

Multicore Based Open Loop Motor Controller Embedded System for Permanent Magnet Direct Current Motor

¹Baskaran, K. and ²K.S. Vijula Grace

¹Department of CSE, Government College of Technology, Coimbatore, India

²Department of ECE, Noorul Islam University, Kanyakumari District, India

Abstract: Problem statement: In an advanced electronics world most of the applications are developed by microcontroller based embedded system. **Approach:** Multicore processor based motor controller was presented to improve the processing speed of the controller and improve the efficiency of the motor by maintaining constant speed. It was based on the combination of Cortex processor (Software core) and Field Programmable Gate Arrays (FPGA, Hardware core). These multicore combination were help to design efficient low power motor controller. **Results:** A functional design of cortex processor and FPGA in this system was completed by using Actel libero IDE and IAR embedded IDE software PWM signal was generated by the proposed processor to control the motor driver circuit. All the function modules were programmed by Very-High-Speed Integrated Circuit Hardware Description Language (VHDL). The advantage of the proposed system was optimized operational performance and low power utility. Multicore processor was used to improve the speed of execution and optimize the performance of the controller. **Conclusion:** Without having the architectural concept of any motor we can control it by using this method. This is an low cost low power controller and easy to use. The simulation and experiment results verified its validity.

Key words: FPGA Cortex-M1, FPGA, core PWM, OLED, core UART core AHB, coremem Ctrl

INTRODUCTION

In an electronic era the requirement of electronic systems are increased exponentially. The size of the system must be reduced for betterment of easy way of using. Power requirement is also to be considering while designing such a systems. Now a day some of the advanced 32 bits processors are used to design an embedded based electronic system. ARM 7 (Advanced RISC Machine) processor is widely used to design PDAs, Controllers for Automation, LCD monitors and 3G mobile phones. Instead of using single core processor like ARM core if we use more than one core ie., multicore processor to establish a single task, we may get better performance and very high throughput.

Optimization of code, memory size, execution speed of such multicore is very important for an improved performance. In recent years, programmable logic devices have developed rapidly, especially the Field Programmable Gate Arrays (FPGA) (Al-Ayasrah *et al.*, 2006). It has low power consumption, flexible programming, shorter development cycle, easier to transplant and other advantages. The highly

configurable Cortex-M1 processor provides a good balance between size and speed for embedded applications. It runs at over 70 MHz in Actel M1-enabled ProASIC3 and Fusion devices. The processor runs a subset of the new Thumb-2 instruction set and features support for tightly coupled memory and a sophisticated low latency interrupt controller to improve embedded performance and capabilities A configurable Nested Vectored Interrupt Controller (NVIC) is facilitates low latency interrupt and exception handling and also simplifies the programming. In order to satisfy most of the control requirements like fast real-time computation; reliability, accuracy and safety in harsh environment., recent research studies proved that digital hardware solutions, such as Field-Programmable Gate Array (FPGA), are an appropriate alternative over software solutions (DSP and μ Controllers) and analog solutions

In this study Cortex-M1 and FPGA combination of Multicore is used to design an embedded system for controlling motor (Lin *et al.*, 2010; Idkhajine *et al.*, 2009). In functional design of FPGA, we use modularized design with Very-High-Speed Integrated Circuit Hardware Description Language (VHDL).

Corresponding Author: K.S. Vijula Grace, Department of ECE, Noorul Islam University, Kanyakumari District, India

Here section 1 describe about hardware function dividing, section 2 describe about Core Designing using IDE. Section 3 describe about Function modules implemented in core. Section 4 Material and methods. Finally it is concluded with simulation result

SECTION I

Hardware structure and function dividing: A. Hardware structure: Based on the structure of multicore processor combination the Hardware circuit is designed to perform the better control of PMDC motor .Hardware circuit consists of buffer optocoupler, Driver circuit. Optocoupler is act as an isolator between control circuit and power circuit.Optocoupler circuit and pin diagram is shown in Fig. 1 and 2 respectively. Driver circuit is for controlling the ON-OFF time of IGBT which output is directly connected to motor circuit to control the speed of the motor. Moc sensor (MoC 7811) is normally used as positional sensor to sense the speed.

