

FPGA implementation of high speed PI like Fuzzy control system for industrial automation applications

G. Kamalesh¹ T. Thammi Reddy²

¹ECE Department, GPR College of Engineering, Kurnool, India

²ECE Department, Associate Professor of GPR College of Engineering, Kurnool, India

Abstract: The modern digital control systems demand faster and robust calculation components for robotic and industrial automation applications. This type of elements are becoming more essential with the utilization of some new control algorithms, like the fuzzy control, the adaptive control, the sliding mode control, etc. The PID controllers are most widely used controllers in the industrial control systems. Fuzzy logic control presents a computationally efficient and robust alternative to conventional controllers for many systems. Although the traditional control methods which use mathematical models of systems to design an adequate controller, the fuzzy logic controllers use fuzzy deductions or inferences for the synthesis of controllers are powerful and robust.

Digital controllers are implemented two styles; hardware based and software based. The software based implementation can be carried out on PC or any DSP processor. Such implementations will be inherently slower due to serial nature of the processor's execution style. The FPGA platform carrying advantages of both ASIC and processor is more suitable option. On FPGA one can easily achieve higher speeds occupying only less area. In this project PI like fuzzy logic controller (PIFLC) will be implemented in VHDL for FPGA platform. This is a general purpose controller that can be used for different applications. This controller has three stages: the fuzzification, the inference and the defuzzification. The first component in the PIFLC is the fuzzifier that transforms crisp inputs into a set of membership values in the interval [0, 1] in the corresponding fuzzy sets. The knowledge base of the fuzzy controller consists of a database of linguistic statements (rules), which states the relationship between the input domain fuzzy sets and output domain fuzzy sets. Inference block implements this logic. The last step is the defuzzification and the final output is determined by weighted average of all contributions of the output sets.

Keyword: PI Like Fuzzy Control System, Industrial Application, FPGA, Fuzzy logic.

I. Introduction

Fuzzy Logic is a problem-solving control system methodology that lends itself to implementation in systems ranging from simple, small, embedded micro-controllers to large, networked, multi-channel PC or workstation-based data acquisition and control systems. It can be implemented in hardware, software, or a combination of both. Fuzzy Logic provides a simple way to arrive at a definite conclusion based upon ambiguous, imprecise, noisy, or missing input information. Fuzzy Logic's approach to control problems mimics how a person would make decisions, only much faster. Fuzzy Logic incorporates a simple, rule-based IF X AND Y THEN Z approach to a solving control problem rather than attempting to model a system mathematically. The Fuzzy Logic model is empirically-based, relying on an operator's experience rather than their technical understanding of the system. For example consider what you do in the shower if the temperature is too cold: you will make the water comfortable very quickly with little trouble. Fuzzy Logic is capable of mimicking this type of behavior but at very high rate. Fuzzy Logic requires some numerical parameters in order to operate such as what is considered significant error and significant rate-of-change-of-error, but exact values of these numbers are usually not critical unless very responsive performance is required in which case empirical tuning would determine them. For example, a simple temperature control system could use a single temperature feedback sensor whose data is subtracted from the command signal to compute "error" and then time-differentiated to yield the error slope or rate-of-change-of-error, hereafter called "error-dot". Error might have units of degs F and a small error considered to be 2F while a large error is 5F. The "error-dot" might then have units of degs/min with a small error-dot being 5F/min and a large one being 15F/min. These values don't have to be symmetrical and can be "tweaked" once the system is operating in order to optimize performance. Generally, Fuzzy Logic is so forgiving that the system will probably work the first time without any tweaking.

Fuzzy Logic Used:

- Define the control objectives and criteria: What am I trying to control? What do I have to do to control the system? What kind of response do I need? What are the possible (probable) system failure modes?
- Determine the input and output relationships and choose a minimum number of variables for input to the Fuzzy Logic engine (typically error and rate-of-change-of-error).
- Using the rule-based structure of Fuzzy Logic, break the control problem down into a series of IF X AND Y THEN Z rules that define the desired system output response for given system input conditions. The number and complexity of rules depends on the number of input parameters that are to be processed and the number fuzzy variables associated with each parameter. If possible, use at least one variable and its time derivative. Although it is possible to use a single, instantaneous error parameter without knowing its rate of change, this cripples the system's ability to minimize overshoot for a step inputs.

