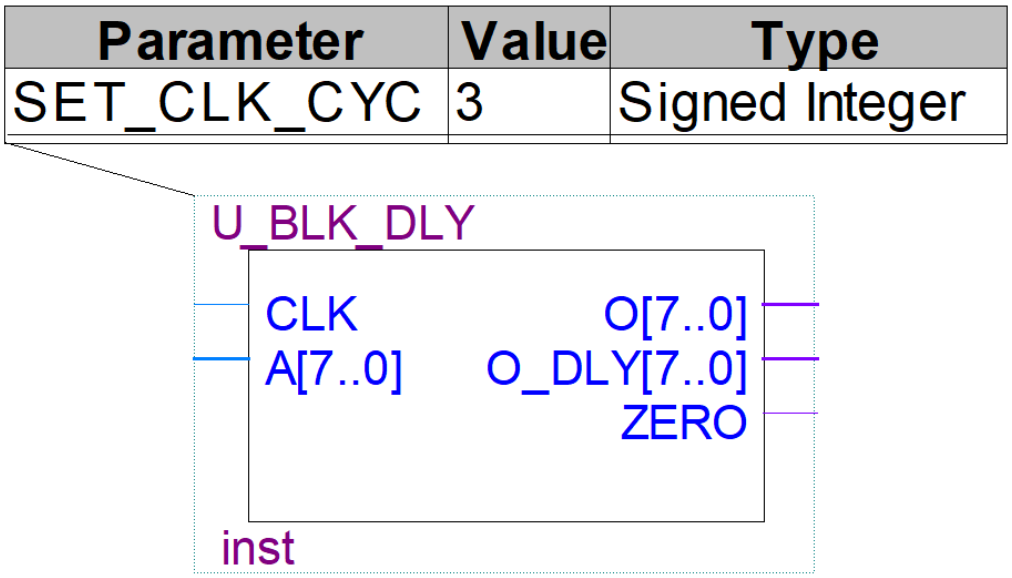

Figure 4. Signal delay block.

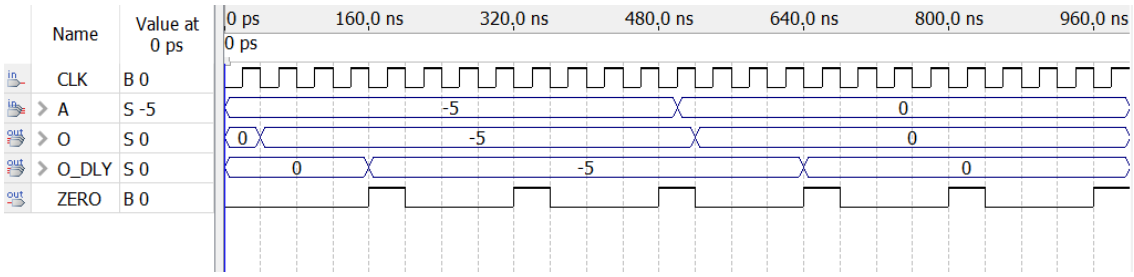

Figure 5. Vector waveform of signal delay block.

3.2.1.2. Delta calculation block

For calculating the ΔSIG input, the data is processed through the Delta Calculation Block shown in **Figure 6**, where two 8-bit inputs are subtracted for every 3 clock cycles. The input

DOI: http://dx.doi.org/10.17509/xxxx.xxxx
p- ISSN 2775-6793 e- ISSN 2775-6815

data A and B signals passed from the Signal Delay Block. The Delta Calculation Block processes the calculation for every 3 falling edge clock cycles and outputs the result shown in **Figure 7**.

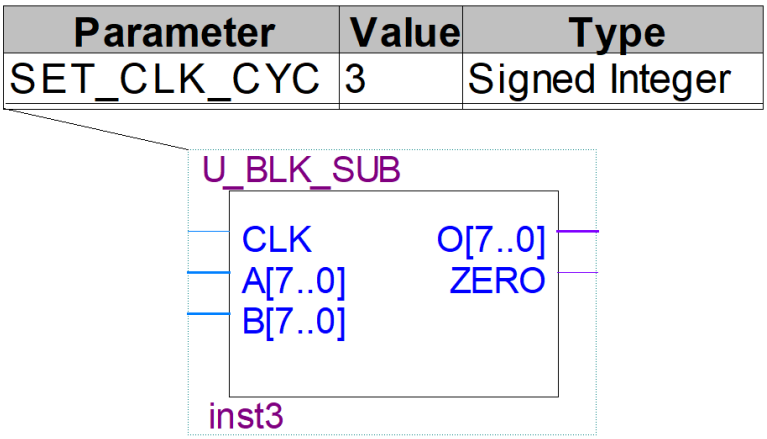

Figure 6. Delta calculation block.

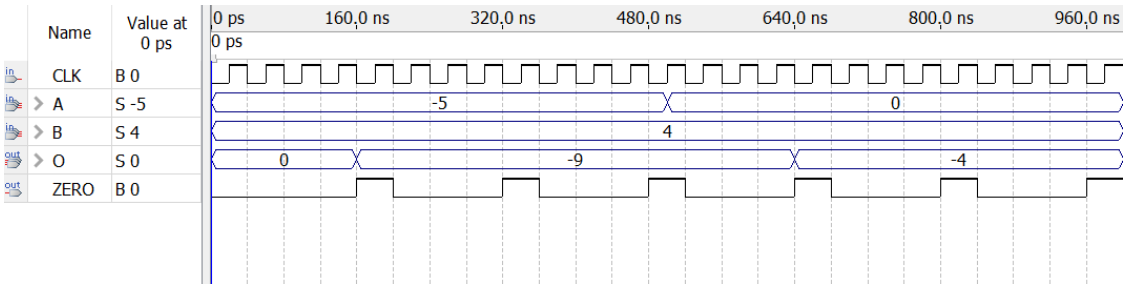

Figure 7. Vector waveform of delta calculation block.

3.2.1.3. Division calculation block

The Division Calculation Block shown in **Figure 8** is a BDF generated using Altera Mega functions. The BDF accepts two 8-bit inputs and is calculated for the quotient and the remainder of the division. As the system is designed as a pipeline, the calculation proceeds every 3 clock cycles. As shown in **Figure 9**, input A is the numerator, and B is the denominator. For every 3 falling edges of the clock cycles, the block will output a quotient and the remainder as 8-bit SIGNED.

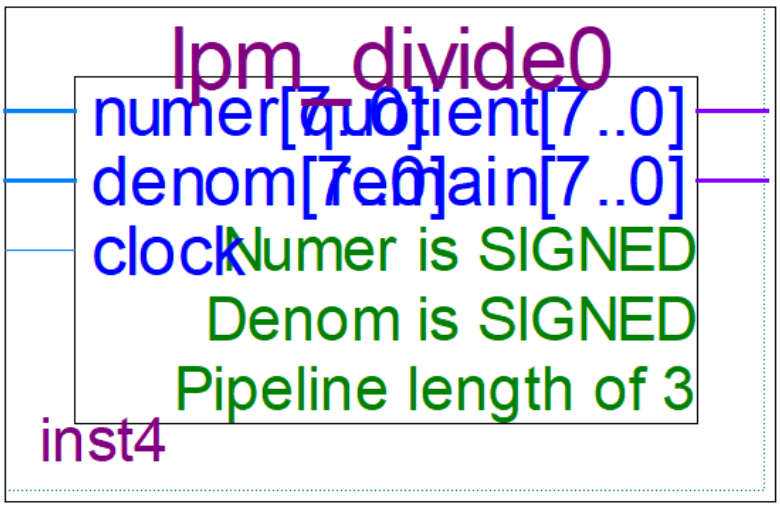

Figure 8. Division calculation block.

