TJES

ISSN: 1813-162X
Tikrit Journal of Engineering Sciences
available online at: http://www.tj-es.com

Proportional-Integral -Derivative Controller Using Embedded System Design Techniques

Mazin Rejab Khalil, Saja Basil Mahmood

Department of Electrical Engineering, Technical College, Mosul, Nenawa Province, Republic of Iraq

(Received 12 March2012, Accepted 10 August 2012, Available online 18 September2013)

ABSTRACT

This paper aims to design a proportional-integral-derivative(PID)controller to be configured on field programmable gate arrays(FPGAs) using Embedded Design Techniques(EDTs). Soft-core processor system is built using Integrated Software Environment (ISE 10.1)supplied by Xilinx and configured on Spartan 3E slice. The system is programmed by C language to act as PID controller .Trapezoidal rule is used for integration and finite difference method is used for derivative calculation .The designed controller is connected to a Matlab /Simulink of a DC motor model to check its functionality.

Keywords: Soft-core processor, FPGA, PID controller, Matlab / Simulink , Spartan 3E

تصميم مسيطر تناسبي تكاملي تفاضلي باستخدام تقنيات الأنظمة المطمورة

الخلاصة

يهدف هذا البحث إلى تصميم مسيطر تناسبي تكاملي تفاضلي وتشكيله على مصفوفة البوابات المبرمجة حقليا باستخدام تقنيات الأنظمة المطمورة. تم بناء نظام المعالج ذو النواة القابلة للبرمجة باستخدام بيئة البرمجيات المتكاملة (ISE)المدعمة من قبل شركة Xilinx وتنفيذها على شريحة Spartan 3E. تم برمجة هذا النظام بلغة C ليعمل كمسيطر تناسبي تكاملي تفاضلي ،استخدمنا قاعدة شبة المنحرف لأجراء التكامل وطريق الفرق المحدد لحساب التفاضل . تم ربط المسيطر المصمم إلى نموذج محرك للتيار المستمر في نظام المحاكاة في (Matlab) للتحقق من وظائفه .

الكلمات الدالة : المعالج ذو النواة القابلة للبرمجة، مصفوفة البوابات القابلة للبرمجة ،المسيطر التناسبي التكاملي التفاضلي، نظام المحاكاة في الماتلاب، بطاقة(*Spartan 3E*).

* Corresponding author: E-mail address: saja.basil@yahoo.com

List of abbreviations

R : Resistance of the stator
L : Inductance of the stator
Km: Motor of the torque constant
Kf : Viscous coefficient
Kb : Back electromotive force constant
J : Moment of the inertia
N : No of samples
H : Step size
e(n) : Samples of error signals
F (n) : Output of the integration part
d(n) : Output of the derivative part

Introduction

The Proportional-Integral-Derivative (PID) controllers have been widely used over the past five decades due to their simplicity ,robustness, effectiveness and applicability for a broad class of systems. Despite the numerous control design approaches that have been used, it is estimated that, nowadays PID controllers are still employed widely in industrial controller systems [1].

Field programmable gate arrays (FPGAs) represent a suitable media for designing PID controller due to small size, light weight and adaptability, reliability and performance, long-term maintenance [2].

EDTs helps the designer to construct a processor system with peripherals on FPGAs .The processor system can then be programmable to act as what the designer aims, in this paper the processor system is programmed to act as PID controller. The soft-core processor in this work used is MicroBlaze type.

Md. Shabiul Islam and Nowshad Amin in 2008[3] designed Fuzzy algorithm based PID controller using VHDL for transportation cruising system as a controlled.

loan Lita et al in 2009 [4] presented an approach for designing PID controller using field programmable analog array (FPAA) with application in temperature control in nuclear domain.

Vikas Gupta et al in 2011 [5] have designed PID controller using MATLAB and Simulink to generate a set of coefficients associated with the desired controller characteristics that are used to design PID controller on the FPGA.

The target of the paper is to design a PID controller to be implemented on FPGA, using EDTs and configured on Spartan 3E slice. The designed controller is attached to a Matlab/simulink model of a DC motor for the purpose of testing its performance.

Embedded Design Techniques

Fig.(1) shows the Basic embedded design process flow. It is composed of hardware development part and software development part. The two parts are transformed to a bit system saved in a file called a system.bit. Then they may be initialized together to be configured in the FPGAs[6] .

