



A COMPARISON ANALYSIS OF PWM CIRCUIT WITH ARDUINO AND FPGA

A. Zemmouri¹, R. Elgouri^{1,2}, Mohammed Alareqi^{1,3}, H. Dahou¹, M. Benbrahim^{1,2} and L. Hlou¹

¹Laboratory of Electrical Engineering and Energy System Faculty of Sciences, University Ibn Tofail Kenitra, Morocco

²Laboratory of Electrical Engineering and Telecommunications Systems, National Schools of Applied Sciences, University Ibn Tofail Kenitra, Morocco

³Community College, Sana'a, Yemen

E-Mail: abdelkarim.zemmouri@gmail.com

ABSTRACT

This paper discusses a comparison between two PWM circuits the first based on control boards such as Arduino by its IDE (The Arduino integrated development environment and the second FPGA mono processor architecture (MicroBlaze) as the circuit formed by the VHDL. This study compared the space occupied and the power dissipated by PWM circuit between both platforms Arduino and FPGA. The PWM circuit was simulated practically on Arduino boards and FPGA vertex5 by the oscilloscope and data acquisition with Signal Express software provided by NI LabVIEW.

Keyword: PWM, arduino, FPGA, microblaze, VHDL.

1. INTRODUCTION

The Pulse Width Modulation (PWM) function plays as an interface between the control section and the power circuit associated. Thus, its consequences for all performances of the system. The importance of this function gives the different use of platforms such as Arduino [1], CPLD and FPGA [2]. Arduino is an open-source electronics prototyping platform. It contains a programming interface, a microcontroller, voltage regulation and a number of easily accessible digital pins (input / output) so that we can program the microcontroller and use it to do what we like. We can use the digital pins as PWM outputs and easy to use software [1] [3]. It can read information from many different types of sensors and can control other devices such as motors, lights and other sorts of electronics.

The work environment is the Arduino software that has an extended language using C++ libraries. These Arduino boards are based on Atmel microcontrollers ATMEGA8, ATMEGA168, ATMEGA328, etc. Using Xilinx to generate the PWM provides the flexibility to modify the designed circuit without changing the hardware.

Consequently, less hardware, easy and fast circuit modification, a comparatively low cost for a complex circuitry and rapid prototyping make it as the most favorable choice for the PWM generation [4] [5]. FPGAs can generate the PWM signals in two ways:

- Using hardware description language VHDL [6].
- Using embedded processor (MicroBlaze) with its own embedded memory blocks [7].

These blocks are programmed in C language instructions, data and libraries of C++ [8].

This work shows a comparative study between the implementation of PWM circuit in the Arduino or in the FPGA, by either VHDL or C for the case of Microblaze as shown in Figure-1.

The rest of the paper is organized as follows. After introduction, defined the PWM circuit in the Arduino and the FPGA is presented in section 2. Section 3 presents studies the space occupied by the circuit between the two architectures. Section 4 shows the power dissipated. Section 5 shows some results and analysis. Finally, concluding remarks are given in Section 6.

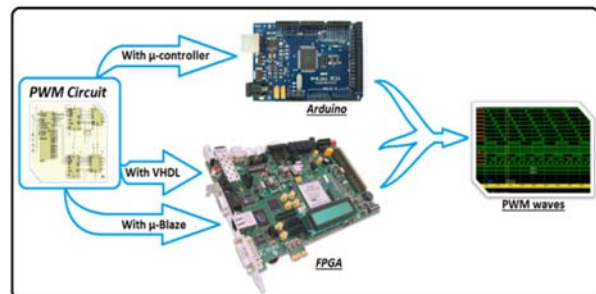


Figure-1. A typical use of arduino microcontroller and FPGA with PWM circuit.

2. PWM CIRCUIT IN ARDUINO MICROCONTROLLER AND FPGA

Pulse-width modulation is an effective method for adjusting the amount of power delivered to an electrical load (Control of Electrical Machines, voltage inverter ...).

Implementation PWM in Arduino or FPGA has advantages like uses fewer components and the possibility of varying the duty cycle and frequency through software.

The hardware resources requirements for generating a signal of PWM are:

- Calculators: the microcontroller or the MicroBlaze Embedded which has modules is called "PCB" (Capture / Compare / PWM).
- Depending upon its method of use, the CCP modules use Timer.
- A CCP module is necessarily related to a timer.