DOI: http://dx.doi.org/10.17509/xxxx.xxxx
p- ISSN 2775-6793 e- ISSN 2775-6815

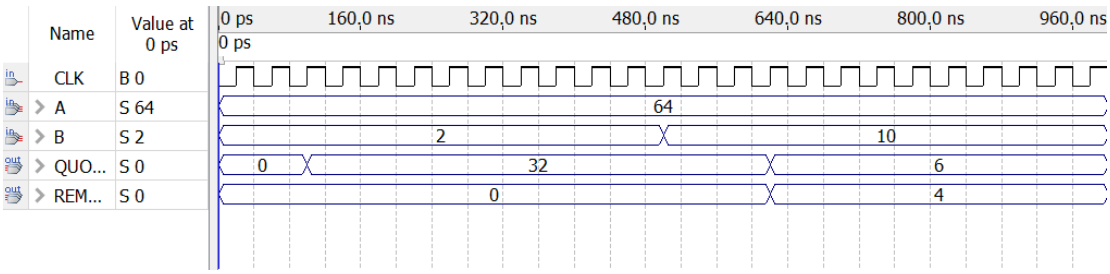

Figure 9. Vector waveform of delta calculation block.

3.2.1.4. Incremental conductance MPPT block

Figure 10 is where the MPPT algorithm is processed. The BDF of this design accepts four data inputs and generates an 8-bit duty cycle value. To prevent short circuit issues during the MPPT, the maximum PWM duty cycle is set to 150_{10} or 10010101_2 approximately 58.83% ON time, as shown in **Figure 14**.

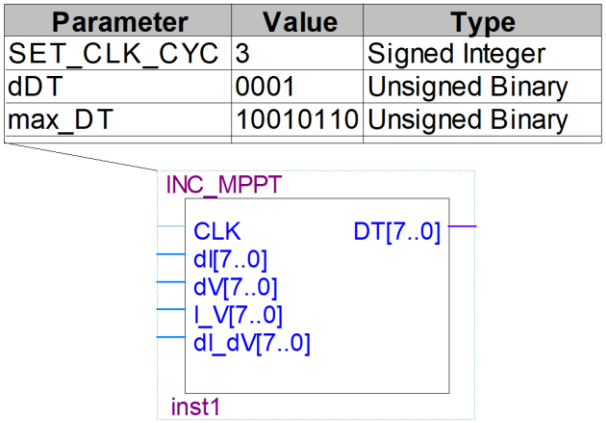

Figure 10. Incremental conductance MPPT block.

3.2.1.5. PWM generation block

To generate the PWM signal, the PWM Generation Block accepts an 8-bit duty cycle value from the INC MPPT block and creates a variable PWM signal based on the input value, as shown in **Figure 11**. The PWM generation block accepts an input value of UNSIGNED (8-bit) ranging from 0-255, translating to a duty cycle of 0–100%. The vector waveform in **Figure 12** shows the PWM generation from the input 8-bit value.

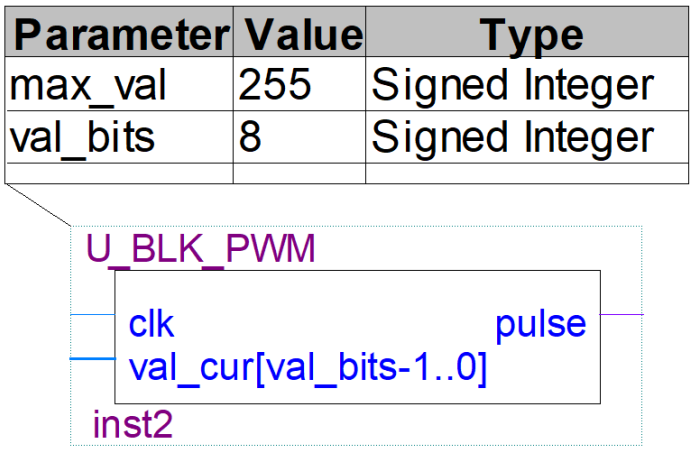

Figure 11. PWM generation block.

DOI: http://dx.doi.org/10.17509/xxxx.xxxx
p- ISSN 2775-6793 e- ISSN 2775-6815

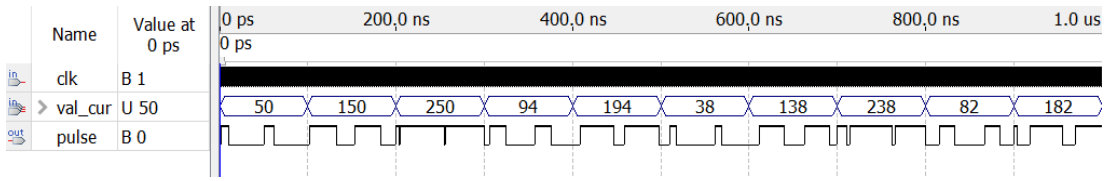

Figure 12. Vector waveform of PWM generator.

3.2.2. Assembled block of the MPPT VHDL

Figure 13 shows the complete connection between BDF blocks; as mentioned, all the blocks share the same clock source. For a completed BDF, there are three inputs and one output. The current and voltage sources are both 8-bit signals passed from the ADC0804 chip, and the output is a PWM signal.

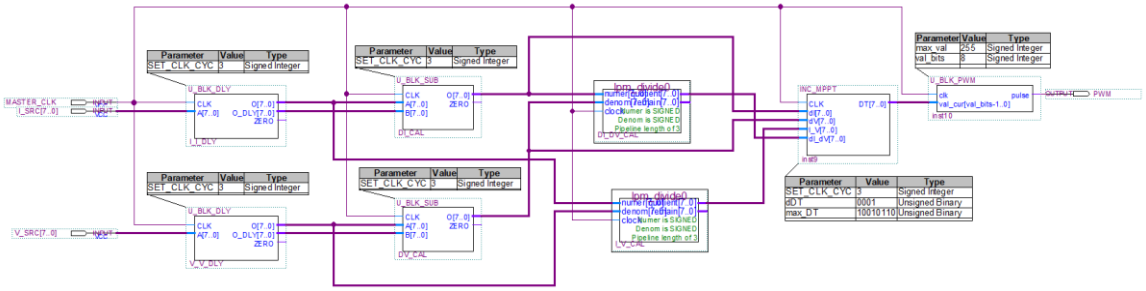

Figure 13. The completed BDF block of INC MPPT.

3.2.3. Assembled FPGA maximum power point tracking simulation and experimentation

Due to the limited performance of the simulation software with $1\mu S$ runtime, the *MASTER CLK* is set to $10pS$ (10 THz) to obtain enough clock cycles for the simulation.

