Manuscript received June 16, 2007; revised Sep. 17, 2007

FPGA Realization of Fuzzy Temperature Controller for Industrial Application

SHABIUL ISLAM[1], NOWSHAD AMIN[2], M.S.BHUYAN[1], MUKTER ZAMAN[1],
BAKRI MADON[3], MASURI OTHMAN[4]
[1]Faculty of Engineering, [2,4]Dept. of Electrical, Electronic and System Engineering,
[2]Solar Energy Research Institute, [3]Kriptic Devices Sdn Bhd,
[1]Multimedia University, [2,4]University Kebangsaan Malaysia,
[1]63100 Cyberjaya, [2,4]43600 Bangi, [3] Jalan 3/101C, Taman Cheras, 56100 Kuala Lumpur,
[1, 2, 3, 4] MALAYSIA
E-mail : shabiul@mmu.edu.my

Abstract: - This paper describes FPGA realization of a Fuzzy Temperature Controller (FTC) using VHDL intended for industrial application. The system is built up with four major modules namely fuzzification, inference, implication and defuzzification. The composition method selected for the fuzzy model is the Max-Min composition while the Mamdani Min operator is chosen as the implication method. Each module is modeled individually using behavioral VHDL, and the combined using structural VHDL. Successful timing and functional simulations are carried out to verify the correct functionality of the algorithm. The verified VHDL model is synthesized using synthesis tool to get gate-level architecture of the FTC chip. The designed synthesized netlist is downloaded into FPGA board from Altera for verifying the functionality of the FTC chip. The operating frequency of the FTC chip is 5MHz with a critical path of 199.3ns.

Key-Words: - VHDL, Temperature controller, Fuzzy, Synthesis, FPGA

1 Introduction

An effective and efficient controller for the surrounding environment is crucial in many technical processes. Ranging from IC fabrication to the production of chemical solutions, any changes in the ambient parameters can have a drastic effect in the outcome of a process, at the very least lowering the yield or quality of the product. Among the crucial parameters that merits close supervision is the temperature of the environment. As such, temperature controller is critical to the quality, appearance and consumer acceptance of a manufacturer's products.

The processes that requires temperature controller has various unfavorable characteristics including non-linearity, dead zone time, external disturbances and so on. Conventional approximations do not produce satisfactory temperature controls for controlling complex processes, which is usually the case in the industry because they suffer from various drawbacks such as slow stabilization, overshooting and overall slow response.

A fuzzy system improves the relative performance of a temperature control process with respect to the conventional scheme. It compensates non-linear errors, accelerates the response and reduces the steady-state error. The Fuzzy Logic Controller (FLC) is also able to bring the temperature constant at the desired value regardless of changes in the load or environment. This project attempts to enable a fuzzy-based control of the temperature employing VHDL as a mean of improving upon conventional methods.

Several works had been done in this area. Zhiqiang et al. [1] had developed a closed loop control system incorporating fuzzy logic for a class of industrial temperature control problems employing a unique FLC structure with an efficient realization and a small rule. Their works demonstrated in both software simulation and hardware test in an industrial setting that the fuzzy logic control is much more capable than the current temperature controllers. This includes compensating for thermo mass changes in the system, dealing with unknown and variable delays and operating at very different temperature set points without retuning. Thyagarajan *et al.* in [2] presented four control schemes designed using advanced techniques for regulating the temperature of the Air Heat Plant. The four control schemes are namely, PID, fuzzy logic control, FLC using genetic algorithms (FLC-GA) and Neuro-Fuzzy control (NFC). All these schemes are evaluated with respect to set-point tracking using performance indices. Their works highlighted superiority of FLC over PID, FLC-GA FLC and NFC schemes. Some more works utilizing fuzzy logic to control temperature for specific applications is discussed here [3]-[4].

In this paper, the control system is implemented using VHDL, aiming for FPGA implementation. There are several advantages for this approach. Firstly, FPGA implementation allows immediate manufacturing realization and negligible prototype costs. In addition, FPGA offer affordable and practical solutions to custom applications as well as allow new vista in designing reconfigurable digital systems. In testing, FPGA allow designers the freedom to redesign portions of their circuit for optimization, without performing full redesign iterations to improve a design [5].

