Volume 2, Issue 5, May 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Introduction of 64-bit Fuzzy Inference Processor through the Implementation of System with 8-Bit FIP

Kamini M. Sayam*
M-Tech VLSI & R.T.M.
Nagpur University

Prof. S. V. Bhalerao
Department of Electronics & R.T.M.
Nagpur University

Abstract— The design of 64-bit fuzzy processors, such as fuzzifier, defuzzifier, inference engine and rule base so that to use this designed system in a real time application so that the speed of the process increases to a very far level due to which the overall efficiency must be increased so that it should be very beneficial for user. By designing the same for 8-bit fuzzy inference processors we can get the corrected traffic at output of ATM module. For the better result and to speed up the processing time for ATM module we can implement the same system with 64-bits for more combination of different input parameters.

Keywords— Fuzzifiers, Defuzzifiers, Inference engine, Rule base, ATM Network.

I. INTRODUCTION

Today most fuzzy logic systems are implemented software on microprocessors. If there is a need for higher operation speed, we can add a fuzzy processor. For 8-bit and 16-bit systems the FP speeds-up the system. But when using a 32 bit or 64-bit microprocessor, an optimized fuzzy controller algorithm implement as software is faster than the simple FP. Our idea of a 64-bit fuzzy system was to minimize the needed hardware requirement for the fuzzy processor by using already existing components of the 64-bit microprocessors and implement only those components which speed-up the fuzzy algorithm. This project covers the circuit and architecture level designing of various components of the fuzzy processors, such as, fuzzifiers, defuzzifiers, inference and rule base. A comparative analysis of the performance of these components will be performing. Further, it is seen that the design emphasis should be more on inference engine performance and defuzzification units, because of the complexity of computations handled by them. The optimization in these units results in a significant improvement in the overall performance of the system. The methods and approaches adapted for the realization of four important components of fuzzy processors, such as, fuzzifier, defuzzifier, inference engine and rule-base has been thoroughly studied. The overall performance of the processors has a significant dependence on the performance and optimization in these units. The architecture and circuit level designing of the fuzzy processors will finally realized in the form of FPGAs in most of the cases using VHDL and Verilog hardware description languages. Fuzzy logic is being accorded increasing interest in various fields of application such as process control, decision-making support systems and According to the field of application, fuzzy systems feature time constraints that may differ considerably. In washing machine controls and auto focus imaging, for instance, the inference execution speeds required are quite low, while in real-time applications they are extremely high. One possible approach to computing fuzzy inferences uses suitably programmed traditional general-purpose processor. The Key features of the design including its architecture, data path and instruction set are presented. The design is implemented using VHDL and verified on Xilinx ISE simulator.

II. LITERARTURE RIVIEW

To achieve two merits from the advantage of acceleration of the fuzzy inference over usual microprocessors. Authors of this paper [1] have considered and designed a VLSl fuzzy chip, in which suitable architecture is adopted for processing many rules and for using it as an external memory chip. In this the architecture of the fuzzy chip is explained in relation to the above merits and the circuit design with a module compiler is outlined for some characteristic circuit blocks.

The experimental processor chip described in [2] is a mixed analog-digital inference engine with eight inputs and four outputs. Because it calculates the results of 32 rules in parallel, its effective throughput can be orders of magnitude faster than that of a sequential processor. This processor is designed to perform inferences over a set of 32 rules, each of which must be cast in the form.

Paper [3] presents the mixed analog-digital fuzzy logic inference processor chip calculates the result of an inference over a 32-rule knowledge base in parallel. Simulations predict a computation time for the array of about 2 psec.

In this paper [4] authors had described the simplified architecture of a FIP By treating inputs of fuzzy system as a fuzzy singletons can be implemented using mainly min/max functional blocks and presentable counters the inference speed of the simplified architecture of FIP is enhanced significantly.

The architecture of a 64-bit Fuzzy Inference Processor is presented in this paper [5]. A fuzzy system consisting of the FIP and a 64-bit microprocessor speeds-up the inference by up to 10 times. In addition authors present an optimized inference algorithm which achieves 50 fold acceleration for the calculation of the rule base. The FIP will be used only for the inference while the fuzzification and defuzzification will be done by the microprocessor OLp, which will also do the controlling of the FIP. This results in a simple architecture and low hardware requirement. They use the MINIMAX algorithm and an internal resolution. Up to 8 membership functions can be used for every input and output. A prototype (with 32-bit) was simulated on FPGA's and needed 18011s for the calculation of one rule with 8 inputs and 2 outputs. Using the FIP and the new optimized organization of the rule base, a 64-bit fuzzy system is nearly as fast as special fuzzy ASIC's.