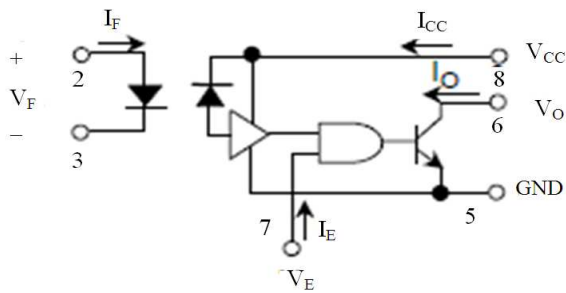
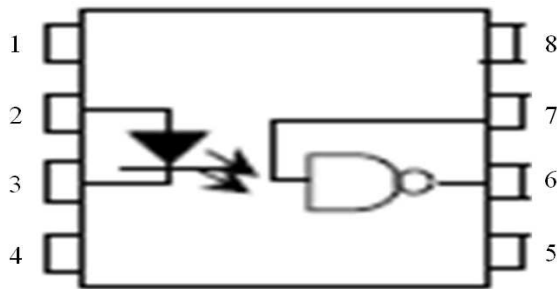


Fig. 1: Circuit diagram of optoisolator



- 1: N.C.
- 2: Anode
- 3: Cathode
- 4: N.C.
- 5: GND
- 6: Output (Open collector)
- 7: Enable
- 8: V_{CC}

Fig. 2: Pin configuration of optoisolator

Function dividing: Cortex-M1 core is configured with the following peripherals corePLL, corePWM, coreAHB, coreGPIO, coreI2C. PLL is locked control the maximum frequency up to 40MHz from 30 KHz input from RC oscillator. PLL output is given as a input for cortex-m1 and all other peripherals available on the design. CorePWM is a general purpose, multi-channel Pulse Width Modulator (PWM) module for motor control, tone generation, battery charging, heating elements and more (Lin *et al.*, 2010). In addition, there is a special “Low Ripple DAC” mode that creates a pulse train that does not modulate the width of the pulses PWM generate a pulse with frequency 10KHz. CoreTimer is for generating a delay and also to count the pulse output.

It is configured to generate an interrupt when counter reaches one min delay to calculate Revolution of the motor Per Min (RPM).

Corei2c is to implement i2c protocol on core to communicate i2c based devices like Optical Light Emitting Diode (OLED). coreGpio provides an Advanced Peripheral Bus (APB) register based interrupt to up to 32 general purpose input and output. Moc sensor output is given as an input to coreGpio, it is configure to generate an interrupt for every rising and falling edge of the pulse. That interrupt is processed through Non Vectored Interrupt Controller (NVIC) module on core without any delay to count RPM.

About cortex-M1: ARM Cortex-M1 is a general purpose, 32-bit microprocessor that offers high performance and small size in FPGAs. ARM Cortex-M1 runs a subset of the Thumb-2 instruction set (ARMv6-M), which includes all base 16-bit Thumb instructions and a few Thumb-2 32-bit instructions (BL, MRS, MSR, ISB, DSB and DMB). This enables very tight and efficient code to be written for the processor that is ideal for the limited memory typically found in embedded applications. The main blocks in ARM Cortex-M1 are the processor core, the Nested Vectored Interrupt Controller (NVIC), the AHB interface and the debug unit. The processor core supports 13 general purpose 32-bit registers, including the Link Register (LR), Program Counter (PC), Program Status Register (xPSR) and two banked Stack Pointers (SP).

The NVIC is closely coupled to the ARM Cortex-M1 core to achieve low-latency interrupt processing. The versions currently available for use in M1 devices support 1 interrupt with 4 levels of priority. Future versions will support up to 32 interrupts. To simplify software development, the processor state is automatically saved on interrupt entry and restored on interrupt exist, with no instruction overhead.

The ARM Cortex-M1 Thumb instruction set's 16-bit instruction length allows it to approach twice the density in memory of standard 32-bit ARM code while retaining most of the ARM performance advantage over a traditional 16-bit processor using 16-bit registers. This is possible because Thumb code operates on the 32-bit register set in the processor. Thumb code is able to provide up to 65% of the code size of ARM and 160% of the performance of an equivalent ARM processor connected to a 16-bit memory system.

ARM Cortex-M1-Enabled FPGA: ARM Cortex-M1 is available for use in a growing number of Actel M1 devices. Fusion integrates a 12-bit analog-to-digital converter, as many as 40 analog I/Os, up to 8 Mbits of flash memory and FPGA fabric all in a single device. When used in conjunction with a softprocessor such as ARM Cortex-M1, Actel Fusion devices represent the definitive soft MCU platform.