One major benefit of hardware implementation over software is the simulation speed. Hardware-based simulation allows the simulation process to take advantage of the parallel execution of instructions. Other advantages of programmable hardware are the ability to perform bit-level operations on "unusual," i.e., not powers of two, word lengths and the possibility to allocate only a certain number of bits to represent internal variables. Hence, it can be seen that the FPGA combines the flexibility of software and the speed of hardware [6].

2 Development of the FTC Algorithm

This section covers the specifications of the fuzzy model of the temperature controller. The models of the controller based on fuzzy rules are known as, the Fuzzification module, Inference module, Implication module, and Defuzzification module. All the relevant and crucial parameters are explained and illustrated, including the set of fuzzy rules applicable. The fuzzy model has been coded in C++, also presented in following paragraphs.

Two inputs "Error" (an error signal), and "CError" (rate of change in error after a fixed period) are used in the model. Each input consists of 4 triangular membership functions over a normalised range from 0 to 1. Fig.1 and Fig. 2 illustrate the 4 fuzzy variables, for both inputs are termed ZE (Zero), PS (Positive Small), PM (Positive Medium) and PL (Positive Large).

Based on these 2 inputs, the fuzzy logic model determines the amplitude of the voltage signal that is necessary to be sent to the heater in order to maintain a constant temperature in the industrial process. This is provided by "Output" (output signal) from the model, with a normalised range of [0 1]. Similar to the inputs, the "Output" signal has 4 triangular membership functions spaced over this range shown in Fig. 3.

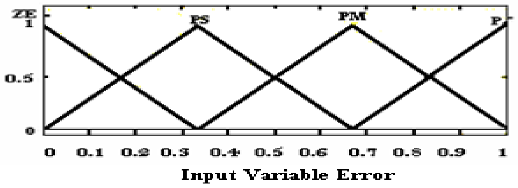


Fig.1 Membership Functions of Input Fuzzy Variable "Error"

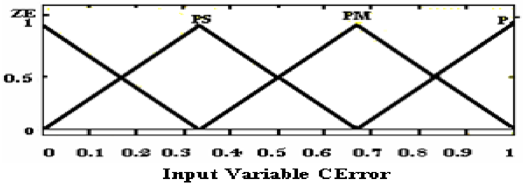


Fig. 2 Membership Functions of Input Fuzzy Variable "CError"

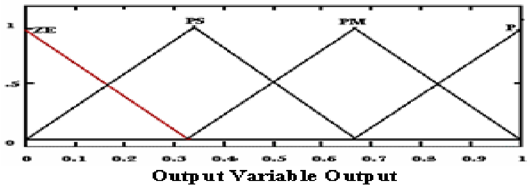


Fig. 3 Membership Functions of Output Fuzzy Variable "Output"

Fig. 4 shows 16 fuzzy rules (IF/THEN) used in the model. The connective ELSE term is interpreted as an intersection (OR operation) while the connective AND is given a minimum interpretation. The output is produced based on various combinations of the two fuzzy inputs.

1. IF ERROR is ZE AND CERROR is ZE THEN OUTPUT is ZE ELSE
2. IF ERROR is ZE AND CERROR is PS THEN OUTPUT is PS ELSE
3. IF ERROR is ZE AND CERROR is PM THEN OUTPUT is PM ELSE
4. IF ERROR is ZE AND CERROR is PL THEN OUTPUT is PL ELSE
5. IF ERROR is PS AND CERROR is ZE THEN OUTPUT is PS ELSE
6. IF ERROR is PS AND CERROR is PS THEN OUTPUT is PS ELSE
7. IF ERROR is PS AND CERROR is PM THEN OUTPUT is PM ELSE
8. IF ERROR is PS AND CERROR is PL THEN OUTPUT is PL ELSE
9. IF ERROR is PM AND CERROR is ZE THEN OUTPUT is PM ELSE
10. IF ERROR is PM AND CERROR is PS THEN OUTPUT is PM ELSE
11. IF ERROR is PM AND CERROR is PM THEN OUTPUT is PM ELSE
12. IF ERROR is PM AND CERROR is PL THEN OUTPUT is PL ELSE
13. IF ERROR is PL AND CERROR is ZE THEN OUTPUT is PL ELSE
14. IF ERROR is PL AND CERROR is PS THEN OUTPUT is PL ELSE
15. IF ERROR is PL AND CERROR is PM THEN OUTPUT is PL ELSE
16. IF ERROR is PL AND CERROR is PL THEN OUTPUT is PL