Small or portable systems require compact fuzzy chips. In response to these needs, authors [6] designed the SAE 81C99 fuzzy inference processor. It is highly flexible and operates very quickly. In general, it outperforms Standard hardware by at least one order of magnitude.2 Moreover, due to its very small silicon area, the SAE 81C99 is the first fuzzy logic hardware module that can be integrated at low cost on a microcontroller chip as a coprocessor macro cell. The processor implements its knowledge-base memory as an application-specific, on-chip ROM, or as an off-chip memory device of arbitrary type.

In this paper [7] author presents a design of an analogue fuzzy inference processor with emphasis on the simplicity of architecture, circuitry and implementation. The design of an analogue fuzzy processor with 4-rule aggregation is discussed, based on a standard 2pm N-well process. A complete 4-rule analogue fuzzy inference processor has been designed for a 2mm x 2mm CMOS VLSI chip, with the core circuitry occupying 900pm x 1100pm. Their proposal, which emphasises the simplicity of architecture, circuitry and implementation, is highly suited towards the development of application specific fuzzy logic processors with far greater inference processing power.

A novel Signed-Digital fuzzy processor [8] using the logic oriented neural networks is proposed. Since the signed-digit number system has a redundant property to represent the binary numbers, the high speed adder in the processor can be realized in the signed-digit system without a delay of the carry propagation. In this paper the novel circuits of the main two systems of inference engine and defuzzifier in the fuzzy processor are constructed with the logic oriented neural networks. The results show that the proposed circuits can perform the operations in higher speed than those of common fuzzy processor. A novel Signed-Digital fuzzy processor by using LOGO neural networks has been proposed. The results show the reliable operation has been obtained with a very high performance of a simple construction.

The paper [9] presents the design of a VLSI architecture dedicated to execution of fuzzy inference. The execution speed is up to 16 MFLIPS for inferences with 128 rules with 4 linguistic variables, 1 I MFLIPS with 8 linguistic variables. This level of performance is obtained by means of a method that reduces the number of rules to be processed. Moreover, rule processing is split into a sequence of pipeline stages which make it possible to process the active rules at each clock cycle and a certain degree of parallelism is introduced into these stages. In this paper authors have presented the architecture of a processor capable of efficient fuzzy inference execution. The architectural model defined provides very high execution speeds, up to 16.7 MFLIPS for inferences with 128 rules with 4 linguistic variables. These performance levels were reached by means of an efficient method which allows detection of the active rules to be processed. Rule processing has been split into a sequence of pipeline stages which make it possible to obtain the degree of activation of the rules at each clock cycle.

The paper [10] presents the architecture of a VLSI processor for applications based on fuzzy logic. The main features of the architecture are a pre-computation phase of the positive degree of truth of the antecedent with fuzzy inputs; a detection phase of the rules positive degree of activation, parallelism in some phases of inference which is split into a sequence of pipeline stages. The processing speed is up to 16.7 MFLIPS for inferences with 64 rules with 4 linguistic variables. The processor was modelled in VHDL, successfully simulated, and then synthesized using CMOS 0.5pm as target technology. The processing speed, expressed in MFLIPS, reaches 16.7MFLIPS. The silicon area estimated is 20 mm2.

A simple CMOS fuzzy processor [11] using a neuron-MOS Center-of-Mass detector as the defuzzifier has been proposed. The processor receives the membership values in analog voltages and fuzzy inference is conducted by an array of MINIMAX circuits. The neuron-MOS defuzzifier yields a crisp output as a digital code, thus providing a smooth interfacing between the analog sensory inputs and digital systems. In addition, the crisp output value is given by a digital code, presenting the possibility of very smooth interfacing to digital systems. The vMOS COM circuit has been shown to operate well within 100 nsec.