Cortex-M1 hardware design description: The Cortex-M1 processor system uses CoreAI, which allows the processor to configure, control and interact with the Analog Block inside the Actel Fusion FPGA. CoreAI has the following features:

- Thin processor interface around Fusion AB (Analog Block) hard macro
- ADC conversions controlled by processor writes
- Internal logic interface for controlling the ACM (Analog Configuration MUX)
- Internal logic to divide clock for generating ACM clock
- Optional hardware-controlled inputs to directly control some AB functions
- Interrupt logic for various events (such as end of ADC conversions)

The UART in the system connects to an off-chip USB-to-UART chip, which allows you to communicate with the target system via a COM port on your machine (using HyperTerminal).

To transmit data, it is first loaded into the transmit data buffer in normal mode and into the transmit FIFO in FIFO mode. Data can be loaded into the data buffer or transmit FIFO until the TXRDY signal is driven inactive. The transmit state machine will immediately begin to transmit data and will continue transmission until the data buffer is empty in normal mode and until the transmit FIFO is empty in FIFO mode. The transmit state machine first transmits a START bit, followed by the data (LSB first), then the parity (optional) and finally the STOP bit. The data buffer is double-buffered

in normal mode, so there is no loading latency. The receive state machine monitors the activity of the rx signal. Once a START bit is detected, the receive state machine begins to store the data in the receive buffer in normal mode and the receive FIFO in FIFO mode. When the transaction is complete, the rxrdy signal indicates that valid data is available. Parity errors are reported on the parity_error signal (if enabled) and data overrun conditions are reported on the overflow signal. Framing errors are reported on the FRAMING-ERR signal.

UART also included are 4 output bits to LEDs and 2 input bits from push-buttons or DIP switches (depending on the target board). The Fusion Advanced Development Kit and the Fusion Embedded Development Kit boards contain an Actel Fusion AFS1500 device that has 1 MByte of embedded flash memory (also referred to as nonvolatile memory, or NVM) and 30 KBytes of internal SRAM. The Fusion Embedded Development Kit board has 1 Mbyte of SRAM, comprised of two 4 Mbit \times 16 bit chips. The Fusion Advanced Development Kit board has 2 Mbytes of SRAM, comprised of two 1 Mbit \times 16 bit chips. This hardware design connects to all of these memories, not necessarily using the entire memory space of each device. Actel's Fusion devices have an on-chip 100 MHz RC oscillator. You will feed this clock source to a PLL inside the Fusion device that modifies the clock frequency. The output of the PLL is the system clock.

SECTION 2

Core designing using IDE: The Actel Fusion® mixed-signal FPGA satisfies the demand from system architects for a device that simplifies design and unleashes their creativity. Actel Fusion mixed-signal FPGAs bring the benefits of programmable logic to many application areas, including power management, smart battery charging, clock generation and management and motor control.

Here Actel fusion device is used to configure our multicore device with FPGA core and Cortex-M1 core. CoreI2C provides an APB-driven serial interface, supporting I2C, SMBus and PMBus data transfers. Data transfers up to at least 400 kbps nominally; faster rates can be achieved depending on external load and/or I/O pad circuitry.

Actel Fusion Development Kit boards contain an Actel Fusion AFS1500 device that has 1 MByte of embedded flash memory (also referred to as nonvolatile memory, or NVM) and 30 KBytes of internal SRAM. The Fusion Embedded Development Kit board has 1 Mbyte of SRAM, comprised of two 4 Mbit \times 16 bit chips. The Fusion Advanced Development Kit board

has 2 Mbytes of SRAM, comprised of two 1 Mbit \times 16 bit chips. This hardware design connects to all of these memories, not necessarily using the entire memory space of each device. Actel's Fusion devices have an on-chip 100 MHz RC oscillator. You will feed this clock source to a PLL inside the Fusion device that modifies the clock frequency. The output of the PLL is the system clock.