Fig. 4 Fuzzy IF/THEN rules

The Max-Min Composition method is used for the fuzzy model and Mamdani Min operator is chosen as the implication method. Centroid (Centre of Area or COA) defuzzification method is used in the model that is a well-known and commonly used method [7]. The COA method takes into account the area of the resultant membership function as a whole and favours “central values” in the universe of discourse (or region) [7].

Fig. 5 is a Matlab generated plot that shows the surface of the 16 fuzzy rules used in the model. It is important to note that the surface changes in a gradual and smooth manner as either/both fuzzy variables “Error**”** or “CError” increases from 0 to 1. This smooth change in the surface indicates that the rules as a whole are consistent and hence, an accurate output might be produced by the system. In addition, both inputs and output range have been normalised to within [0 to 1]. This gives the system a measure of flexibility in being adaptable to various input/output parameters, through the use of appropriate simple conversion circuits. Hence, the model is able to accommodate different processes and environments without major changes within the algorithm. Besides, that both inputs and the output range have been normalised within [0 to 1] that makes the system flexible and adaptable to a variety of input/output parameters, through the use of appropriate simple conversion circuits. Hence, the model is able to accommodate different processes and environments without major changes within the algorithm.

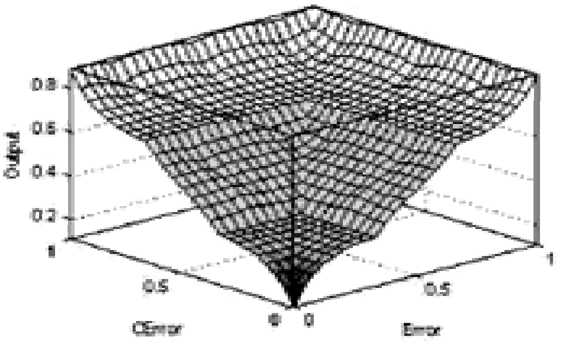


Fig. 5 Surfaces of Fuzzy Rules

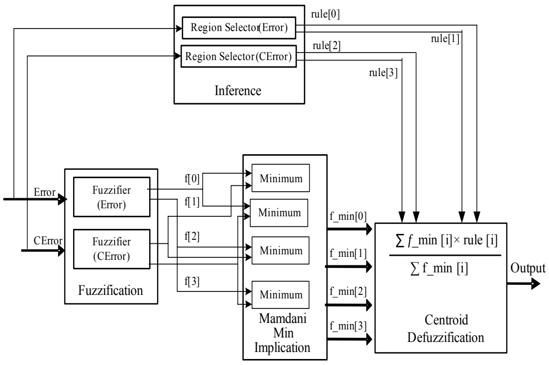


Fig. 6 Block diagram of FTC system

3 Modelling of the FTC in C++

We first developed the FTC algorithm in C++ that serves as a reference for the VHDL codes as well as a verification tool for the developed VHDL model.

Fig. 6 depicts the C++ Model. The FTC has been divided into four modules according to function, i.e. Fuzzification, Inference, Implication, and Defuzzification. It accepts 2 crisp inputs; "Error” and “CError”, and produces a crisp output value, “Output”, using 16 rules (descried in Fig 4).

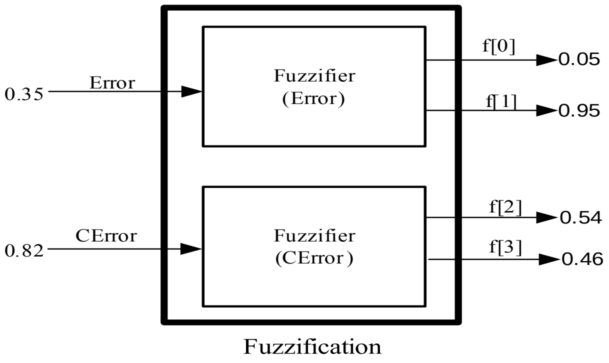


Fig. 7 Fuzzification model

3.1 Fuzzification module

This module as shown in Fig. 7 is divided into two similar parts; both serving the same function. The module accepts two crisp (i.e. real world) signals (“Error**”** and “CError”) and produces 4 fuzzified values (2 fuzzy values for each input) forward to Implication module. In this model, each input signal discourse upon 4 triangular membership functions using Equation (1).