- Create Fuzzy Logic membership functions that define the meaning (values) of Input/Output terms used in the rules.
- Create the necessary pre- and post-processing Fuzzy Logic routines if implementing in S/W, otherwise program the rules into the Fuzzy Logic H/W engine.
- Test the system, evaluate the results, tune the rules and membership functions, and retest until satisfactory results are obtained.

II. Block Diagram Of Pi-Like Fuzzy Controller Modification

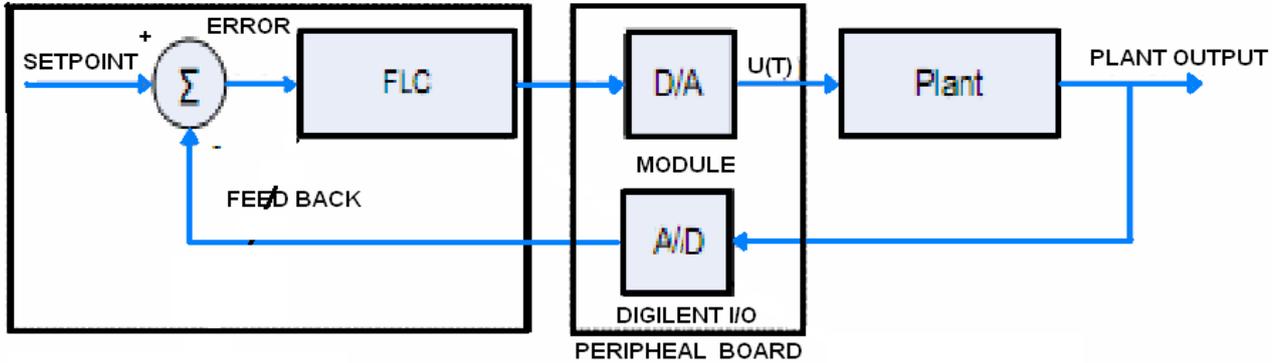
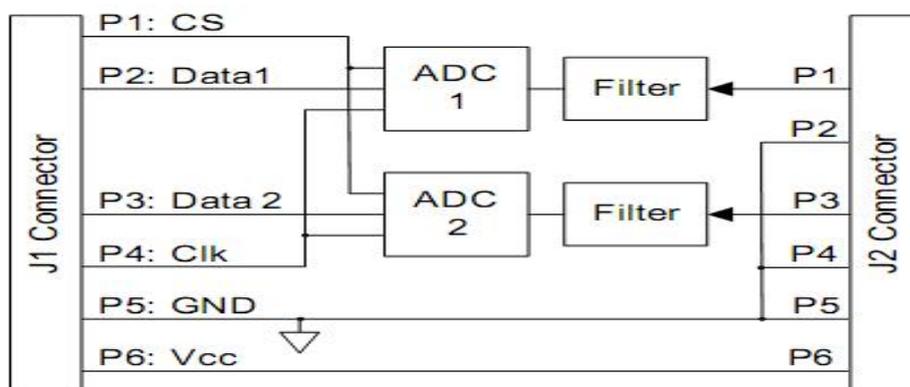
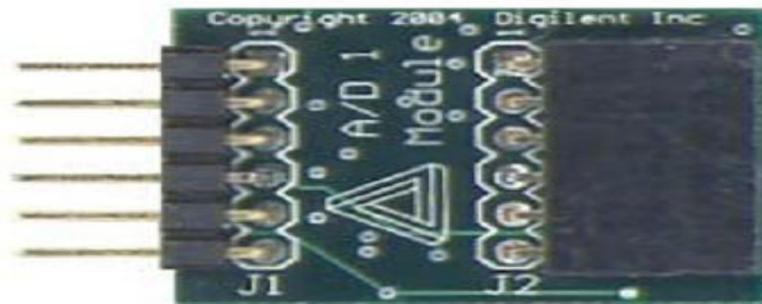