Features of actel fusion family of mixed-signal FPGAs: High-Performance Reprogrammable flash Technology, Embedded Flash Memory, Integrated A/D Converter (ADC) and Analog I/O, On-Chip Clocking Support, low power consumption- single 3.3 V Power Supply with On-Chip 1.5 V Regulator and Sleep and Standby Low-Power Modes. In-System Programming (ISP) and Security, variable-aspect-ratio 4,608-Bit SRAM Blocks. Flash-based Fusion devices are live at power-up and do not need to be loaded from an external boot PROM. On-board security mechanisms prevent access to the programming information and enable secure remote updates of the FPGA logic. Designers can perform secure remote in system reprogramming. Fusion devices utilize a 128-bit flash-based key lock and a separate AES key to secure programmed IP and configuration data. The Flash ROM data in Fusion devices can also be encrypted prior to loading. Flash-based FPGAs store their configuration information in on-chip flash cells. Once programmed, the configuration data is an inherent part of the FPGA structure and no external configuration data needs to be loaded at system power-up. Therefore, flash-based Fusion FPGAs do not require system configuration components such as EEPROMs or microcontrollers to load device configuration data. Fusion devices have low dynamic power consumption and support both low power standby mode and very low power sleep mode, offering further power

Advanced architecture: The Fusion device consists of several distinct and programmable architectural features, including the following:

- Embedded memories
- Flash memory blocks
- Flash ROM
- SRAM and FIFO
- Clocking resources
- PLL and CCC
- RC oscillator
- Crystal oscillator

- No-Glitch MUX (NGMUX)
- Digital I/Os with advanced I/O standards
- FPGA VersaTiles
- Analog components
- ADC
- Analog I/Os supporting voltage, current and temperature monitoring
- 1.5 V on-board voltage regulator
- Real-time counter

SECTION 3

Function modules implemented in core: All the functional modules are programmed by very high speed Integrated circuit hardware description Language (VHDL). Fusion device controller development process is supported by specialized software tools. Design Flow includes the following steps:

- Create a smart design components with in IDE. In order to design an Environment We can select the required processor and coprocessor from the available library to create an design environment as per our requirements. It consists of CoreI2C, CoreUART, CorePLL, CoreMemCtrl, CoreAHB, CorePHB and main Cortex-M1 processor. All the above said components are drag and place to create smart design
- Connect signals manually and automatically. Promote signals to top level. Few signals are consider as a top level signals, they are mainly assigned to set up a outside pins for the design core
- Create flash memory system and design storage client to allow to include software code
- Perform pre synthesis functional simulation
- Place and routing the design
- Program the FPGA

Figure 3, shows the design of the controller which obtained by following the above said design flow steps. This can be created in the Libero IDE workspace using the same IDE.

Core AHB Lite: Core AHB Lite is a multi-master AHB bus. It provides 2 AHB master interfaces and 16 AHB slave interfaces. Each master is connected to all 16 slaves. The number of slaves can be enabled or disabled through the configuration. The slaves that are not enabled (selected) will be optimized during synthesis. If none of the slaves are elected for the given master, then the master interface will be disabled.