$$Y = mX + c, \qquad (1)$$

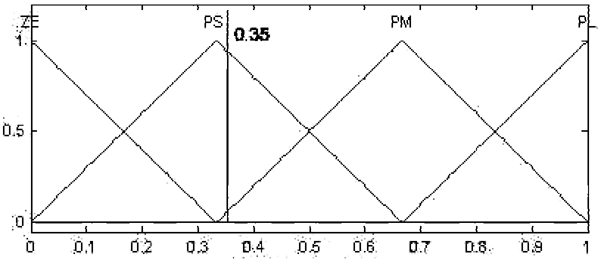


Fig. 8 Active membership functions for“Error” = 0.35

Where Y represents the fuzzy value, m represents the gradient of the membership function, X is the crisp input and c symbolises the intersection the membership curve makes with the Y-axis.For example, according to Fig. 8 when “Error” signal 0.35 intersects with fuzzy variables PS and PM, where PS is taken as the first fuzzy variable f[0] and PM is taken as the second fuzzy variable, f[1].

Based on the membership functions, f[0] = 0.05 and f[1] = 0.95 shown in Fig 7. Similarly, in Fig. 9, “CError”=0.82 intersects with the fuzzy membership functions PM and PL, where PM is assigned to f[2] = 0.54 and f[3] = 0.46 corresponds to PL shown in Fig 7

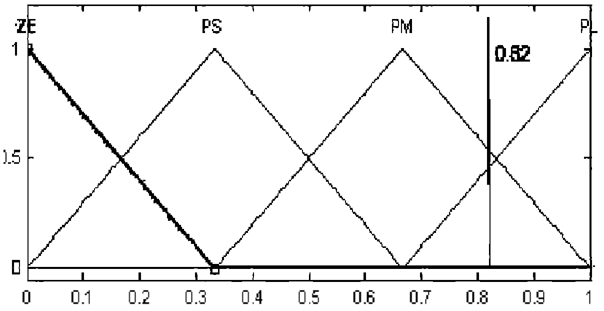


Fig. 9 Active membership functions “CError”= 0.82

3.2 Inference module

In this module, appropriate rules are selected to be fired based on the fuzzy variables that are chosen according the regions that the variables fall in.

Table 1: Regions of Error and “CError”

Input	Region	Range
Error/ CError	A	0.0000 to 0.3334
	B	0.3334 to 0.6666
	C	0.6666 to 1.000

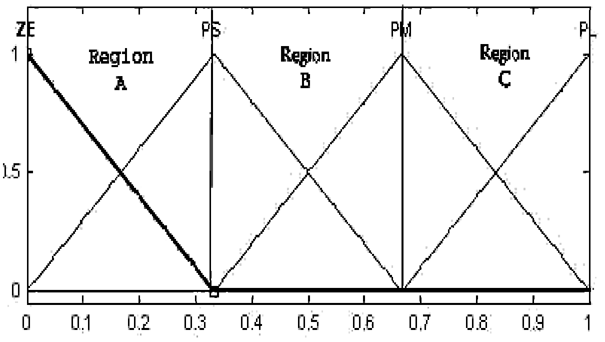


Fig.10 Region Division for “CError”

Only two fuzzy variables are activated at any given time. As a result, each fuzzy variable results in firing two rules. As a consequence, a total of 4 rules are fired.

This process is achieved by dividing the universe of discourse into 3 regions where each region containing only two fuzzy variables. Table 1, shows “Error” = 0.35 and “CError” = 0.82 lies in region B and in region C respectively.