For medical diagnosis an auto-decision-making system [12] which has high performance, low-power, pipelined parallel fuzzy processor based on a dedicated single-chip architecture performs high-speed fuzzy inferences with processing speed up to 5.0 mflips at a clock frequency of 40 MHz using 256 rules having one consequent each, 16 input variables, and 16-bit resolution. The processor operates in real time producing results within an interval of 1.92ms. The processor implemented on board consumes as low as 70 mill watt power. The processor performs medical diagnosis with 97.5 percent accuracy. The architecture of a 64-bit fuzzy inference processor (FIP) is presented. A fuzzy system consisting of the FIP and a 64-bit microprocessor speeds-up the inference by up to 10 times. In addition author presents an optimized inference algorithm which achieves 50 fold acceleration for the calculation of the rule base. The FIP will be used only for the inference while the fuzzification and defuzzification will be done by the microprocessor (μP), which will also do the controlling of the FIP. This results in a simple architecture and low hardware requirement. We use the min/max algorithm and an internal resolution of 8-bit. Up to 8 membership functions can be used for every input and output. A prototype (with 32-bit) was simulated on FPGA's and needed 180 ns for the calculation of one rule with 8 inputs and 2 outputs.

III. PROPOSED SYSTEM

To design & study four important components of a 64-bit fuzzy processors, such as fuzzifier, defuzzifier, inference engine and rule base so that to use this designed system in a real time application so that the speed of the process increases to a very far level due to which the overall efficiency must be increased so that it should be very beneficial for user. The project consist of architecture level designing of various components of the fuzzy processors, such as, Fuzzifiers, Defuzzifiers, Inference engine Rule engine.

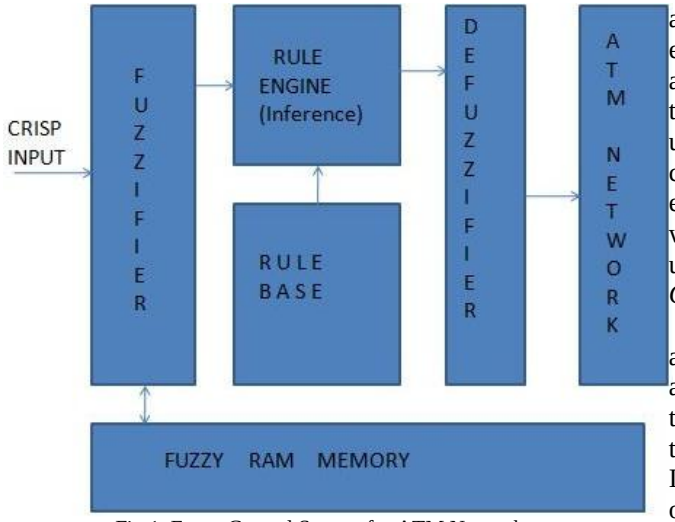

Fig 1. Fuzzy Control System for ATM Network

As the fig. 1 shows our proposed system has basically four units along with ATM network

A. Fuzzification

Fuzzification is the process of decomposing a system input and/or output into one or more fuzzy sets. Many types of curves can be used, but triangular or trapezoidal shaped membership functions are the most common because they are easier to represent in embedded controllers. Each fuzzy set spans a region of input (or output) value graphed with the membership. Any particular input is interpreted from this fuzzy set and a degree of membership is interpreted. The membership functions should overlap to allow smooth mapping of the system. The process of fuzzification allows the system inputs and outputs to be expressed in linguistic terms so that rules can be applied in a simple manner to express a complex system.

B. Rule Base Design

The selection of rule base is based on our experience and beliefs on how the system should behave. Design of a rule base is two-fold: First, the linguistic rules are set; afterwards, membership functions of the linguistic values are determined. The trade-off involving the design of the rule base is to have a set of minimum number of linguistic rules representing the control surface with sufficient accuracy to achieve an acceptable performance. Recently, in the fuzzy control literature, some formal techniques for obtaining a rule base by using Artificial Neural Networks or Genetic Algorithms have appeared. Nevertheless, we have used the conventional trial and error approach under the guidance of some design rules of thumb. Usually, to define the linguistic rules of a fuzzy variable, Gaussian, triangular or trapezoidal shaped membership functions are used. Since triangular and trapezoidal shaped functions offer more computational simplicity, we have selected them for our rule base. Then, the rule base is tuned by observing the progress of simulation, such as cell loss occurrences and demand versus throughput curves. The tuning can be done with different objectives in mind. For example, any gain in throughput must be traded off by a possible increase in the delay experienced at the terminal queues. However, since the tuning of the fuzzy rules is intuitive, and can be related in simple linguistic terms with user's experience, it should be a straightforward matter to achieve an appropriate balance between a tolerable end-to-end delay and the increase in throughput. Alternatively an adaptive fuzzy logic control method can be used which can tune the parameters of the fuzzy logic controller on line, using measurements from the system. The tuning objective can be based on a desired optimization criterion, for example, a trade-off between maximization of throughput with minimization of end-to-end delay experienced by the users.