3.2.3.1. Testing condition 1: Increasing current and voltage MPPT

In this testing, the simulated signals of the *I_SRC* and *V_SRC* are continuously increasing current from 1 to 3A. The voltage source remains the same, from 9 to 18V. In response, the PWM output signal went to its maximum value but stayed at the set value of 150_{10} or 10010110_2 (see **Figure 14**).

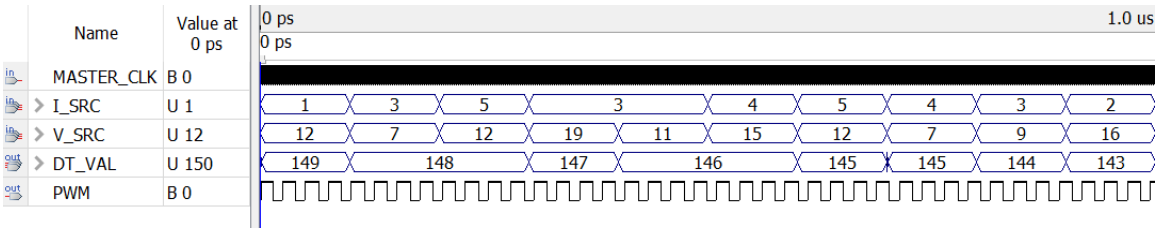

Figure 14. Simulation signal of MPPT increasing current and voltage MPPT.

3.2.3.2. Testing condition 2: Decreasing current and voltage MPPT

In this test, the simulation was set to decrease the current from 3 A to 1 A. The voltage source remains the same, from 9 V to 18 V. The PWM signal shown in **Figure 15** decreases according to the INC algorithm, and decreasing the PWM duty ratio helps prevent the solar panel from reaching *right to the MPP* as shown in **Figure 2**.

DOI: http://dx.doi.org/10.17509/xxxx.xxxx
p- ISSN 2775-6793 e- ISSN 2775-6815

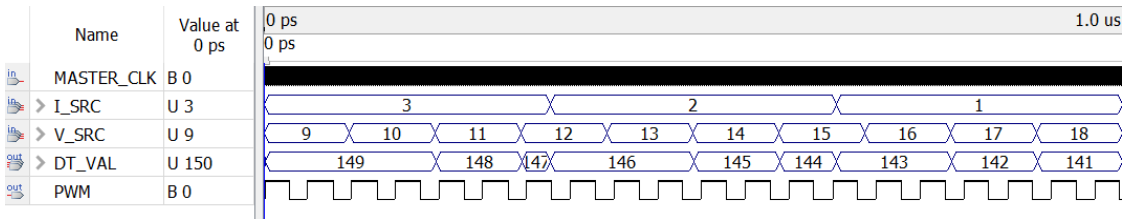

Figure 15. Simulation signal of MPPT decreasing current and voltage MPPT.

3.2.3.3. Testing condition 3: Variable current and constant voltage MPPT

Testing the algorithm with variable current and constant voltage is shown in **Figure 16**. **Figures 17, 18,** and **19** show the changing duty cycle of the PWM signal for the MPPT algorithm. The experiment was tested with a fixed voltage of 12 V and a variable current of 1 A to 7 A. By increasing the current as a consequence of increasing the power, the duty cycle increases to obtain the higher power from the solar panel, following **Figure 2**.

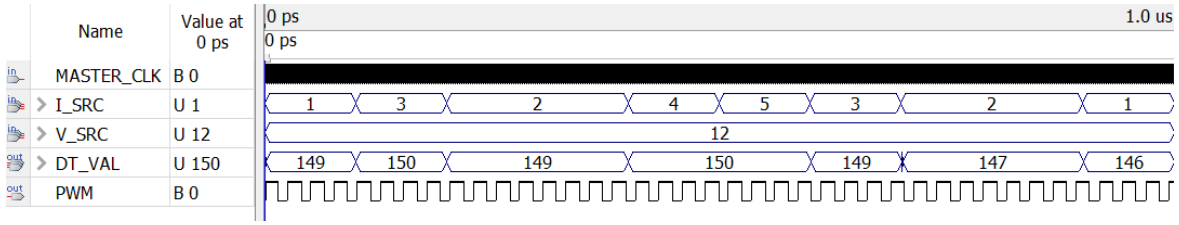

Figure 16. Simulation signal of MPPT variable current and constant voltage.

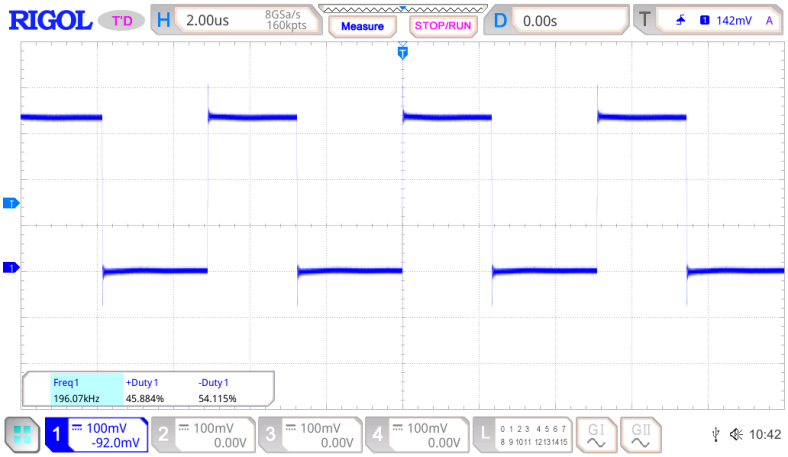

Figure 17. Output PWM signal, when fixed 12V and solar current 1A.

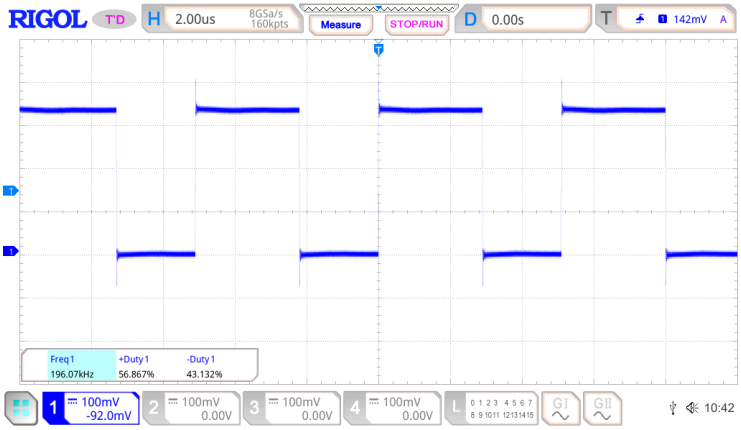

Figure 18. Output PWM signal, when 12V and solar current 3A.