Fig .1Layout of the proposed controller in a unity feedback control system

Pi-Like Fuzzy Controller Modification is the general layout of the controller chip in a unity feedback control system is shown in Fig.1



2.1analog Input Interface (Ade):

FPGAs are well suited for serial Analog to Digital (A/D) converters. This is mainly because serial interface consumes less communication lines while the FPGA is fast enough to accommodate the high speed serial data. The ADCS7476MSPS is a high speed, low power, 12-bit A/D converter. Consumes 80 ns time for one cycle. A/D converter is a high speed serial interface that interfaces easily to FPGAs. The A/D interface adapter (AD1_PMOD) is implemented within the FPGA. (Fig.2). Inside the FPGA, this adapter facilitates parallel data acquisition. Sampling is initiated at the rising edge of a clock applied at the line sample. The timing diagram of the communication protocol obtained with Modelsim is illustrated in Fig. The whole conversion and acquisition period is 1.2 μ s allowing sampling up to a rate of 20 MHz Sample per second.

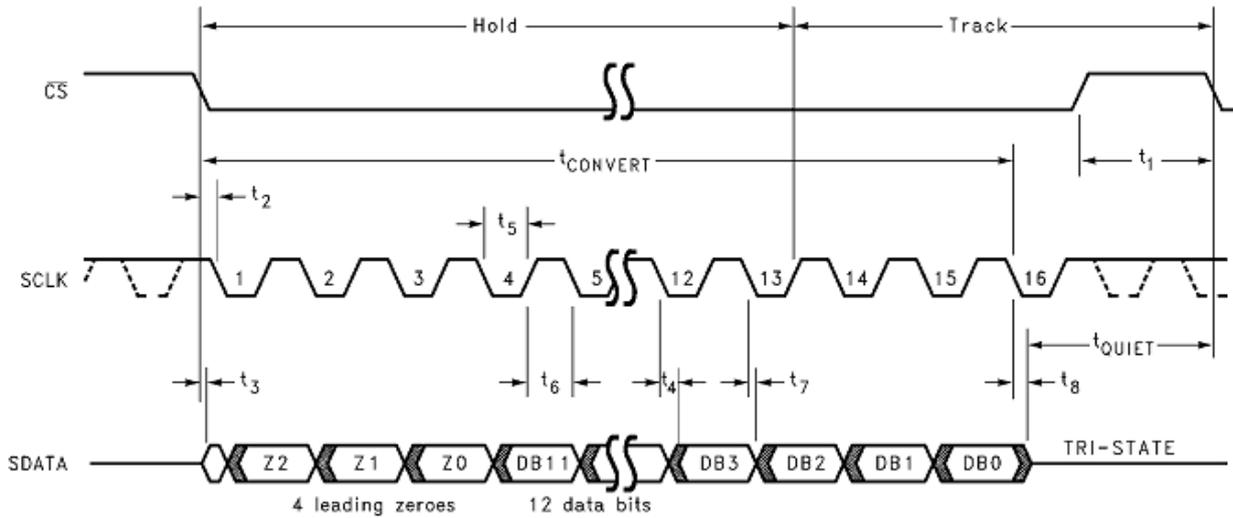


Fig 2. ADCS7476 Serial Interface Timing Diagram Timing diagram

2.2analog Output Interface (Dac):

The DAC121S101 is a dual, 12-bit voltage out Digital to Analog (D/A) converter. This device uses a versatile 6-wire serial interface that operates at a clock up to 20 MHz. The serial input register is 16 bits wide; 12 bits act as data bits for the D/A converter. It is interfaced to an FPGA as illustrated in Fig. 3. The D/A interface adapter (DAC_toplevel), which is implemented within the FPGA, facilitates parallel data input for the dual D/A converters. The timing diagram of the communication protocol is illustrated in Fig. 5. The transmission period of a sample is 80 ns allowing D/A conversion at an excellent rate of 12.5MHz.