Principle of operation of these resources is done in two steps:

- uses a counter that fixes the frequency of the signal to be generated.
- Comparing continuously the counter value to a fixed value, in a timer counting register for each CCP module, which sets the duty cycle?

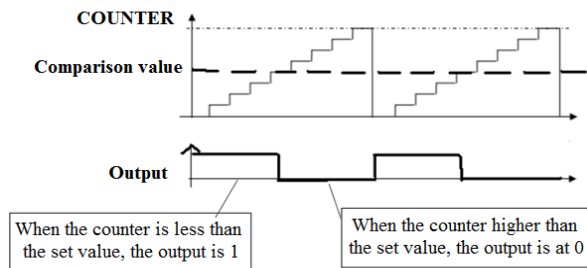


Figure-2. Nomenclature for definition of PWM duty cycle.

3. SPACE OCCUPIED

The FPGA based hardware controller enables to realize almost ideal real-time digital feedback controller because of its capability to realize very fast calculation of the control method with is a few second.

The space reserved for this type of controller integrated PWM offer the opportunity to implement blocks

of regulation, to reduce the time of transfers of two-way information between different devices.

The FPGA-based embedded system is done in two methods proposed by Xilinx:

- By VHDL which is a physical description of the circuit blocks (input/output, clock, process, component,...).
- By a MicroBlaze embedded processor with its devices programmed in C.

These two methods rely on the FPGA as the LUTs, Registers, BUFG/BUFGCTRLs and input/output. Except that the PWM of the Microblaze circuit exceeds that of FPGA by some block as well as BSCANs, DSP48Es, PLL_ADVs, Memory used or Block RAM/FIFO.

The PLL_ADV block is used to generate a clock of 120 MHz, i.e. 1CLK = 8, 33 ns for the PWM system according to the external oscillator 100 MHz of FPGA and to serve as a frequency for a wide range of frequency synthesizer.

The DSP48E receives a clock well synthesize by the PLL_ADV to speed up numeric calculations and ensure good quality results. These blocks embedded in the FPGA are well detailed in table 1 with their percentages of use in each circuit.

Table-1. Space occupied by elements of the Pwm circuit, either by the VHDL or by the embedded processor [10].

Device utilization summary						
	VHDL			Microblaze [10]		
Slice logic utilization	Used	Available	Utilization	Used	Available	Utilization
Number of Slice Registers	36	69,12	1%	1,649	69,12	2%
Number of Slice LUTs	38	69,12	1%	1,897	69,12	2%
Number used as logic	37	69,12	1%	1,754	69,12	2%
Number of occupied Slices	12	17,28	1%	938	17,28	5%
Number with an unused Flip Flop	3	39	7%	945	2,594	36%
Number with an unused LUT	1	39	2%	697	2,594	26%
Number of fully used LUT-FF pairs	35	39	89%	952	2,594	36%
Number of slice register sites lost to control set restrictions	4	69,12	1%	346	69,12	1%
Number of bonded IOBs	6	640	1%	12	640	1%
Number of LOCed IOBs	6	6	100%	12	12	100%
Number of BUFG/BUFGCTRLs	1	32	3%	2	32	6%
Number of BSCANs	--	--	--	1	4	25%
Number of DSP48Es	--	--	--	3	64	4%
Number of PLL_ADVs	--	--	--	1	6	16%
Number used as Memory	--	--	--	138	17,92	1%
Number of BlockRAM/FIFO	--	--	--	4	148	2%
Total Memory used (KB)	--	--	--	144	5,328	2%



Arduino is a microcontroller board based on the ATmega640 / 1280/1281/2560/2561. The board can be programmed and configured with the Arduino Software IDE (Integrated Development Environment).

this series microcontroller provides the following features: 64K / 128K / 256K bytes of In-System Programmable Flash with Read-While-Write capabilities, 4K bytes EEPROM, 8K bytes SRAM, six flexible Timer/Counters with compare modes and PWM (It has 54/86 digital pins of input / output 4/15 which can be used as PWM outputs) [9].