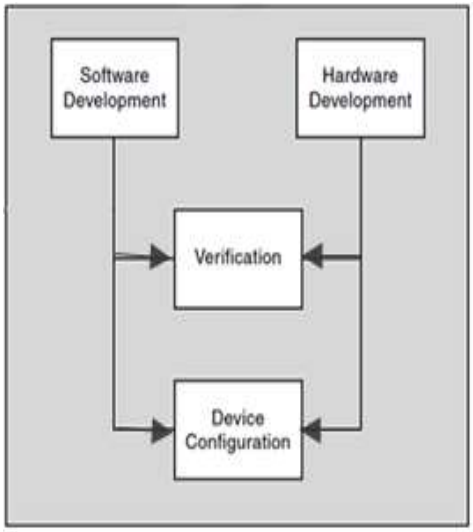

Fig. 1. Basic embedded design process flow

Hardware Part

The hardware part of the designed embedded system is shown in Fig.(2).The hardware part of the system is described by microprocessor hardware specification file MHS while the software part is described by the microprocessor software specification file MSS.

The hardware part is composed of:

●The MicroBlaze embedded soft core processor is a reduced instruction set computer (RISC) optimized for implementation Xilinx Field Programmable Gate Arrays(FPGA) [7].

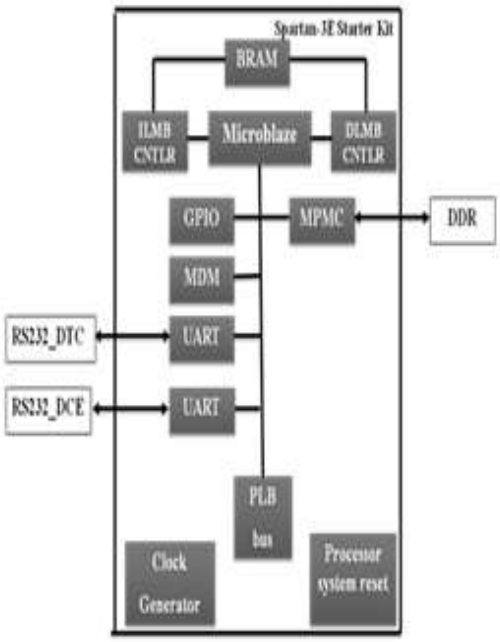

Fig.2. The block diagram of the system to be designed

●Processor Local Bus (PLB) v4.6 provides bus infrastructure for connecting an optional number of PLB masters and slaves into an overall PLB system [8] .

● Multi-Port Memory Controller MPMC is a fully parameterizable memory controller that supports SDRAM/DDR/DDR2 memory [9] .

● DDR-SDRAM with 64 Mbytes is used for program execution and is it is accessed by MicroBlaze using Multi-Port Memory Controller (MPMC) [9].

●BRAM Block is a configurable memory module attaches to a variety of BRAM Interface Controllers [10].

●LMB BRAM Interface Controller is the interface between the LMB and the bram_block peripheral. A BRAM memory subsystem consists of the controller along with the bram_block peripheral[11].

●Universal Asynchronous Receiver Transmitter (UART) Lite Interface connects to the PLB (Processor Local Bus) and provides the controller interface for asynchronous serial data transfer. This soft IP core is designed to interface with the PLBV46[12] .

The block diagram of the implemented design is shown in Fig.(3).

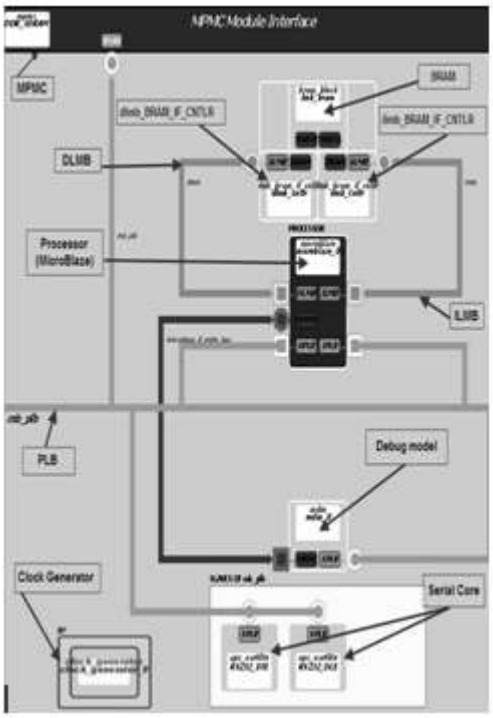

Fig.3. The resultant block diagram of hardware part the embedded system issued by the platform studio

The addresses map of the peripherals is shown in Fig.(4).