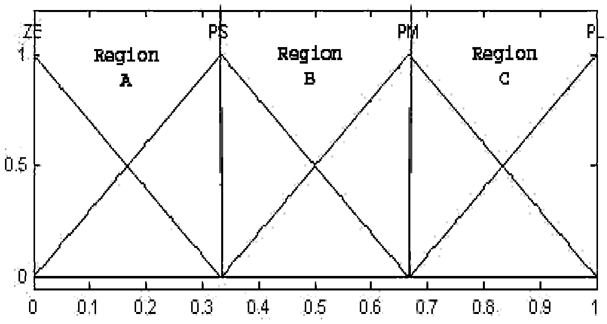


Fig.11 Region Division for “Error”

Fig.10 and Fig.11 show the universe of discourse divided according to the stated regions. Region B of “Error” containing fuzzy variables PS and PM; and region C of “CError” holding fuzzy variables PM and PL. As a consequence, rules 7, 8, 11 and 12 stated in Fig. 4 will be selected. The “Output” value is represented using fuzzy singleton sets in place of membership functions like those of the two inputs. The use of singleton values allows faster inferencing as well speeding up the defuzzification process. The downside of singleton values is that a certain amount of accuracy is sacrificed as each output values now represents a range of input values. Table 2, shows the four singleton values chosen to represent the fuzzy output variable.

Table 2: Singleton Values for Fuzzy Output

Output	Singleton Value	Location of Discourse
ZE	0	0.0000
PS	33.34	0.3334
PM	66.66	0.6666
PL	100	1.0000

Hence, when rules 7, 8, 11 and 12 are fired, the following singleton values (66.66, 100, 66.66, and 100) are assigned to the outputs of the module shown in Fig.12.

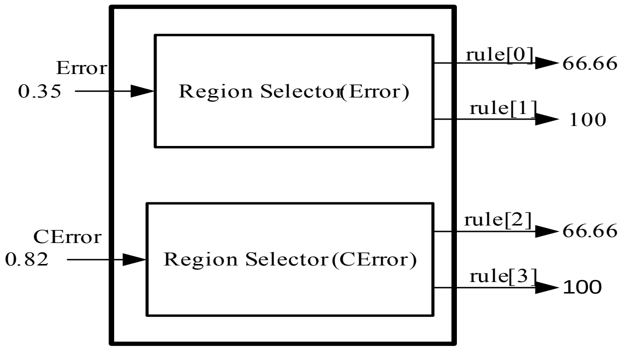


Fig.12 Inference module

3.3 Implication module

This module receives the 4 fuzzy variables from the Fuzzification Module and performs the Mamdani Min implication operation on the combination of these four variables. Continuing the previous examples, composition functions and the implication operation are described in Fig 13.

f_min[0]=f[0]^f[2]=f[0] AND f[2]=0.05^0.54= 0.05
f_min[1]=f[0]^f[3]=f[0] AND f[3]=0.05^0.46= 0.05
f_min[2]=f[1]^f[2]=f[1] AND f[2]=0.95^0.54= 0.54
f_min[3]=f[1]^f[3]=f[1] AND f[3]=0.95^0.46= 0.46

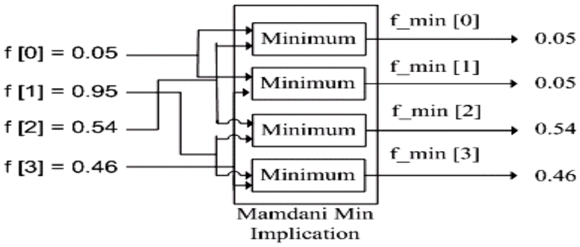


Fig.13 Implication module

The four resulting output of the implication operation will then be fed into the Defuzzification module.

3.4 Defuzzification module

This module accepts four inputs each from the Implication module (i.e. f_min[0-3]) and the Inference module (i.e. rule[0-3]) and produces the defuzzified/crisp output signal for the control system. Centre of Area (Centroid) defuzzification scheme is chosen expressed in Equation (2).

$$\frac{\sum f\_\min[i] \times rule[i]}{\sum f\_\min[i]} \qquad (2)$$

The output of the module is in the form of a percentage value, which can be easily converted into a normalised form. Using the values from the previous example, the output of the defuzzification module is calculated 82.113% i.e. 0.82113 (normalized).