DOI: http://dx.doi.org/10.17509/xxxx.xxxx
p- ISSN 2775-6793 e- ISSN 2775-6815

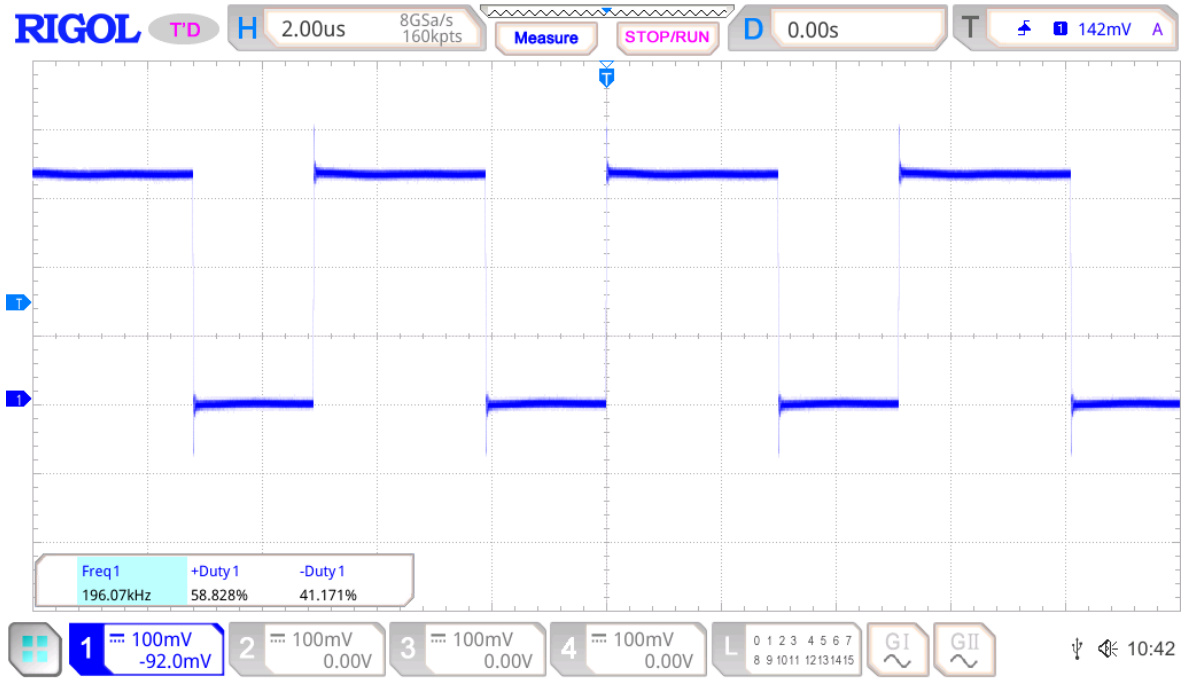

Figure 19. Output PWM signal, when fixed 12V and solar current 7A.

By decreasing the current from 7 A to 1 A as a drop in power, the MPPT algorithm reduces the duty cycle to maintain the maximum power point shown in **Figures 20, 21, and 22.**

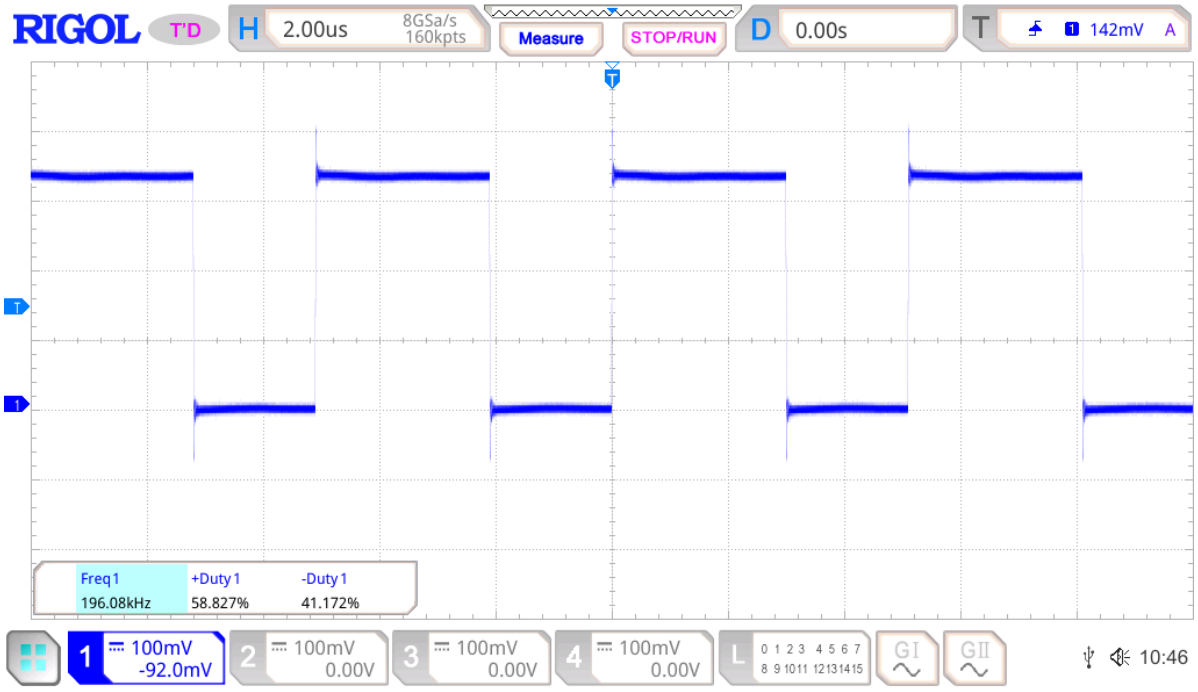

Figure 20. Output PWM signal, when fixed 12V and solar current 7A.

DOI: http://dx.doi.org/10.17509/xxxx.xxxx
p- ISSN 2775-6793 e- ISSN 2775-6815

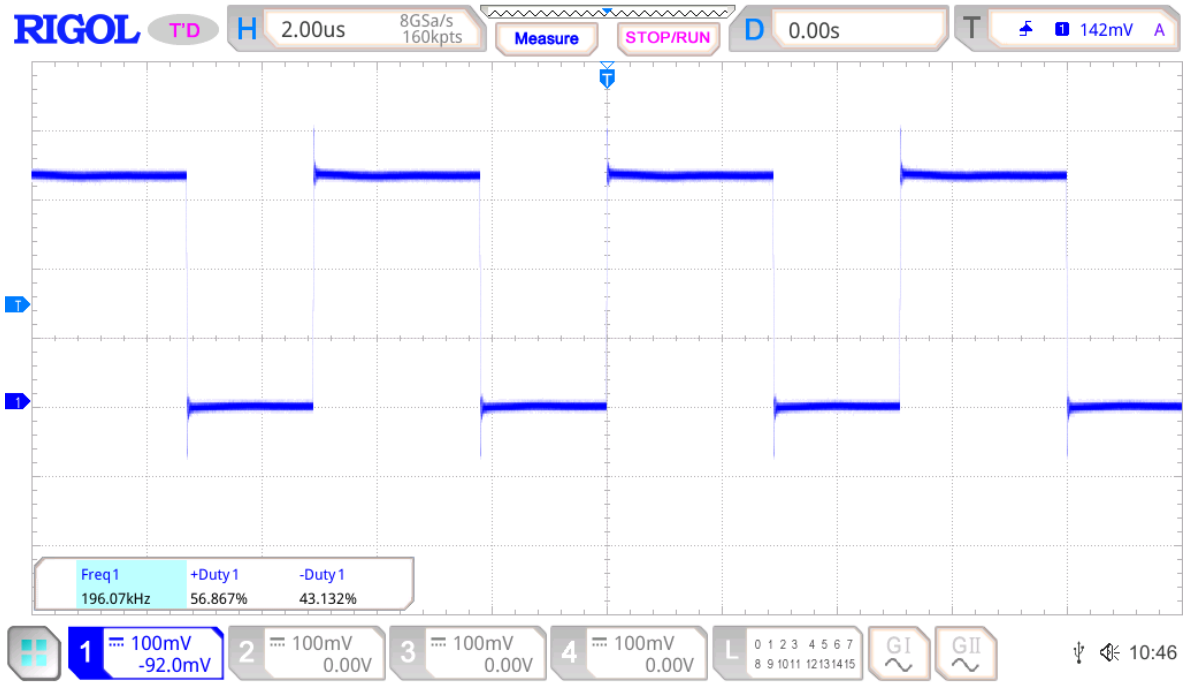

Figure 21. Output PWM signal, when fixed 12V and solar current 3A.