So the Arduino board does not have enough resources, which occupies less space compared to the FPGA (either by the MicroBlaze or alternatively by VHDL). But, In the case of complex projects which include the PWM must use FPGA because it can support high-level algorithms by its memory capacity (internal and external memory).

4. POWER DISSIPATED

The power dissipated by the Arduino output pin is in the order of 165000 mW, on the opposite of that FPGA output pin is about 5664 mW.

This large difference in energy consumption leads us to say that each card field of special use.

The diversity of FPGA and low energy consumption help make the implementation of PWM circuit most reliable, flexible and good quality for the control of external circuits. Table-2 explains the energy balance of the PWM circuit of each card.

Table-2. Power dissipated by Pwm output pins: Fpga / Arduino [10].

Signal name	Drive (mA)	Voltage (mV)	Puissance (mW)
PIN_OUT FPGA	12	472	5664
PIN_OUT ARDUINO	50	3300	165000

The large power dissipation of the Arduino is explained by its cable assembly transmission lines, which consumes power when streaming data. On the other hand, FPGA architecture is purely on board which reduces the length of cables and consequently the dissipated energy.

5. SIMULATION

The results of experimental simulations show that the PWM circuit may exhibit an excellent driving performance for both cards. Arduino's performances exhibited by different simulations and applications. Among the most remarkable simulations is that measured relation between the PWM duty cycle and the measured normalized output by the 10-bit PWM output, which shows a proportionality between the two, as shown in Figure-3.

PWM output using the Timer1 library connecting the load directly to the PWM output pin of Arduino. The blue points are the measured values and black line the linear regression fit [11].

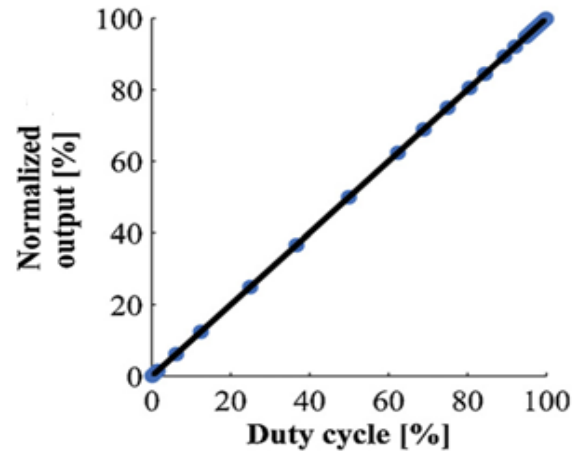


Figure-3. The proportionality between the duty cycle and the 10-bit PWM output [11].

Using a data acquisition card (DAQ, National Instruments TM USB-6210) with its supplied NI Lab VIEW Signal Express software for interactive data logging.

The measured 10-bit PWM signals from Arduino at different duty cycles (shown in Figure-4) using the Timer1 library. The PWM frequency was the default 400 Hz (period of one cycle 2.5 ms). These PWM signals have the same behavior as those of the FPGA board with a frequency band between 4 kHz and 570 Hz.

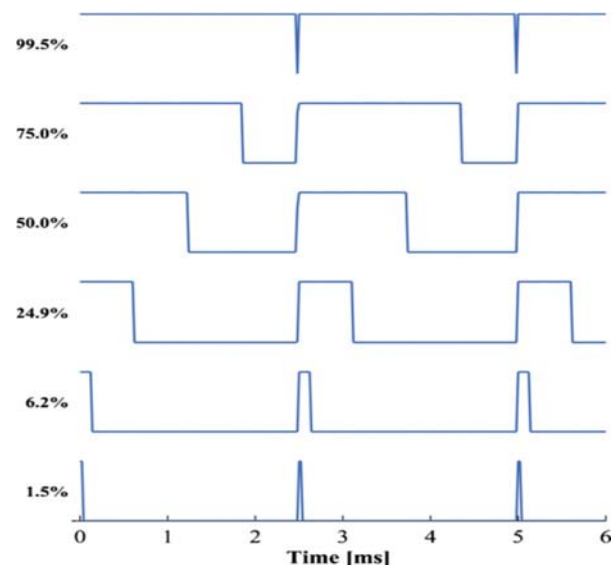


Figure-4. Arduino PWM signals at different duty cycles [11].