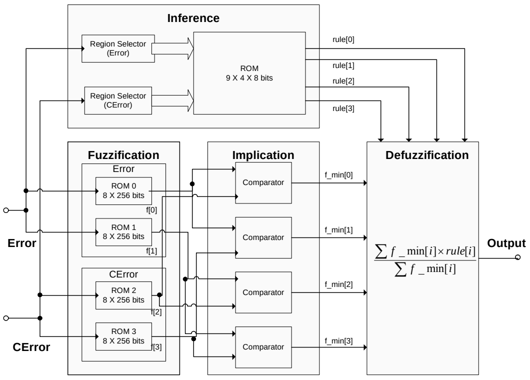


Fig.14 VHDL Model of FTC system

4 VHDL Implementation

The FTC model has been converted into behavioural level using VHDL. The generated VHDL four hardware components are interconnected in a similar manner as the C++ model shown in Fig 14.

Fig 15 shows the Register Transfer Level (RTL) view of the FTC with all the declared signals to establish relationships between the various hardware components.

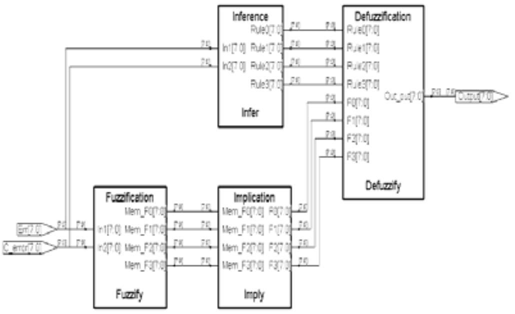
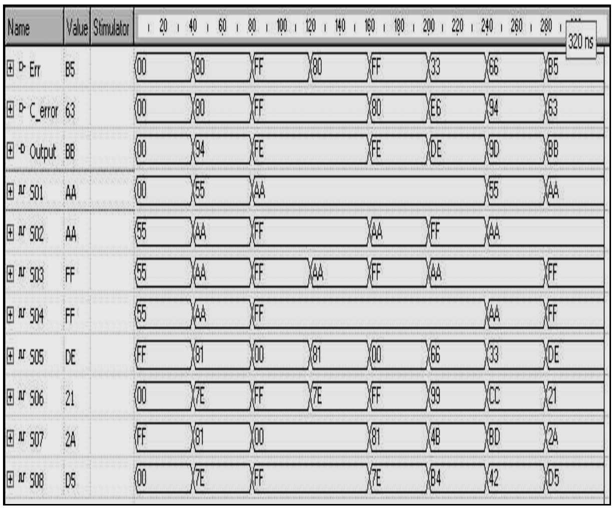


Fig.15 RTL view of the FTC System

5 Functional and Timing Simulations

Upon successful completion of the VHDL coding functional simulation is performed to verify the correct functionality and to determine the deviation or tolerance parameters of the FTC using generated test bench.

A set of stimuli as inputs (functional vectors that changes with at fixed time duration) is fed into the test bench. The waveform in Fig 16 shows the values of the inputs and the corresponding output in hex form at the various instances determined by the stimuli in the test bench.



Name	Value	Stimulator	20 40 60 80 100 120 140 160 180 200 220 240 260 280 320 ns
Err	B5		00 80 FF 80 FF 33 66 B5
C_error	63		00 80 FF 80 E6 94 63
Output	BB		00 94 FE FE DE 9D BB
S01	AA		00 55 AA 55 AA
S02	AA		55 AA FF AA FF AA
S03	FF		55 AA FF AA FF AA FF
S04	FF		55 AA FF AA FF
S05	DE		FF 81 00 81 00 66 33 DE
S06	21		00 7E FF 7E FF 99 CC 21
S07	2A		FF 81 00 81 4B BD 2A
S08	D5		00 7E FF 7E B4 42 D5

Fig 16 Waveform of functional simulation of the FTC

The same test stimuli are used for timing simulation taking into account the propagation delay. In addition, the simulation at this stage is performed upon nodes that are synthesisable. Slice of the timing waveform is shown in the Fig 17.