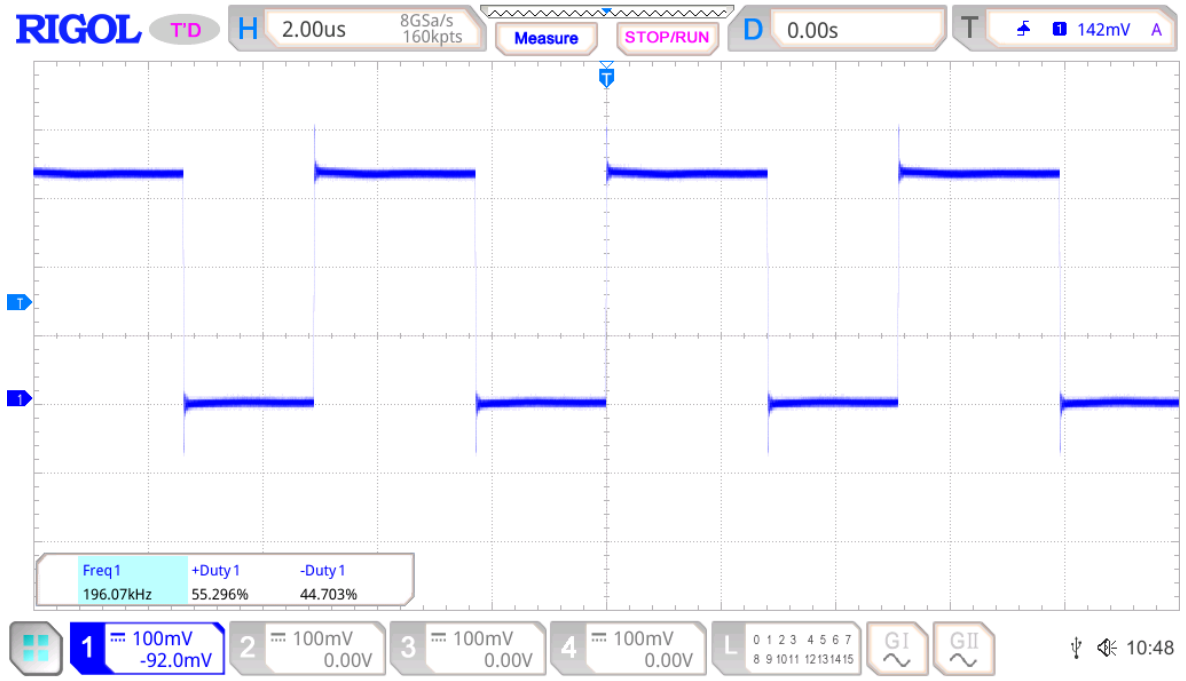

Figure 22. Output PWM signal, when fixed 12V and solar current 1A.

3.2.3.4. Testing Condition 4: Constant Current and Variable Voltage MPPT

Testing the algorithm with constant current and variable voltage is shown in **Figure 23.**

3.2.3.5. Testing Condition 5: Variable Current and Variable Voltage MPPT

Testing the algorithm with variable current and variable voltage is shown in **Figure 24. Figures 25, 26, and 27** show the changing duty of the MPPT PWM signal according to the dynamic voltage and current of the solar panel.

DOI: http://dx.doi.org/10.17509/xxxx.xxxx
p- ISSN 2775-6793 e- ISSN 2775-6815

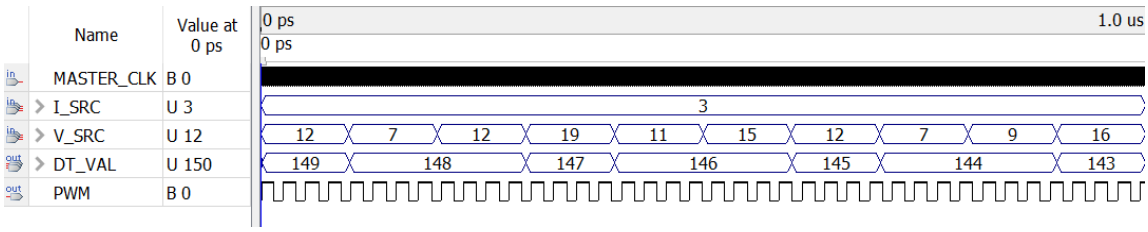

Figure 23. Simulation signal of MPPT constant current and variable voltage.

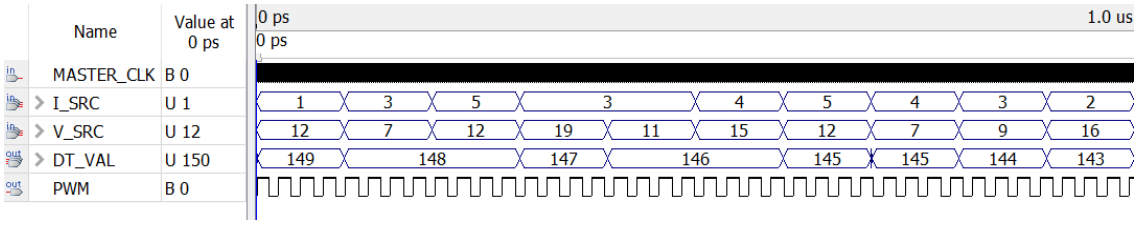

Figure 24. Simulation signal of MPPT variable current and variable voltage.

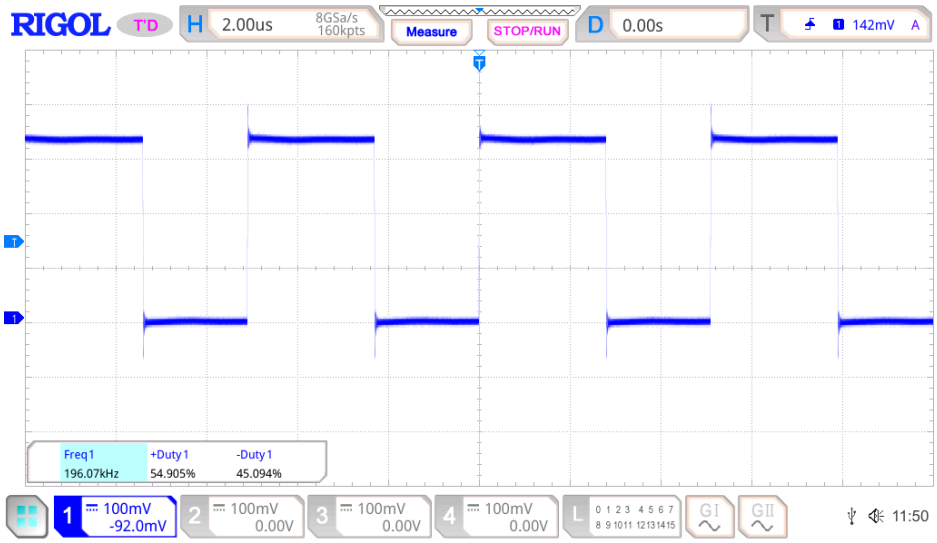

Figure 25. Output PWM signal, when variable current and variable voltage MPPT with solar current 7A.