The simulations of the PWM circuit on FPGA are displayed by the oscilloscope, which the duty cycles 6.43%; 75.34% and 98% as shown in Figures 5 (a), (b) and (c).

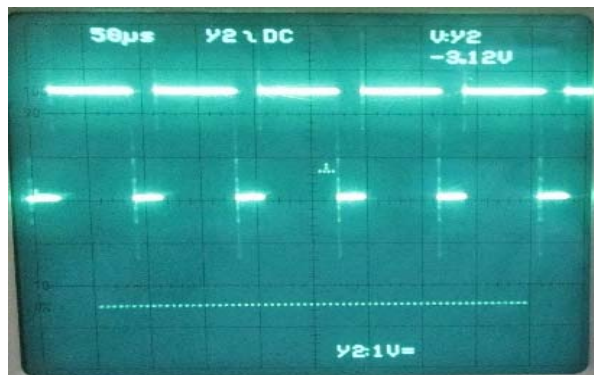


When the output pin is active (state 1) or off (state 0), the output voltage does not stabilize.

It takes the form of small undulations the same logical state; hence, the need to implement other embedded processing block as well as PID (Proportional-Integral-Derivative) for the regulation against all transitions that occur during rising edge and falling edge as showing in the Figure-5 (b) and (c).



a) 6, 43 %.



b) 75, 34 %.



c) 98 %.

Figure-5. Output PWM of FPGA in an oscilloscope with a different duty cycle: a) 6, 43%, b) 75, 34% et c) 98%.

6. CONCLUSIONS

The PWM circuit using Arduino is designed for small-scale projects, which require power but do not require large memory.

However, that is based on the FPGA offers a wide range of space to create treatments blocks and purely embedded recordings. As well as the data processing in parallel for large scale applications with low power consumption.

REFERENCES

- [1] D. Radianto, G.M. Dousoky, M. Shoyama. 2015. Design and Implementation of Fast PWM Boost Converter based on Low Cost Microcontroller for Photovoltaic Systems. IEEE IECON2015-Yokohama. pp. 2324-2328.
- [2] E. Koutroulis, A. Dollas, K. Kalaitzakis. 2005. High-frequency pulse width modulation implementation using FPGA and CPLD ICs. Journal of Systems Architecture. pp. 332-344.
- [3] L.Acho. 2015. A discrete-time chaotic oscillator based on the logistic map: A secure communication scheme and a simple experiment using Arduino. Elsevier Journal of the Franklin Institute. 352, pp. 3113-3121.
- [4] S. Mekhilef, N. AbdRahim. 2002. Xilinx FPGA based three-phase PWM inverter and its application for utility connected PV system. IEEE TENCON.2002.1182752, pp. 2079-2082.
- [5] A. Fratta, G. Griffero, S. Nieddu. 2004. Comparative Analysis among DSP and FPGA-based Control Capabilities in PWM Power Converters. IEEE-IECON'2004, Pusan (Corea del Sud). pp. 257-262.
- [6] S. Mekhilef, A. Masaoud. 2006. Xilinx FPGA based multilevel PWM single phase inverter. IEEE Engineering e-Transaction. 1(2): 40-45.
- [7] A. Zemmouri, M. Alareqi, R. Elgouri, M. Benbrahim, L. Hlou. 2015. Integration and Implimentation System-On-A-Programmable-Chip (Sopc) In FPGA. Journal of Theoretical and Applied Information Technology. 76: 127-133.
- [8] M.R. Khalil, S.B. Mahmood. 2013. Designing of a pulse width modulation system using embedded system design techniques. Journal of Theoretical and Applied Information Technology. 49(1): 101- 106.



- [9] atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet
(http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf).
- [10] A. Zemmouri, R. Elgouri, M. Alareqi, M. Benbrahim, L. Hlou. 2017. Design and Implementation of Pulse Width Modulation Using Hardware/Software MicroBlaze Soft-Core. International Journal of Power Electronics and Drive Systems (IJPEDS). 8(1): 167-175.
- [11] P. Teikari, R.P. Najjar, H. Malkki, K. Knoblauch, D. Dumortier, C. Gronfier, H. M. Cooper. 2012. An inexpensive Arduino-based LED stimulator system for vision research. Journal of Neuroscience Methods. 211, pp. 227-236.