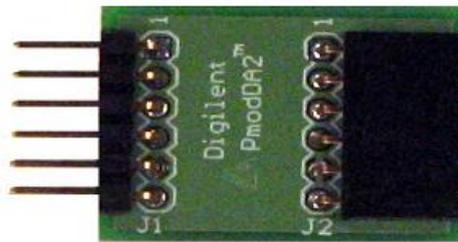
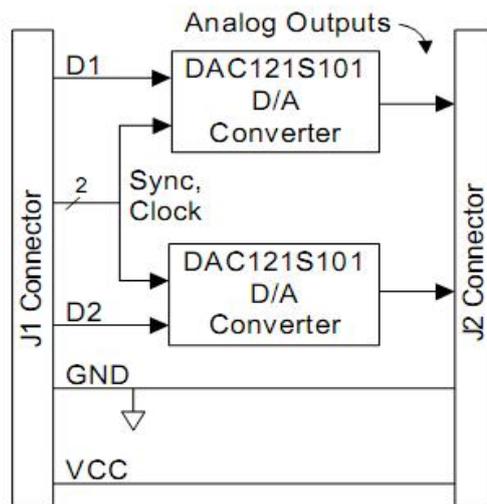


Figure 1
Digilent PmodDA2



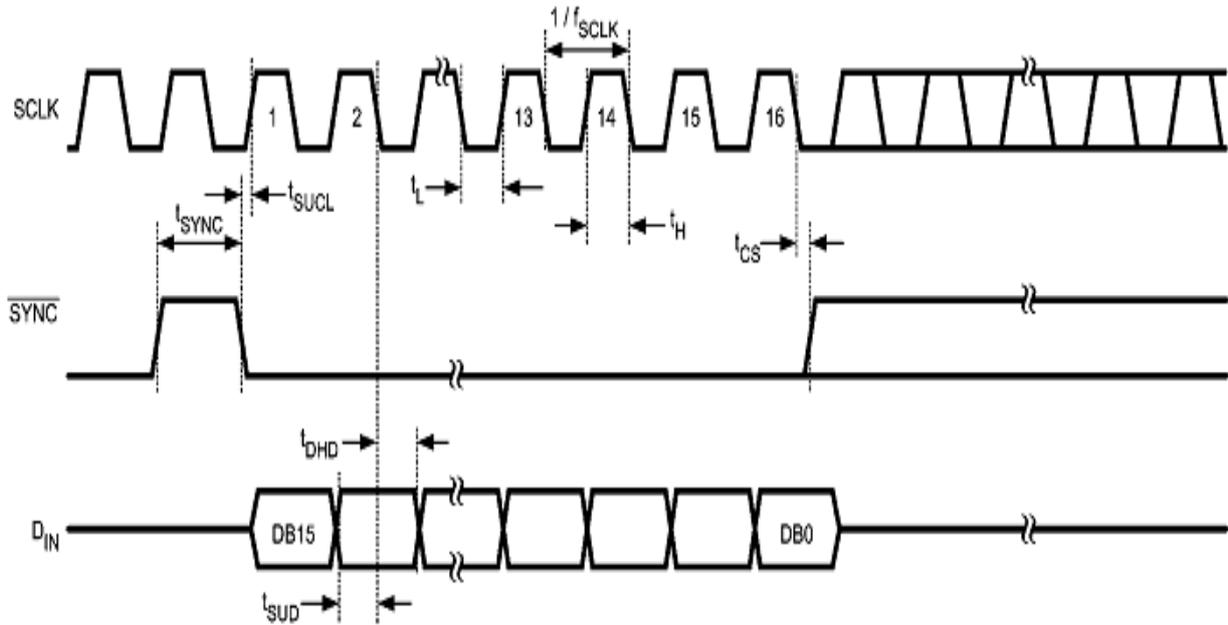


Fig.3. DAC121S101 Timing

2.3fuzzy Controller:

The PI-like Fuzzy Logic Controller (PIFLC) has two inputs the error (e) and its change (Δe) and one output the change of control (Δu). This controller follows in its logic the three stages:

- 1) fuzzification,
- 2) inference and
- 3) defuzzification.