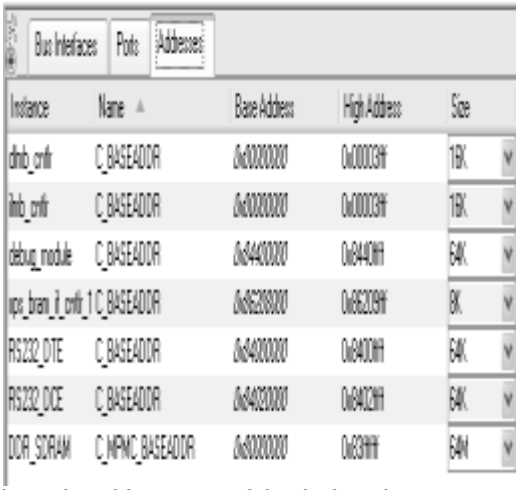

Fig.4. The address map of the designed system components

Software Part

Now that the hardware design is completed, the next step is to define the software part. Which is composed of two main

portion, software part configuring the Board Support Package (BSP) and writing the software applications. The configuration of the BSP includes the selection of device drivers and libraries. The Board Support Package (BSP) is a collection of files that defines the hardware elements of system for each processor. The BSP contains the various embedded software elements, such as software driver files, selected libraries, standard I/O devices, interrupt handler routines, and other related features[6].

Software applications is the code that runs on the hardware and software platforms. The source code for the application is written in a high level language such as C or C++, or in assembly language. The written C program will be compiled with the GNU Compiler tool. The compiled software routines are available as an Executable and Linkable Format (ELF) file. The ELF file is the binary ones and zeros that are run on the processor hardware .

Designed Embedded Processor System

Fig.(5) displays the philosophy of embedded design techniques that is used to construct the processor system[13], the hardware part and the software part. The hardware part described in MHS file is transferred to a bit file using ISE design flow(design entry, synthesis, implementation, verification, device programming).The software part described in MSS file is transferred into ELF file through software generation steps .Both bit file and ELF file are transferred into a system bit file through Data2MEM stage, there the bit file can be configured on FPGA through JTAG.

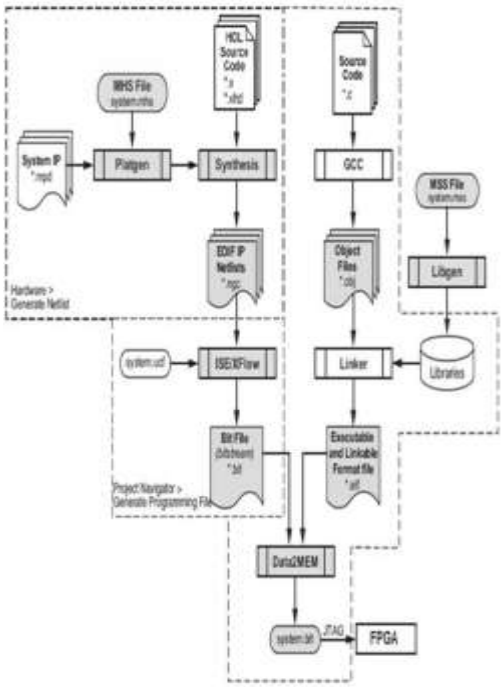

Fig.5. Embedded design philosophy[13]

PID Controller System

Fig. (6) shows the block diagram of the design system when the PID controller is implemented on the FPGA and the controlled system is simulated by the Matlab environment. The simulated controlled system is represented by a DC motor system.

Where

r = the set-point also called the reference or desired point

y = the actual output of the plant (also called the measured output)

e = the error between the actual value and the desired value

u = the control signal

Kp = proportional gain

- Multi-Port Memory Controller MPMC is a fully parameterizable memory controller that supports SDRAM/DDR/DDR2 memory [9] .
- DDR-SDRAM with 64 Mbytes is used for program execution and is it is accessed by MicroBlaze using Multi-Port Memory Controller (MPMC) [9].