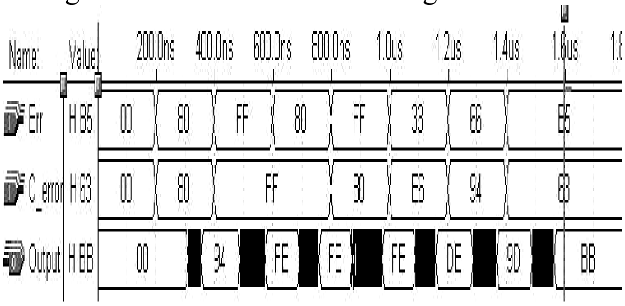


Fig 17 Waveform of Timing Simulation of the FTC

It is important to note that, the outputs are exactly the same for both functional and timing simulation. The difference between these two simulations is that there is a noticeable time delay before the output is available upon the assertion of a set of input. Notice also that the output requires a period before its value stabilises.

6 Synthesis of the FTC

Synthesis is the process of transforming one representation in the design abstraction hierarchy to another representation. Synthesis process has performed using synplify tools [8] for synthesizing the compiled VHDL design codes into gate level schematics. The synthesis tool is also used to optimize the gate level design for area by applying specified options. It initially processes the VHDL building blocks such as multiplier, registers, gates and flip-flops etc, for which it can determine whether logic blocks can be shared between the building blocks function for efficiency performance. While synthesizing the design with the synthesis tool, HDL library browser was used to synthesize the design in a hierarchical manner.

In this step, the VHDL codes are synthesized for converting into RTL view of the FTC architecture. The Technology mapping has chosen in this project from Altera's FLEK10K70 with RC240 package and a speed grade of -4. Then the technology view of the various modules for FTC chip has been carried out. As an example, the flattened technology view of overall FTC system is given in the Fig 18.

The synthesized schematic is also simulated to ensure the synthesized design functions the same way as the validated VHDL model for VLSI implementation.

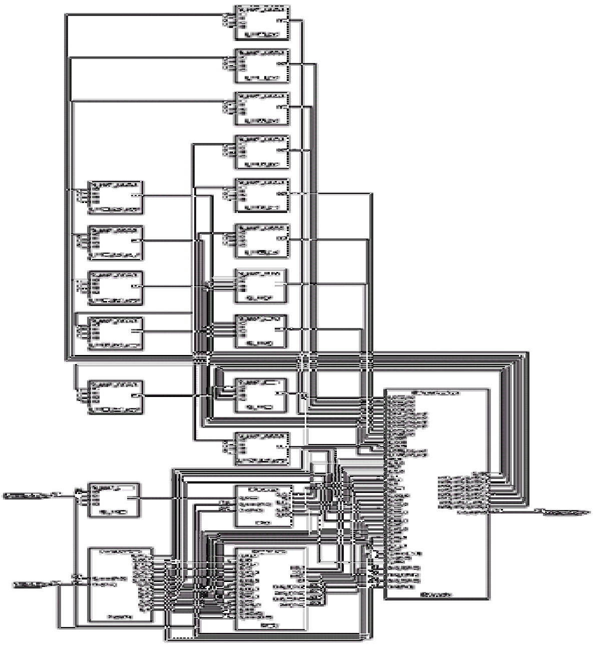


Fig.18 Flattened Technology View of Overall FTC system

7 FPGA Implementation

The generated synthesized netlist of the FTC chip has been downloaded into FPGA (FLEX10K EPF10K70) board from Altera for verification the correctness of the algorithm functionality. Note that the FPGA board contain a built-in 8-bit DIP switch, a dual-digit 7 Segment display, and three expansion slots, each with 42 I/O pins and 7 global pins as shown in Fig 19.

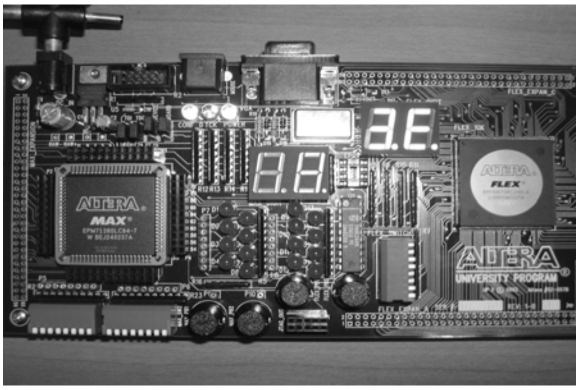


Fig.19 Demonstration of FPGA enabled fuzzy algorithm

To make compatible the fuzzy based FTC chip inputs with I/O pins in FPGA board, the 8-bit DIP switches as manual inputs and 8-bit of two 7-segment LEDs as outputs are chosen for verifying he correct functionality of the FTC chip for VLSI implementation.