We do remark that scaling factors can be introduced for input variables and the output one denoted respectively k_e , k_{de} and k_{du} (Fig. 4).

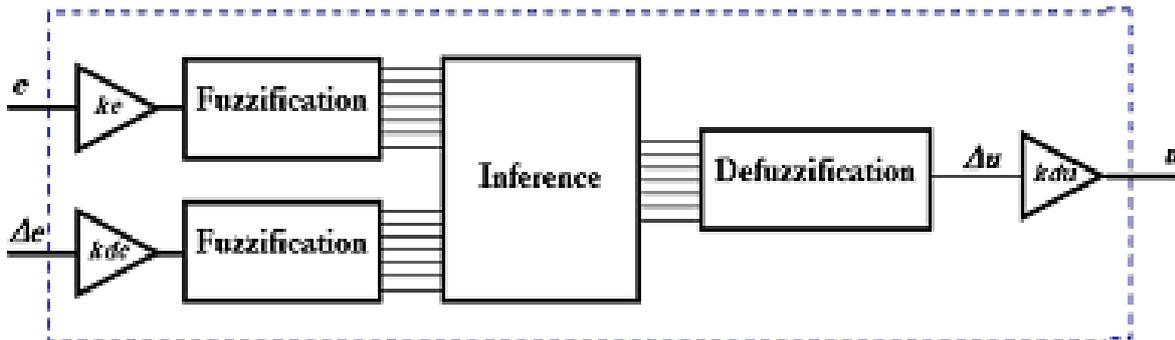


Fig.4.Block diagram of PIFLC

1.3.1fuzzification:

The first component in the PIFLC is the fuzzifier that transforms crisp inputs into a set of membership values in the interval $[0, 1]$ in the corresponding fuzzy sets. The membership function shapes are typically triangular, trapezoidal or exponential. In this, the membership functions are triangular-shaped and the maximum value is scaled to 40 instead of 1 which is found in other documents describing fuzzy theory. This way the calculation complexity is greatly reduced because the multiplying operation becomes only one addition or subtraction. With a setpoint and a measure coded on 8 bits, the intervals of the fuzzy sets are selected in order to cover all the range between -127 and 127. Fig. 7 shows triangular membership functions for error (e) and its change (Δe) with fuzzy labels NB (negative big), NM (negative medium), NS (negative small), ZO (zero), PS (positive small), PM (positive medium) and PB (positive big). For the output, the membership functions are given in Fig. 8 and correspond to singletons in order to simplify the defuzzification process.