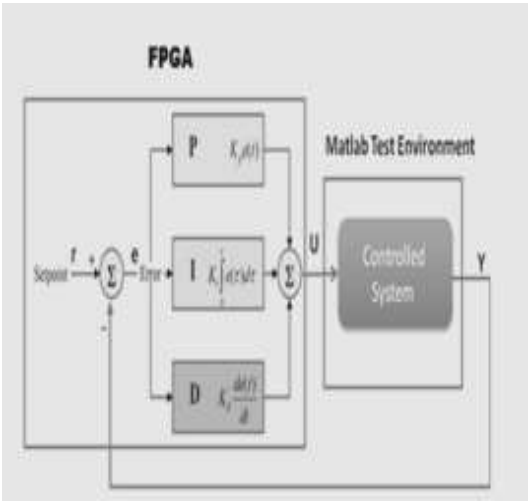

Fig.6.Block diagram of the designed PID acting as a controller to a sample controlled system modelled by Simulink

●BRAM Block is a configurable memory module attaches to a variety of BRAM Interface Controllers [10] .

●LMB BRAM Interface Controller is the interface between the LMB and the bram_block peripheral. A BRAM memory subsystem consists of the controller along with the bram_block peripheral[11] .

●Universal Asynchronous Receiver Transmitter (UART) Lite Interface connects to the PLB (Processor Local Bus) and provides the controller interface for asynchronous serial data transfer. This soft IP core is designed to interface with the PLBV46[12] .

The block diagram of the implemented design is shown in Fig.(7).

The addresses map of the peripherals is shown in Fig.(8).

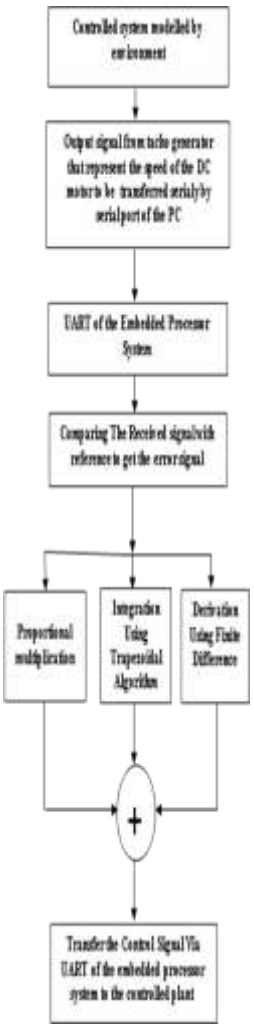

Fig.7.Flow chart of the application program

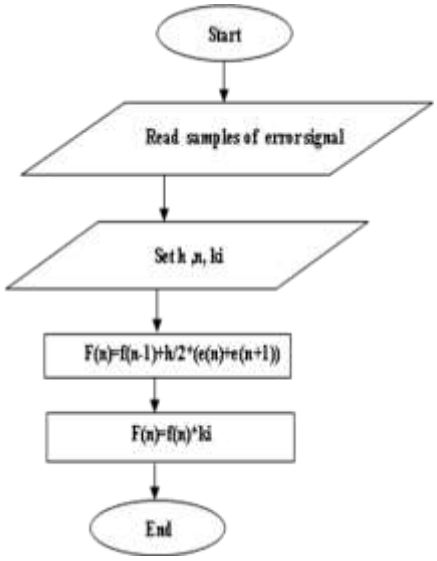

Fig.8. Flow chart of the integration part

Software Part

Now that the hardware design is completed, the next step is to define the software part. Which is composed of two main portion, software part configuring the Board Support Package (BSP) and writing the software applications. The configuration of the BSP includes the selection of device drivers and libraries. The Board Support Package (BSP) is a collection of files that defines the hardware elements of system for each processor. The BSP contains the various embedded software elements, such as software driver files, selected libraries, standard I/O devices, interrupt handler outlines, and other related features [6] .

finite difference is used for derivation, Fig.(9) shows the flow chart of the derivation scheme.

After then the control signal data are sent to Matlab via UARTLite peripheral using the software driver XUartLite_SendByte() functions[15].

Proportional part

proportional part is represented multiplying the error signal by the proportional gain.

Integration Part

Trapezoidal Rule is adopted to integrate the error signal which in the form of (400)samples, Fig.(8) shows the flow chart of the integration part.

Derivative Part

To calculate the derivative of the error control signal, Fig.(9) shown the flow chart of the algorithm is used.