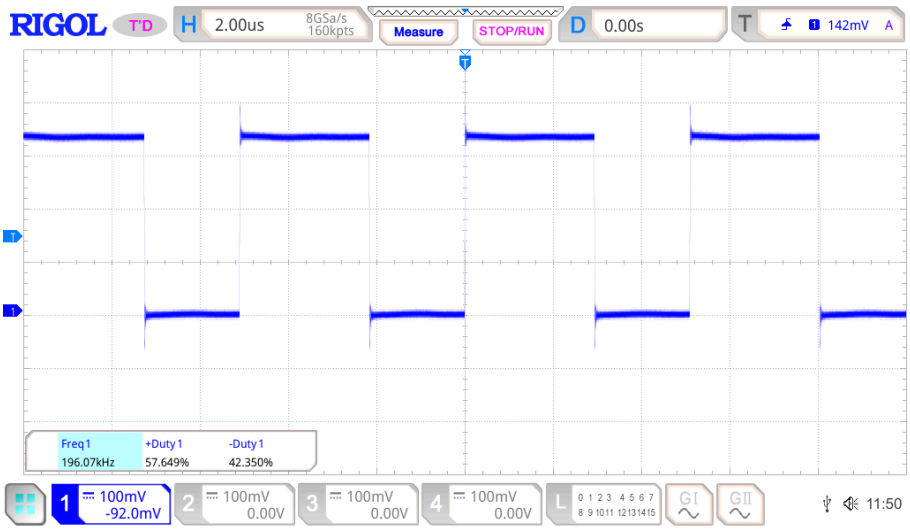

Figure 26. Output PWM signal, when variable current and variable voltage MPPT with solar current 3A.

DOI: http://dx.doi.org/10.17509/xxxx.xxxx
p- ISSN 2775-6793 e- ISSN 2775-6815

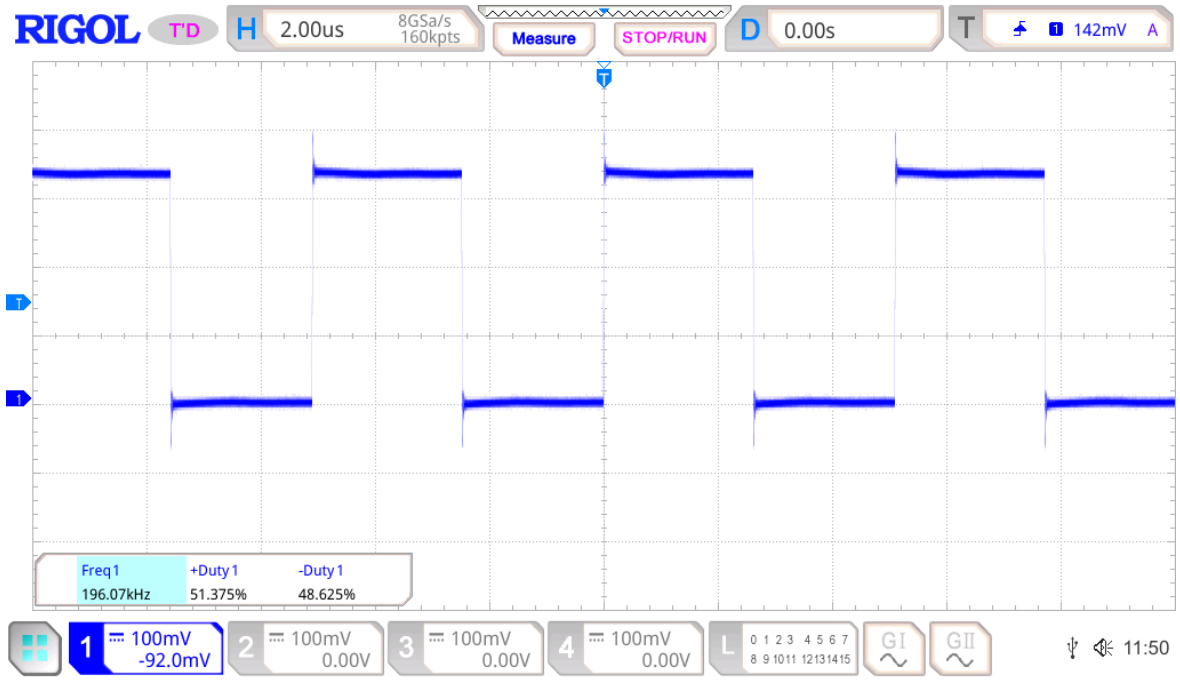

Figure 27. Output PWM signal, when variable current and variable voltage MPPT with solar current 1A.

3.3. Hardware Requirements

3.3.1. Terasic altera cyclone IV EP4CE115F29C7N

The Terasic DE2-115 is an FPGA development board based on Altera Cyclone IV EP4CE115-F29C7N, shown in **Figures 28** and **29**. This FPGA chip has 114,480 logic elements (LEs) and 3,888 embedded memories (Kbits). This feature allows the design without the requirement of external RAM for small applications.

For ADC0804 Analog to Digital Converter, ADC080x is a series of 8-bit analog to digital converter (ADC) with specifications compatible with microprocessors (µP), The specialty of this IC is accepting differential analog voltage input.

In the design of Signal Delay Block **Figure 30**, there is an output port ZERO that generates a clock pulse for every complete process of the function. This signal drives the ADC0804 (pin 3) WR and (pin 5) INTR to generate an 8-bit LSB. The ADC0804 requires a clock source to enable the ADC process. Alternatively, the 150-pF capacitor and 10kΩ resistor are connected to (pin 4) CLKIN and (pin 19) CLKR, as shown in **Figure 31**. The ADC0804 generates no output unless the (pin 3) WR and (pin 5) INTR are pulled low.

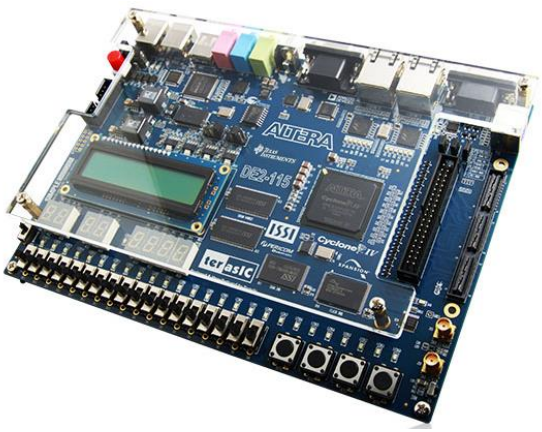

Figure 28. Terasic Altera DE2-115 Cyclone IV EP4CE115F29C7N

DOI: http://dx.doi.org/10.17509/xxxx.xxxx
p- ISSN 2775-6793 e- ISSN 2775-6815

Figure 29. ADC0804 µP compatible analog-to-digital converter.