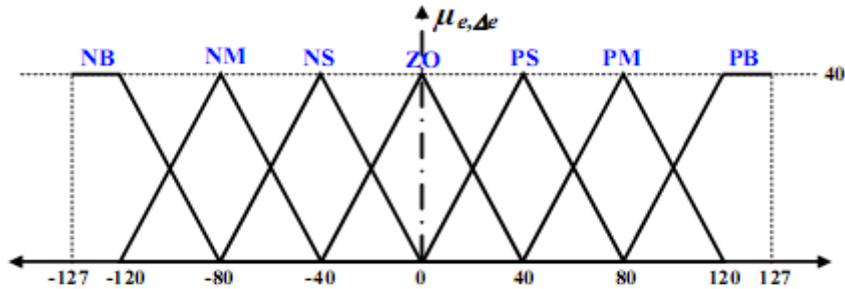


Fig.5. Triangular membership functions of inputs

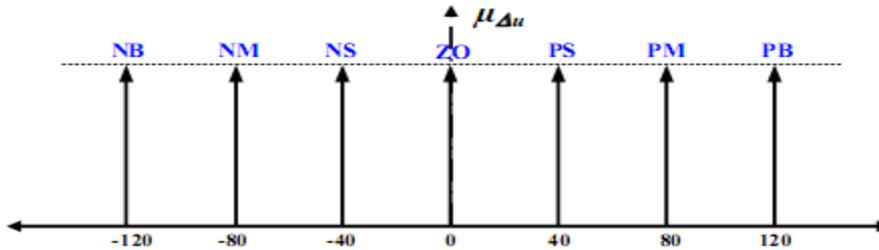


Fig. 6. Singleton membership functions of output

The representation of this is

- 127 to -120 is represented as (NB) i.e 000
- 120 to -80 is represented as (NM) i.e 001
- 80 to -40 is represented as (NS) i.e 010
- 40 to 40 is represented as ZERO i.e 011
- 40 to 80 is represented as (PS) i.e 101
- 80 to 120 is represented as (PM) i.e 110
- 120 to 127 is represented as (PB) i.e 111

2.3.2 Inference:

The knowledge base of the fuzzy controller consists of a database of linguistics statements (rules), which states the relationship between the input domain fuzzy sets and output domain fuzzy sets. For a system with two inputs, the error (e) and change of error (Δe), and single output, each having seven fuzzy sets, the rules can be represented in tabular form as shown in Table I. A maximum of 4 rules can be active at any time with triangular membership functions. The min-max inference method uses the min operator to find the minimum membership degree between the two inputs resulting from rule conditions and the rules are finally combined by using the OR operator and interpreted as the max operation for each possible value of the output variable.

Δe	e	NB	NM	NS	Z	PS	PM	PB
NB		NB	NB	NB	NB	NM	NS	ZO
NM		NB	NB	NB	NM	NS	ZO	PS
NS		NB	NB	NM	NS	ZO	PS	PM
Z		NB	NM	NS	ZO	PS	PM	PB
PS		NM	NS	ZO	PS	PM	PB	PB
PM		NS	ZO	PS	PM	PB	PB	PB
PB		ZO	PS	PM	PB	PB	PB	PB

Table I. Mac Vicar-Whelan Rules Base

2.3.3 Defuzzification:

The last step is the defuzzification and the final output is determined by weighted average of all contributions of the output sets. It is obtained by finding the centroid point of the function which is the result of the multiplication of the output membership function and the Inference output vector Y. The general mathematical formula which is used to obtain the centroid point is:

$$\Delta u = \frac{\sum_{i=1}^5 Y[i] \times S[i]}{\sum_{i=1}^5 Y[i]}$$

Y (i) are the i-th members of the output vector, S (i) are the multiplying coefficients of the output membership function.