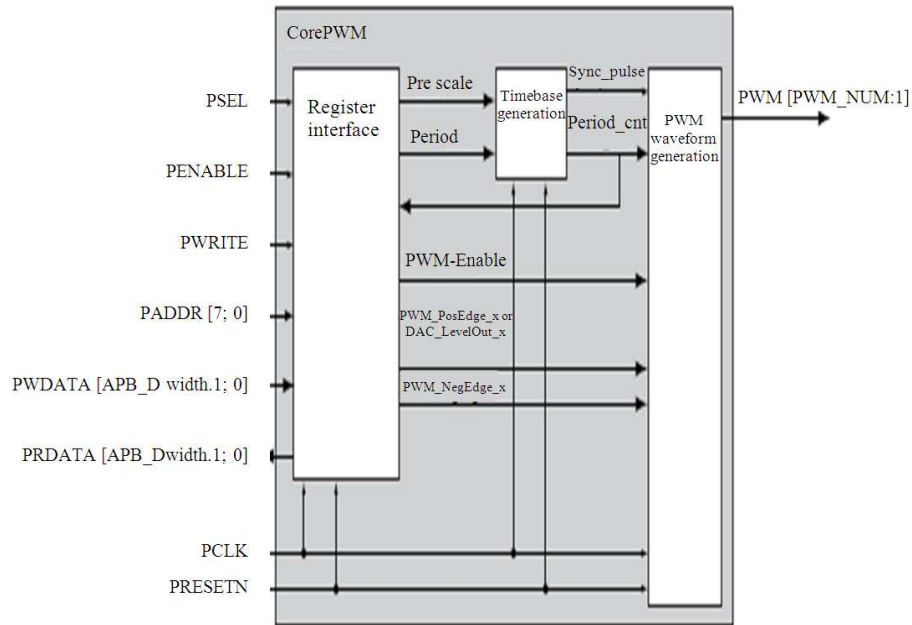


Fig. 4: Principle of generating PWM waveform

Table 1: CorePWM I/O signal descriptions

Name	Type	Description
System signals		
PRESETN	Input	Active low asynchronous reset
PCLK	Input	System clock-all operations and status shall be synchronous to the rising edge of this clock signal
Microcontroller signals		
PSEL	Input	Select line for CorePWM
PENABLE	Input	Read output enable
PWRITE	Input	Write enable
PADDR [7:0]	Input	Register address
PWDATA [APB-DWIDTH-1:0]	Input	Write address/data input
PRDATA [APB-DWIDTH-1:0]	Output	Read data output
PWM signals		
PWM [PWM-NUM:1]	Output	Pulse width modulation output

Opto coupler is act as an isolator between power circuit and motor control circuit. In an electronics circuits an opto-isolator, also called an optocoupler, photocoupler, or optical isolator, is an electronic device designed to transfer electrical signals by utilizing light waves to provide coupling with electrical isolation between its input and output. Opto-isolator is to prevent high voltages or rapidly changing voltages on one side of the circuit from damaging components or distorting transmissions on the other side. In our application we use The TOSHIBA 6N137 consist of a high emitting diode and a one chip photo IC. This unit is 8-lead DIP package. LSTTL/TTL compatible: 5V Supply.

Ultra high speed: 10MBd. Guaranteed performance over temperature: 0-70°C. High isolation voltage: 2500 Vrms min.

A driver circuit is an electrical circuit or other electronic component used to control another circuit or other component, such as a high-power transistor like Insulated Gate Bipolar Transistor (IGBT). They are usually used to regulate current flowing through a circuit. The IR2110/IR2113 are high voltage, high speed power MOSFET and IGBT drivers with independent high and low side referenced output channels.

Proprietary HVIC and latch immune CMOS technologies enable ruggedized monolithic construction. Logic inputs are compatible with standard CMOS or LSTTL output, down to 3.3V logic. The output drivers feature a high pulse current buffer stage designed for minimum driver cross-conduction. Propagation delays are matched to simplify use in high frequency applications. The floating channel can be used to drive an N-channel power MOSFET or IGBT in the high side configuration which operates up to 500 or 600 volts. A chopper circuit is used to break up the input signal so that it can be processed as if it were an AC signal, then integrated back to a DC signal at the output. In this way, extremely small DC signals can be amplified.

A permanent-magnet motor does not have a field winding on the stator frame, instead relying on permanent magnets to provide the magnetic field against which the rotor field interacts to produce torque. Compensating windings in series with the armature may be used on large motors to improve commutation

under load. Because this field is fixed, it cannot be adjusted for speed control. Permanent-magnet motors are convenient in miniature motors to eliminate the power consumption of the field winding.

MOC7811 is a slotted Opto isolator module, with an IR transmitter and a photodiode mounted on it. Performs Non-Contact Object Sensing. This is normally used as positional sensor switch (limit switch) or as Position Encoder sensors used to find position of the wheel. Here it is used to convert the speed to pulse output. This change in the logic level can be sensed by the microcontroller or by discrete hardware. Figure 5 Shows the block representation of the multicore based motor controller embedded system.

Timer to generate interrupt: The CoreTimer module is an APB slave that provides access to an interrupt-generating, programmable decrementing counter. The width of the decrementing counter in the CoreTimer module can be statically configured as either 16 or 32 bits. Programmable registers provide a means to dynamically control the operation of the timer. If the interrupt is enabled, an interrupt is generated when the decrementing counter reaches zero. There are two modes of operation available for Core Timer: Continuous mode and One-Shot Timer mode. An interrupt is generated when the counter reaches zero and is only cleared when the Interrupt Clear Register, TimerIntClr, is written to. A register holds the value until the interrupt is cleared.