Table 3: Summary of FPGA Details

Resources used	1865/3744, Logic Cells (49%)
	34,300 / 70,000 Gates (49%)
Max Operating Frequency	5.00 MHz
Critical path	Error7-Output7 (199.3ns)

We have seen that the maximum propagation delays of any paths between the input and output nodes are around 200ns at the most. Hence, an approximate maximum operating frequency of the FTC chip can be inferred at around 5.00MHz. Furthermore, a similar inference can be made for the critical path. Based on the longest maximum propagation delay, it can be said that the critical path is from the Error7 to Output7, taking a duration of 199.3ns. The statistical results on FPGA implementation is given in Table 3.

8 Application

Temperature control is widely used in various processes. These processes, no matter it is a process of large industrial plant, or a process in home appliance, share several unfavourable features such as non-linearness, interference, dead time, and external disturbance, etc.

Conventional control approaches usually cannot achieve satisfactory results for this kind of processes. Besides this all other processes that requiring temperature control has various unfavorable characteristics including non-linearity, dead zone time, external disturbances and so on. Currently used conventional approximations do not produce satisfactory temperature controls for controlling complex processes, which is usually the case in the industry because they suffer from various drawbacks such as slow stabilization, overshooting and overall slow response.

This fuzzy system based temperature controller can be applied in any kind of environment by which we can get an improvement of relative performance with respect to the conventional scheme. It compensates non-linear errors, accelerates the response and reduces the steady-state error. The FLC is also able to bring keep the temperature constant at the desired value regardless of changes in the load or environment. Thus we can experience a great solve of the overshooting problem. It also can able to improve the slow stabilization problem. Thus it application can be implemented on almost all scale industry.

9 Conclusion

A fuzzy logic temperature controller has been designed with an industrial application in mind. The system has been coded, compiled and simulated in VHDL using EDA tools, specifically_Aldec Active-HDL 3.5. The hardware implementation demonstrated complete, correct functionality and met all the initial system requirements. The hardware components of the FTC chip has been verified using FPGA board and ensure that the FTC chip work properly._At present the system inferred maximum operating frequency is 5MHz with a critical path of 199.3ns. This could take advantage of the high speeds achievable using hardware, and as a result would be a beneficial and economic investment for designs requiring fuzzy logic.

References:

[1] Zhiqiang, Gao, Trautzsch, T.A., and Dawson, J.G., A stable self-tuning fuzzy logic control system for industrial temperature regulation, *IEEE Industry Applications Conference,* Vol.2, 2000, pp. 1232 – 1240.

[2] Thyagarajan, T., Shanmugam, J., Ponnavaikko, M., and Rao, P.G., Advanced control schemes for temperature regulation of air heat plant, *IEEE International Fuzzy Systems Conference Proceeding,* Vol.2, 1999, pp. 767 – 772.

[3] Ayala, I.L., and Solis, I.J. 1991. IEEE Transactions on Industry Applications. Volume: 27 , Issue: 1, pp:108 – 111

[4] Coeyman, B., and Bowles, J.B., Fuzzy logic applied to reboiler temperature control, *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*, Vol.1, 1996, pp.511 – 516.

[5] K.T. Tho, K.H. Yeow, F. Mohd-Yasin, M.S. Sulaiman, and M.I. Reaz, VHDL Modeling of Boolean Function Classification Schemes for Lossless Data Compression, *WSEAS Transactions on Computers,* Vol.3, No.2, 2004, pp. 365-368.

[6] Brown, S.D., Francis, R.J., Rose, J., and Vranesic, Z.G.,*Field-Programmable Gate Arrays*, Kluwer Academic Publishers,1996

[7] Mohd-Yasin, A. Tio, M.S. Islam, M.I. Reaz and M.S. Sulaiman, VHDL Prototyping of Temperature Controller Based on Fuzzy Logic for Industrial Application, *2nd International Conference on Artificial Intelligence in Engineering and Technology, Sabah, Malaysia.*

[8] http://www.synplicity.com/.