III. FPGA Implementation Of High Speed PI Like Fuzzy Control System Final Outputs

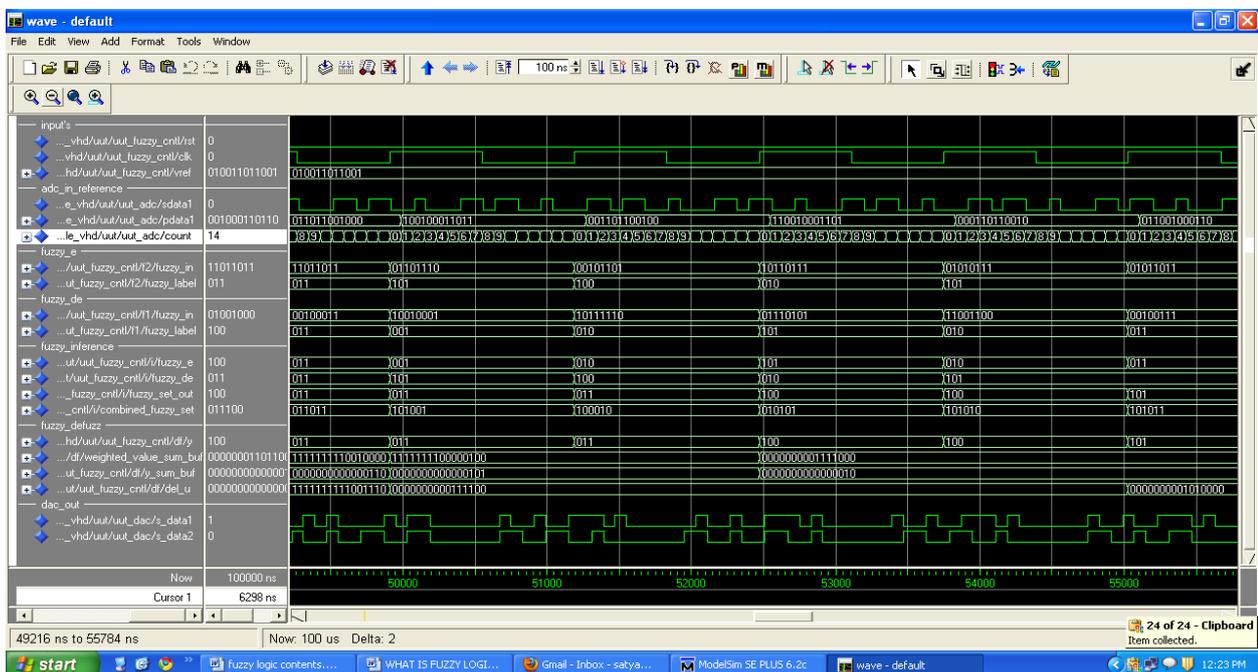
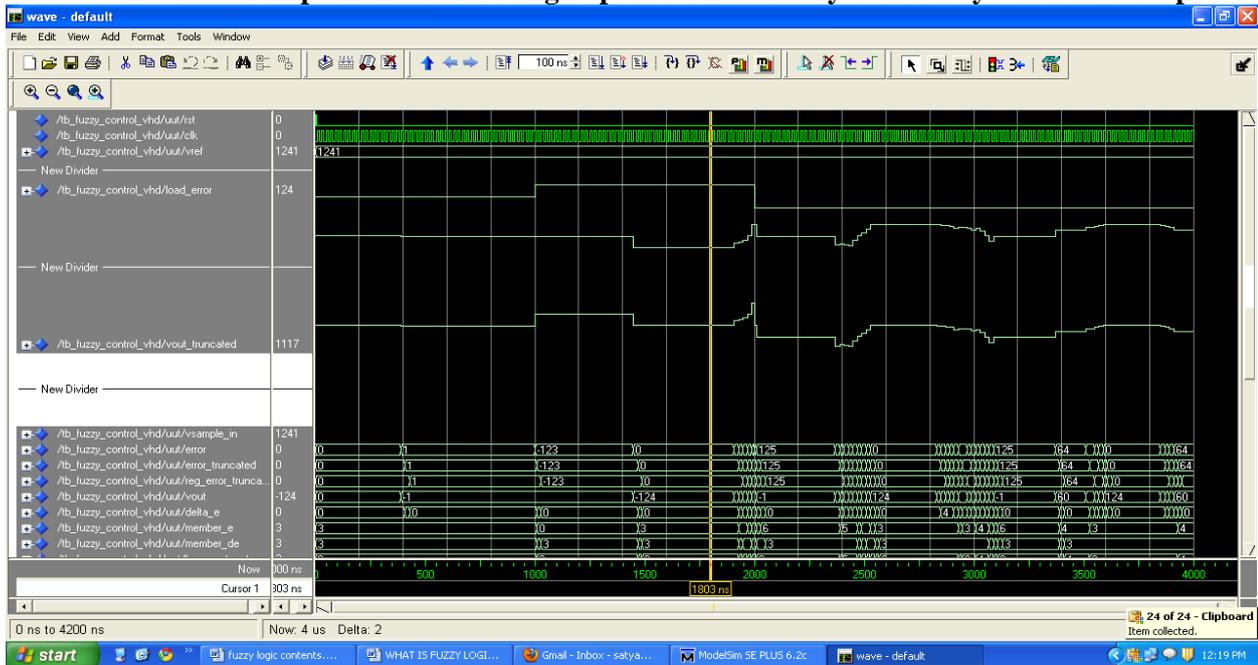