The counter in CoreTimer is clocked with PCLK, but a clock enable signal produced by the prescaler is used to enable the counter to operate from a lower effective frequency than that at which PCLK is running. The interval between clock enable pulses can be adjusted via the Prescale field in the Timer Control Register. It is possible to generate a clock enable pulse every 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1,024 periods of PCLK. Figure 6 Shows the block diagram of UART. It has UART and APB I/FW rapter block, Transmit FIFO and receive FIFO block, transmit state machine, Receive state machine block. Baud value from UART and APB I/FW rapter block is given as a input for the baud generator which will send a signal to the transmit and receive state machine. UART and APB I/FW rapter block have both APB and UART related signals group.

About PMDC motor: Permanent Magnet DC motors are useful in a range of applications, from battery powered devices like wheelchairs and power tools, to conveyors and door openers, welding equipment, X-ray and pumping equipment, to name a few. They are

frequently the best solution to motion control and power transmission applications where compact size, wide operating speed range, ability to adapt to a range of power sources or the safety considerations of low voltage are important. Their ability to produce high torque at low speed make them suitable substitutes for gear motors in many applications. Because of their linear speed-torque curve, they particularly suit adjustable speed and servo control applications where the motor will operate at less than 5000 rpm inside these motors, permanent magnets bonded to a flux-return ring replace the stator field windings found in shunt motors. A wound armature and mechanical brush commutate ion system complete the motor. The permanent magnets supply the surrounding field flux, eliminating the need for external field current. This design yields a smaller, lighter and energy efficient motor (Wang *et al.*, 2008).

Communication between core to core: In motor control system, high-speed and reliable communication between the Cortex-M1 and FPGA is required. CoreUART is a serial communication controller with a flexible serial data interface that is intended primarily for embedded systems. Core UART apb supports two modes: programmable and fixed. These modes enable the user to set parameters as fixed or as configurable during system operation.