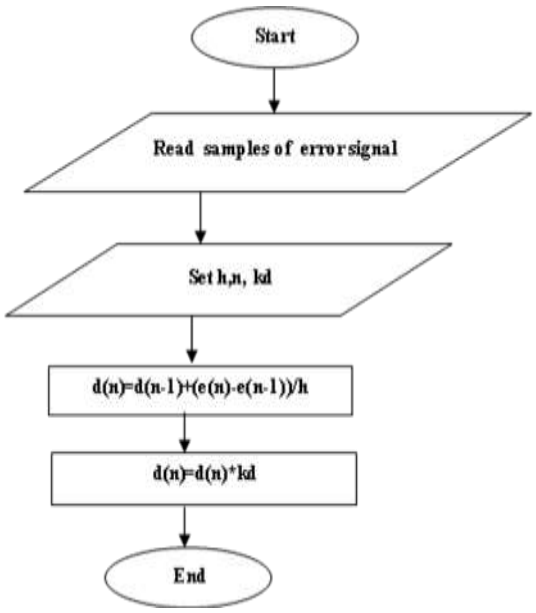

Fig.9. Flow chart of the derivative

Results and Discussions

In order to verify the performance of the designed PID controller. A control system simulated on Matlab as shown in Fig. (9) has been used .

The PID controller is implemented within the simulator media. For the purpose of comparing its performance with that configured on FPGAs using EDTs. The results are as following:

1.Figs (10,11,12) display the output samples of the PID controller implemented in the FPGA using EDTs.

-Fig.(10) represents the output samples displayed on the chipscope analyzer that used to monitor samples transferred via uart.

-Fig.(11) represents the PID output samples exhibited on the hyper terminal that corresponds to DTE terminal.

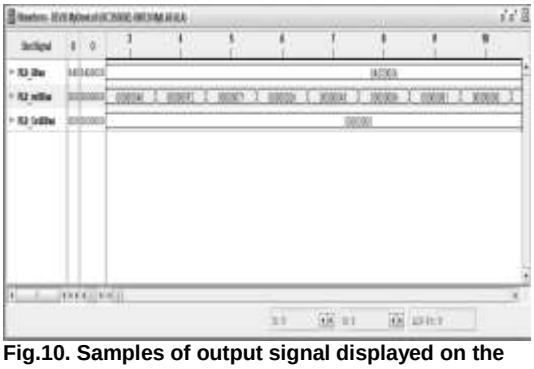

Fig.10. Samples of output signal displayed on the chipscope for monitoring signal on PLB

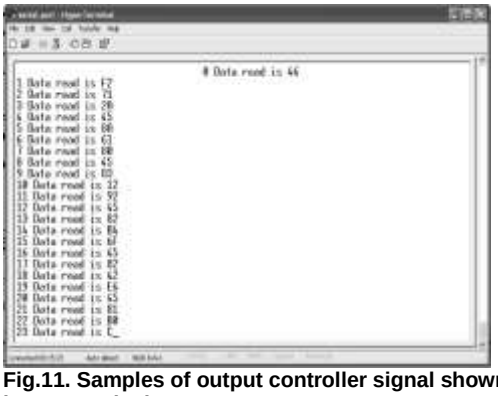

Fig.11. Samples of output controller signal shown on the hyper terminal

It is noted that the values of the output signal are

the same for chipscope analyzer and the hyper terminal. there samples represents the error signal after being processed by the PID controller.

-Fig.(12) displays the output signal of the PID controller after transferring the output samples from FPGAs media to Matlab media. The signal is noticed to be stable after 1st 100 samples.

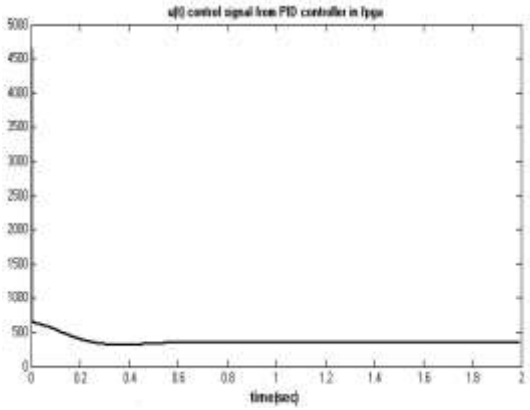

Fig.12. Received control signal from embedded system(fpga) to Matlab

2-Fig.(13) displays output signal of the controlled plant (y) .the system is stable within 0.5 sec .

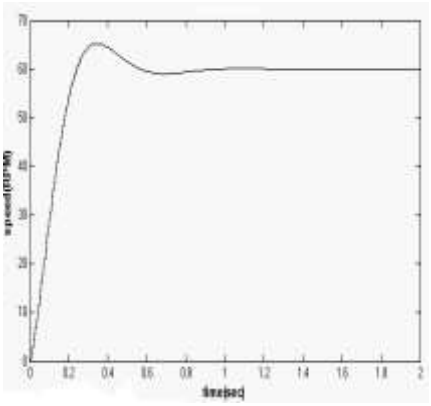

Fig.13. Output signal of the controlled plant

3-The obtained results from PID controller implemented in Matlab /Simulink media is shown in Fig.(14) .It is compliant with that shown in Fig.(12).