Fig.7. PI like Fuzzy control system Final output simulation wave forms

IV. Applications

Fuzzy logic is an innovative technology that enhances conventional system design with engineering expertise. The use of fuzzy logic can help to circumvent the need for rigorous mathematical modeling. Fuzzy logic is a true extension of conventional logic, and fuzzy logic controllers are a true extension of linear control models. Hence anything that was built using conventional design techniques can be built with fuzzy logic, and vice-versa. However, in a number of cases, conventional design methods would have been overly complex and, in many cases, might prove simpler, faster and more efficient. The key to successful use of fuzzy logic is clever combination with conventional techniques. Also, a fuzzy system is time-invariant and deterministic. Therefore any verification and stability analysis method can be used with fuzzy logic, too.

- Pattern recognition and classification
- Fuzzy clustering
- Image and speech processing
- Fuzzy systems for prediction
- Fuzzy control
- Monitoring
- Diagnosis
- Optimization and decision making

V. Advantages

One can circumvent the need for rigorous mathematical modeling with the use of fuzzy logic. Unlike the reasoning based on classical logic, fuzzy reasoning aims at the modeling of reasoning schemes based on uncertain or imprecise information. The past several years have witnessed a rapid growth in the number and variety of applications of fuzzy logic. The most visible applications are in the realms of consumer products, intelligent control and industrial systems. Less visible, but of growing importance, are applications relating to data processing, fault diagnosis, man-machine interfaces, quality control and decision support systems. Although fuzzy logic has been and still is controversial to some extent, its successes are now too obvious to be denied.

VI. Tools and Hard Ware

- Simulation software -Modelsim Xilinx Edition (MXE)
- Synthesis, P&R - Xilinx ISE
- On chip verification - Xilinx Chip scope
- Hardware– Xilinx Spartan 3 Family FPGA board

VII. Conclusion

This paper presents an approach for the implementation of a fuzzy logic controller on an FPGA using VHDL. A fuzzy logic controller with 2-inputs and 1-output is simulated and each block's verification is carried out using logic simulator. The FLC is implemented on a Xilinx Spartan-3 FPGA and used to control first order system to demonstrate its validity. The controller with the analog to digital and digital to analog interfaces was found to be fully functional. The FLC can also be used for control purposes in other applications. Also the length of input and output of the FLC can be increased to achieve better results.

References

- [1] Hassan M.Y. and Sharif W.F. "Design of FPGA based PID-like Fuzzy Controller for Industrial applications".
- [2] Maheshappa H. D., Samuel R. D., Prakashan A., "Digital PID Controller for Speed Control of DC Motors", IETETechnical Review Journal, V6, N3, PP171-176, India 1989.
- [3] Masmoudi N., Hachicha M. and Kamoun L., "Hardware Design of Programmable Fuzzy Controller on FPGA".IEEE International Fuzzy Systems Conference Proceedings, Seoul, Korea, 22-25 August, 1999.
- [4] Masmoudi N., Samet L., Kharrat M.W., Kamoun L. "Implementation of a PID Controller on FPGA", Forth ICECS'97, IEEE International Conference on Electronics, Circuits and Systems, Volume 2, pp 550-554, Caire Egypte15-18, December 1997.
- [5] Montiel O., Maldonado Y., Sepulveda R. and Castillo O., "Simple Tuned Fuzzy Controller Embedded into an FPGA". Fuzzy Information Processing Society, NAFIPS. Annual Meeting of the North American, New York, USA, Mai 19-22, 2008.
- [6] Singh S. and Rattan K. S., "Implementation of a Fuzzy Logic Controller on an FPGA using VHDL". Fuzzy Information Processing Society. 22nd International Conference of the North American NAFIPS, Chicago, Illinois, USA, July 24-26, 2003.
- [7] Tang J., "PID Controller using the TMS320C31 DSK with On-line Parameter Adjustment for Real-Time DC Motor Speed and Position Control", IEEE International Symposium on Industrial Electronics, V2, PP 86-791, Pusan 2001.