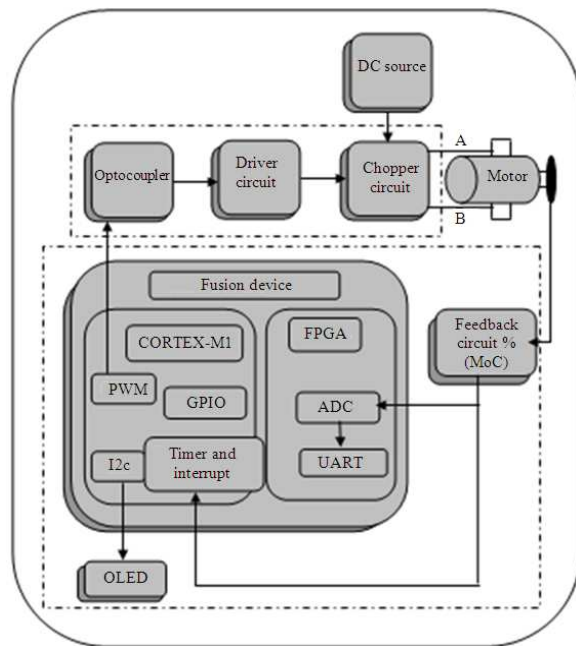


Fig. 5: Block diagram of multicore based motor controller embedded system

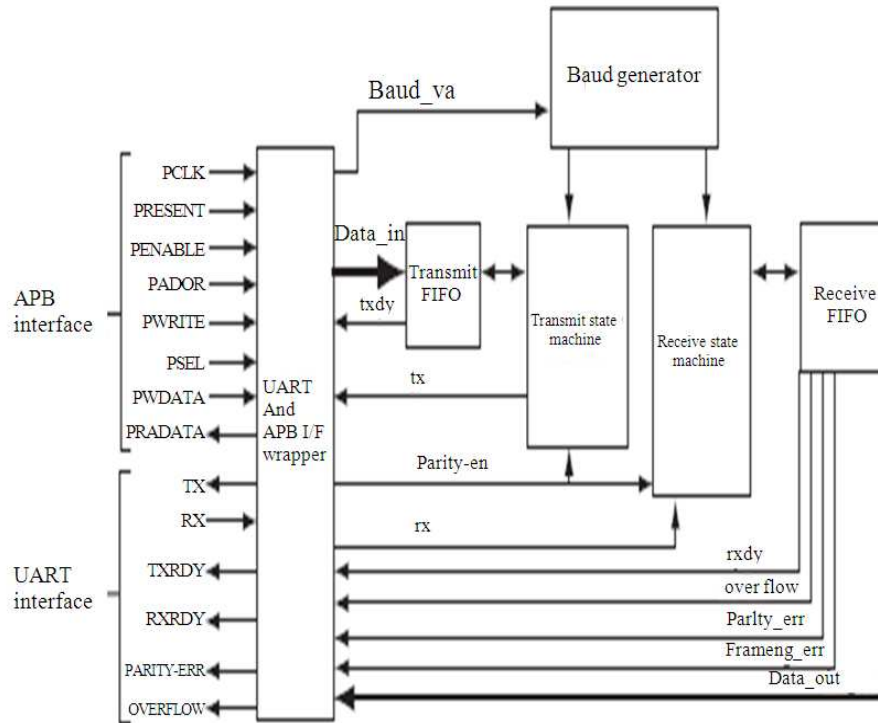


Fig. 6: Block diagram of UART with FIFO

To transmit data, it is first loaded into the transmit data buffer in normal mode and into the transmit FIFO in FIFO mode. Data can be loaded into the data buffer or transmit FIFO until the TXRDY signal is driven inactive. The transmit state machine will immediately begin to transmit data and will continue transmission until the data buffer is empty in normal mode and until the transmit FIFO is empty in FIFO mode. The receive state machine monitors the activity of the rx signal. Once a START bit is detected, the receive state machine begins to store the data in the receive buffer in normal mode and the receive FIFO in FIFO mode (Al-Ayasrah *et al.*, 2006). Communication between Core to core is handover by means of using such a controller with flexible serial communication.

F.RPM calculation and speed display: Moc Sensor is used to sense the speed and given it in the form of pulse. That is given to ADC and the corresponding digital value is transmitted and displayed in the appropriate display (OLED). General formula for speed is distance divided by time. RPM calculation is based on the signals period count set in the counter due to the interrupt produced by the timer.

Software coding: In this software coding are generated by using VHDL language to initialize the core processor pins and functioning of particular processor. IAR Embedded Workbench IDE is used to create an hex file for initialization of the functions of the corresponding core processors generated and used for this application field, In this case C language is used to write a coding. Some of the code optimization techniques also to be used for better performance.

SECTION 4

MATERIALS AND METHODS

In this proposed system Libero IDE software is used to design a controller VHDL language is used to design a softblocks of this design. The analog pulse width modulated (PWM) signal generated by the CorePWM is used to control the ON and OFF of the IGBT available in the Driver circuit of the motor controller. These pulses may varied by changing the signal frequency either by manual or by automation. In this paper 10KHz fixed PWM signal is used to controll the IGBT Drive. Figure 7 shows th 10KHz PWM output obtained from the cortex-M1 core PWM, Fig. 8 gives the ripple free output wave.