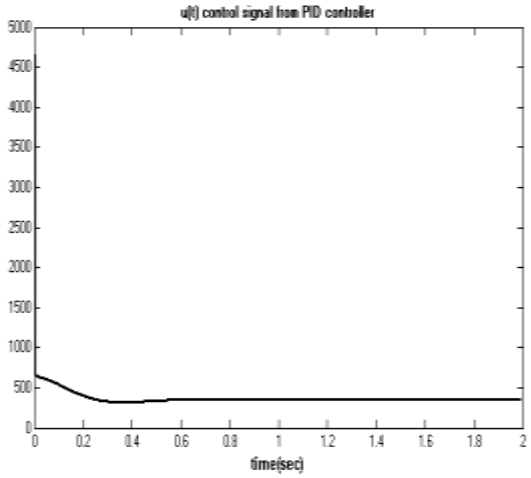

Fig.14. Output of the PID controller with in Matlab media

Table 1.Paramters of DC motor

Parameter	Values and units
R	2 Ω
Kb	0.015 V/rpm
Km	0.015N.m/A
L	0.5 H
Kf	0.2 Nms
J	0.02 kg.m^2

Conclusions

A PID controller is designed using EDTs and configured on the Spartan3E FPGAs to be applied as a controller in a plant of a DC motor controlled system. The design when compared with other techniques achieved a minimization in the hardware and reduction in cost and light weight, flexibility ,reliability. The controlled plant exhibited a realizable performance with acceptable stability when the designed PID controller was applied.

References

1- K.J. Astrom and T. H. Hagglund, New Tuning Methods for PID Controllers, Proc. of 3rd European Conference , pp. 2456-2462, 1995.

2- Scott Hauck and Andre DeHon, Reconfigurable Computing The Theory And Practice Of FPGA-Based Computation", Elsevier Inc, 2008.

3- Md.Shabiul Islam ,Nowshad Amin , Mukter Zaman , M.S.Bhuyan, Fuzzy based PID

Controller using VHDL for Transportation Application, Issue 2, Volume 2, 008v.

4- Loan Lita, Daniel Alexandru Visan, Ion Bogdan Cioe, FPAA Based PID Controller with Applications in the Nuclear Domain, 32nd International Spring Seminar on Electronics Technology, pp. 1-4, 2009.

5- Vikas Gupta and K. Khare and R. P. Singh , Efficient FPGA Implementation of 2nd order Digital Controllers using MATLAB /SIMULINK, VO L. 6, NO. 8, AUG UST , 2011.

6- Xilinx Company, "Embedded System Tools Reference Manual", UG111 April 19, 2010 .

7- Xilinx Company, MicroBlaze Processor Reference Guide, UG081 (v9.0) . http://www.xilinx.com/support/documentation/sw_manuals/mb_ref_guide.pdf.

8- Xilinx Company , Processor Local Bus (PLB) v4.6, data sheet DS531, April 24, 2009. http://www.xilinx.com/support/documentation/ip_documentation/ds531.pdf

9- Xilinx Company, Multi-Port Memory Controller (MPMC) (v6.00.a), data sheet DS643, April 19, 2010. http://www.xilinx.com/support/documentation/ip_documentation/mpmc.pdf

10- Xilinx Company , Block RAM (BRAM) Block (v1.00a) ",data sheet DS444 , March 12, 2007.http://www.xilinx.com/support/documentation/ip_documentation/bram_block.pdf.

11- Xilinx Company, LMB BRAM Interface Controller (v2.10b), data sheet DS452, March 2, 2010.

12- Xilinx Company , XPS UART Lite (v1.01a), data sheet DS643, April 19,2010.http://www.xilinx.com/support/documentation/ip_documentation/xps_uartlite.pdf

13- Xilinx Company , EDK Concepts, Tools, and Techniques, XTP013 EDK 10.1.

14- Kilian, Modern Control Technology Components and Systems, 2nd edition, Delmar Thomson Learning, ISBN-10: 076682358, December 2000.

15- Xilinx Company, Xilinx Device Drivers, Documentation, Jun 24 , 2004.