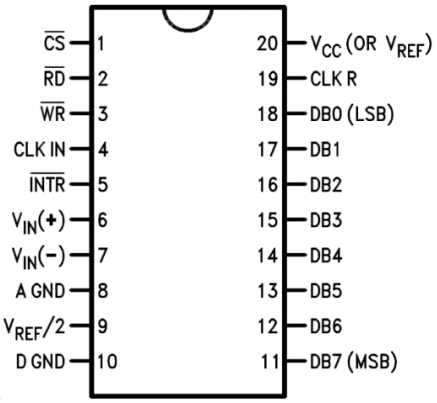

Figure 30. ADC0804 pin pinout mapping.

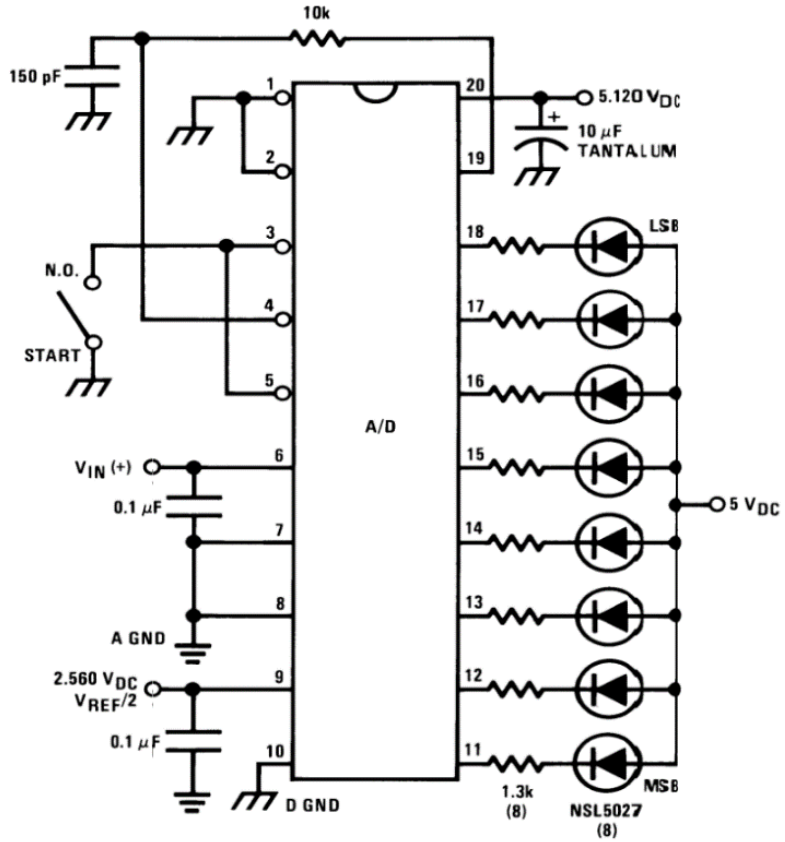

Figure 31. ADC0804 pin configuration.

DOI: http://dx.doi.org/10.17509/xxxx.xxxx
p- ISSN 2775-6793 e- ISSN 2775-6815

3.3.2. VHDL code

3.3.2.1. Signal delay

The signal is explained in the following:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity U_BLK_DLY is

generic(
        constant SET_CLK_CYC: integer := 3
);

port(
        CLK: in std_logic;
        A: in std_logic_vector(7 downto 0);
        O: out signed (7 downto 0);
        O_DLY: buffer signed (7 downto 0);
        ZERO : buffer std_logic
);

end U_BLK_DLY;

architecture model of U_BLK_DLY is
signal TEMP_DLY : std_logic_vector(7 downto 0);

begin
process(CLK) is
variable CLK_CNT : integer := 0;

begin
if falling_edge(CLK) then
        CLK_CNT := CLK_CNT + 1;
        if CLK_CNT = (SET_CLK_CYC + 1) then
                O_DLY <= signed(TEMP_DLY);
                CLK_CNT := 0; ZERO <= '1';
        else
            TEMP_DLY <= A;
                O <= signed(A);
            O_DLY <= O_DLY;
            ZERO <= '0';
        end if;
end if;
end process;

end model;
```

3.3.2.2. Delta calculation

Delta calculation is explained in the following:

DOI: http://dx.doi.org/10.17509/xxxx.xxxx
p- ISSN 2775-6793 e- ISSN 2775-6815

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity U_BLK_SUB is

generic(
        constant SET_CLK_CYC: integer := 3
);

port(
        CLK: in std_logic;
        A, B: in signed (7 downto 0);
        O: buffer signed (7 downto 0);
        ZERO : buffer std_logic
);

end U_BLK_SUB;
architecture model of U_BLK_SUB is

begin
-- CLOCK COUNTER
process(CLK) is
variable CLK_CNT: integer := 0;

begin
if falling_edge(CLK) then
        CLK_CNT := CLK_CNT + 1;
        -- SUBSTRACTION PROCESS
        if CLK_CNT = (SET_CLK_CYC + 1) then
                O <= A - B;
                CLK_CNT := 0;
                ZERO <= '1';
        else
        O <= O;
        ZERO <= '0';
        end if;
end if;
end process;
end model;
```

3.3.2.3. Division calculation

Division calculation is explained in the following:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
LIBRARY lpm;
USE lpm.all;
ENTITY lpm_divide0 IS
```

DOI: http://dx.doi.org/10.17509/xxxx.xxxx
p- ISSN 2775-6793 e- ISSN 2775-6815

```
PORT
(
    clock : IN STD_LOGIC ;
        denom : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
    numer : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
    quotient : OUT STD_LOGIC_VECTOR (7 DOWNTO 0);
    remain : OUT STD_LOGIC_VECTOR (7 DOWNTO 0)
);

END lpm_divide0;
ARCHITECTURE SYN OF lpm_divide0 IS
SIGNAL sub_wire0 : STD_LOGIC_VECTOR (7 DOWNTO 0);
SIGNAL sub_wire1 : STD_LOGIC_VECTOR (7 DOWNTO 0);
COMPONENT lpm_divide

GENERIC (
        lpm_drepresentation : STRING;
        lpm_hint : STRING;
        lpm_nrepresentation : STRING;
        lpm_pipeline : NATURAL;
        lpm_type : STRING;
        lpm_widthd : NATURAL;
        lpm_widthn : NATURAL
);

PORT (
        clock : IN STD_LOGIC ;
        remain : OUT STD_LOGIC_VECTOR (7 DOWNTO 0);
        denom : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
        numer : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
        quotient : OUT STD_LOGIC_VECTOR (7 DOWNTO 0)
);
END COMPONENT;

BEGIN
remain <= sub_wire0(7 DOWNTO 0);
quotient <= sub_wire1(7 DOWNTO 0);
LPM_DIVIDE_component : LPM_DIVIDE