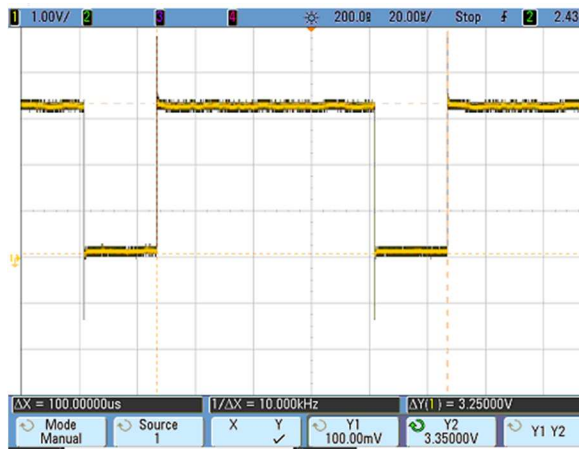


Fig. 7: 10 KHz PWM waveform generated by using Cortex-M1 processor with ripple

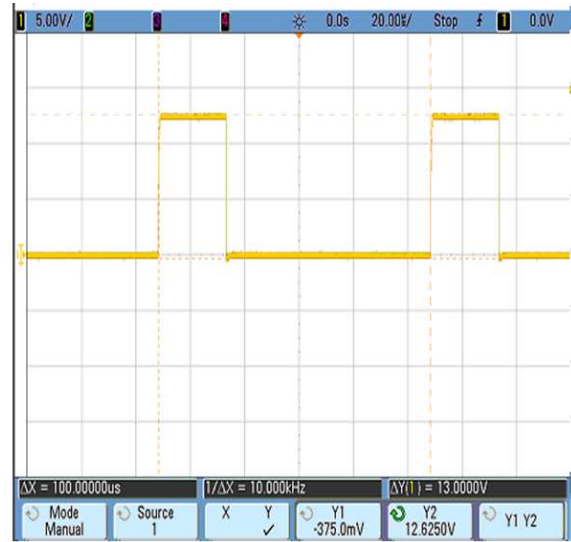


Fig. 9: Output of optoIsolator



Fig. 8: Output of Driver circuit-Ripple free PWM

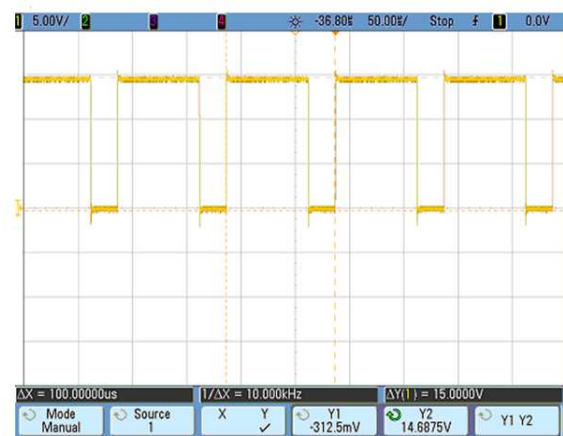


Fig. 10: output of Buffer circuit

RESULTS AND DISCUSSION

Experimental: To test the entire setup newly configured multicore processor is generated in the actel fusion device by using High Performance Reprogrammable Flash Technology. Next a control circuit for Permanent Magnet Motor is completed by using the embedded system based on FPGA and Cortex-M1 processor:

$$\text{Speed} = \text{RPM} = 60/\text{count} \times T_c$$

Where:

T_c = Counter time interval

Various output wave forms obtained from the different hardware circuits out put terminal is shown here Fig. 9 shows the output of optoIsolator given as an input to buffer circuit inside the driver circuit. Figure 10 shows the output of buffer circuit Fig. 11 shows the output of Driver circuit which is given as a input to IGBT based chopper circuit to produce corresponding ON- OFF in IGBT in order to control the motor speed.

The motor output is connected to MoC sensor. This Change in the logic level can be sensed by the FPGA based microcontroller and converted in to digital values by using ADC core in FPGA and transmitted and display in the OLED display unit which is available in the Cortex-M1 processor.



Fig. 11: output of driver circuit

CONCLUSION

In this study a Multicore based motor controller embedded system for permanent magnet motor based on Cortex-M1 core and FPGA core has been completed. The advantages and simplicity of FPGA and Reprogrammable capability of actel fusion device and easiest configurability of cortex family multicore performance can be used to design such a simple and portable multicore based embedded system to control the motor speed. This system has very flexibility and good real time control ability. In this controller Speed, memory can be optimized by means of using effective register allocation and simplified code execution. Actel fusion device has many of its individual features. So controller designed using this system can be used to gain high performance control of motor with fast execution and minimized size.

REFERENCES

- Al-Ayasrah, O., T. Alukaidey and G. Pissanidis, Hatfield, 2006. DSP based n-motor speed control of brushless DC motors using external FPGA design. Proceedings of the IEEE International Conference on Industrial Technology, Dec. 15-17, IEEE Xplore Press, Mumbai, pp: 627-631. DOI: 10.1109/ICIT.2006.372223
- Idkhajine, L., E. Monmasson, M.W. Naouar, A. Prata and K. Bouallaga, 2009. Fully integrated FPGA-Based controller for synchronous motor drive. IEEE Trans. Indus. Elect., 56: 4006-4017. DOI: 10.1109/TIE.2009.2021591
- Lin, F.J., J.C. Hwang, P.H. Chou and Y.C. Hung, 2010. FPGA-Based intelligent-complementary sliding-mode control for pmlsm servo-drive system. IEEE Trans. Power Elect., 25: 2573-2587. DOI: 10.1109/TPEL.2010.2050907
- Wang, D., K. Yu and H. Guo, 2008. Functional design of FPGA in a brushless DC motor system based on FPGA and DSP. Proceedings of the IEEE Vehicle Power and Propulsion Conference, Sept. 3-5, IEEE Xplore Press, Harbin, pp 1-4. DOI: 10.1109/VPPC.2008.4677538