GENERIC MAP (
        lpm_drepresentation => "SIGNED",
        lpm_hint => "MAXIMIZE_SPEED=6,LPM_REMAINDERPOSITIVE=TRUE",
        lpm_nrepresentation => "SIGNED",
        lpm_pipeline => 3,
        lpm_type => "LPM_DIVIDE",
        lpm_widthd => 8,
        lpm_widthn => 8
)
```

DOI: http://dx.doi.org/10.17509/xxxx.xxxx
p- ISSN 2775-6793 e- ISSN 2775-6815

```
PORT MAP (clock => clock,
        denom => denom,
        numer => numer,
        remain => sub_wire0,
        quotient => sub_wire1
);
END SYN;
```

3.3.2.4. Incremental Conductance MPPT

Incremental conductance MPPT is explained in the following:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity INC_MPPT is
-- USER DEFINED DATA TYPE

generic(
        constant SET_CLK_CYC : integer := 3;
        dDT : unsigned (3 downto 0) := x"1";
        max_DT : unsigned (7 downto 0) := x"96" -- 150-
);
-- GPIO

port(
        CLK: in std_logic;
        dI, dV : signed(7 downto 0);
        I_V, dI_dV : signed(7 downto 0);
        -- 8bit PWM
        DT : buffer unsigned(7 downto 0)
);
end INC_MPPT;

architecture model of INC_MPPT is
signal temp_DT : unsigned(7 downto 0) := x"7F"; -- 127-DEC
begin
process(CLK, dI, dV, dI_dV, I_V) is
variable CLK_CNT : integer := 0;

begin
if falling_edge(CLK) then
        CLK_CNT := CLK_CNT + 1;
        if CLK_CNT = SET_CLK_CYC then
                CLK_CNT := 0;
                if dV = 0 then
                        if dI = 0 then
                                temp_DT <= temp_DT;
                        else
                                if dI > 0 then
```

DOI: http://dx.doi.org/10.17509/xxxx.xxxx
p- ISSN 2775-6793 e- ISSN 2775-6815

```
                        temp_DT <= temp_DT + dDT;
                    else
                        temp_DT <= temp_DT - dDT;
                    end if;
                end if;
            else
                if dI_dV = -I_V then
                    temp_DT <= temp_DT;
                    else
                        if dI_dV > -I_V then
                            temp_DT <= temp_DT + dDT;
                        else
                            temp_DT <= temp_DT - dDT;
                        end if;
                    end if;
                end if;
            end if;
end if;

-- ASSIGN OUTPUT DUTY CYCLE
if temp_DT > max_DT then
        temp_DT <= max_DT;
end if;
if temp_DT < x"00" then
        temp_DT <= x"00";
end if;

DT <= temp_DT;
end process;
end model;
```

3.3.2.5. PWM generation

PWM generation is explained in the following:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all; -- USE FOR INCREMENT
entity U_BLK_PWM IS
generic(
        max_val: integer := 255; -- 255 STEP RESOLUTION
        val_bits: integer := 8 -- 8bit RESOLUTION
);

port(
        clk: in std_logic; -- FPGA CLK 50MHZ
        -- DUTY CYCLE VALUE
        val_cur: in std_logic_vector((val_bits - 1) downto 0);
        pulse: out std_logic -- OUTPUT BIT
);
```

DOI: http://dx.doi.org/10.17509/xxxx.xxxx
p- ISSN 2775-6793 e- ISSN 2775-6815

```
end U_BLK_PWM;
architecture model of U_BLK_PWM is
signal cnt: std_logic_vector((val_bits-1) downto 0);
begin
-- COUNTING CLOCK SOURCE
process(clk)

begin
if(clk'event and clk = '1') then
        if(cnt < (max_val - 1)) then
                cnt <= cnt + 1;
        else
                cnt <= (others => '0');
        end if;
end if;
end process;

-- CONTROL PROCESS
process(clk)
begin
if(clk'event and clk = '1') then
        if (val_cur > cnt) then
                pulse <= '1';
        else
                pulse <= '0';
        end if;
end if;
end process;
end model;
```

4. CONCLUSION

According to the simulation results and experiments, the MPPT algorithm can track and generate the output PWM control signal concerning the current and voltage sources. The proposed system has several advantages. First, it is a simple and efficient MPPT algorithm. Second, it is easy to implement in the VHDL language. Third, it can be implemented on a low-cost FPGA development board. The proposed system can be used in a variety of applications, such as solar-powered water pumping, solar-powered street lights, and solar-powered homes. It is a promising solution for the efficient use of solar energy. Due to the design of the system as a *SINGED* and *UNSIGNED* value, the control signal is not as accurate as the *FLOATING-POINT* value system due to the lack of a decimal point in the variation. For further research, a proposal for designing the FPGA system with a *floating-point* value is suggested.

5. AUTHORS' NOTE

The authors declare that there is no conflict of interest regarding the publication of this article. The authors confirmed that the paper was free of plagiarism.

DOI: http://dx.doi.org/10.17509/xxxx.xxxx
p- ISSN 2775-6793 e- ISSN 2775-6815

6. REFERENCES

Al Husaeni, D. N., and Hadianto, D. (2022). The influence of spada learning management system (LMS) on algorithm learning and programming of first grade students at Universitas Pendidikan Indonesia. *Indonesian Journal of Multidiciplinary Research, 2(*1), 203-212.

Br Ginting, S., Maulana, H., Priatna, R., Fauzzan, D., and Setiawan, D. (2021). Crowd detection using YOLOv3-tiny method and Viola-Jones algorithm at mall. *International Journal of Informatics, Information System and Computer Engineering (INJIISCOM)*, *2*(2), 13-22.

Kabalci, E. (2017). Maximum power point tracking (MPPT) algorithms for photovoltaic systems. *Energy Harvesting and Energy Efficiency: Technology, Methods, and Applications, 37,* 205-234.

Riza, L. S., Rosdiyana, R. A., Pérez, A. R., and Wahyudin, A. (2021). The K-Means algorithm for generating sets of items in educational assessment. *Indonesian Journal of Science and Technology*, *6*(1), 93-100.

Sahu, P., and Dey, R. (2022). Maximum power point tracking using adjustable gain based model reference adaptive control. *Journal of Power Electronics*, *22*, 138-150.

Samman, F. A., Srun, C., and Sadjad, R. S. O. (2019). Adaptive look-up table and interpolated pi gain scheduling control for voltage regulator using DC-DC converter. *International Journal of Innovative Computing, Information and Control*, *15*(2), 489-501.

Saran, M. E. A. S., Channareth, S. R. U. N., Sokoeun, U. N., Son, T., Virbora, N. Y., and Saravuth, S. R. I. M. (2022). Optimization of an integrated hybrid onboard charger with high efficiency of mppt solar charger for sustainable energy of 3-wheel solar e-rickshaw and electric vehicles. *International Research Journal of Innovations in Engineering and Technology*, *6*(1), 77.

Srun, C., Chrin, P., Am, S., and Kim, B. (2022). Modeling and simulation of a double-stage single-phase grid-connected PV system. *EPI International Journal of Engineering*, *5*(1), 16-20.

DOI: http://dx.doi.org/10.17509/xxxx.xxxx
p- ISSN 2775-6793 e- ISSN 2775-6815