C. Inference Engine

In artificial intelligence, an inference engine is a computer program that tries to derive answers from a knowledge base. It is the "brain" that expert systems use to reason about the information in the knowledge base for the ultimate purpose of formulating new conclusions. Inference engines are considered to be a special case of reasoning engines, which can use more general methods of reasoning. The inference engine is the generic control mechanism that applies the axiomatic knowledge present in the knowledge base to the task-specific data to arrive at some conclusion. This is the second key component of all expert systems. Having a knowledge base alone is not of much use if there are no facilities for navigating through and manipulating the knowledge to deduce something from it. As a knowledge base is usually very large, it is necessary to have inference mechanisms that search through the database and deduce results in an organized manner. A few techniques for drawing inferences from a knowledge base are described here. In simple rule-based systems, there are two kinds of inference, forward chaining and backward chaining. In order to execute a rule-based expert system using the method of forward chaining us merely need to fire actions whenever they appear on the action list of a rule whose conditions are true. This involves assigning values to attributes, evaluating conditions, and checking to see if all of the conditions in a rule are satisfied. A general algorithm for this might be: while values for attributes remain to be input read value and assign to attribute evaluate conditions fire rules whose conditions are satisfied several points about this require consideration.

D. Defuzzification

After fuzzy reasoning we have a linguistic output variable which needs to be translated into a crisp value. The objective is to derive a single crisp numeric value that best represents the inferred fuzzy values of the linguistic output variable. Defuzzification is such inverse transformation which maps the output from the fuzzy domain back into the crisp domain. Some defuzzification methods tend to produce an integral output considering all the elements of the resulting fuzzy set with the

corresponding weights. Other methods take into account just the elements corresponding to the maximum points of the resulting membership functions. The following defuzzification methods are of practical importance.

E. Asynchronous Transfer Mode

ATM is a standard switching technique designed to unify telecommunication and computer networks. It uses asynchronous time-division multiplexing and it encodes data into small, fixed-sized cells. This differs from approaches such as the Internet Protocol or Ethernet that use variable sized packets or frames.ATM provides data link layer services that run over a wide range of OSI physical Layer links. ATM has functional similarity with both circuit switched networking and small packet switched networking. It was designed for a network that must handle both traditional high-throughput data traffic and real-time, low-latency content such as voice and video. ATM uses a connection-oriented model in which a virtual circuit must be established between two endpoints before the actual data exchange begins. ATM is a core protocol used over the SONET/SDH backbone of the PSTN and Integrated Services Digital Network, but its use is declining in favors of All IP. ATM has the potential of revolutionizing data communications and telecommunications. Based on the emerging standards for Broadband Integrated Services Digital Networks, ATM offers the economically sound bandwidth on demand features of packet-switching technology at the high speeds required for today's LAN and WAN networks and tomorrows. ATM is a cell-relay technology that divides upper-level data units into 53-byte cells for transmission over the physical medium. It operates independently of the type of transmission being generated at the upper layers AND of the type and speed of the physical-layer medium below it. This allows the ATM technology to transport all kinds of transmissions in a single integrated data stream over any medium, ranging from existing T1/E1 lines, to SONET OC-3 at speeds of 155 Mbps, and beyond.ATM is the hottest network technology in the marketplace, enabling extremely high-speed transmission of all user traffic, including voice, data and video. The UNI 3.1 standard enables developers to build equipment that will fully interoperate in today's public and private ATM network environments. This is the only authoritative guide to this new standard, by The ATM Forum, the industry-led consortium that created it. The book presents highly-detailed specifications for interfacing ATM user devices, public and private network equipment. It also covers significant Version 3.1 changes in signaling and signaling virtual channels. : Communications engineers and programmers developing ATM applications and equipment. Version 3.0 of this title was a Prentice Hall PTR bestseller.

F. The ATM Network with Interface

The technology allows both public and private ATM networks. This capability gives a seamless and transparent (to the user) connection from one end user to another end user, whether in the same building or across two continents. The basic network structure is as shown on the following page. Three types of interfaces exist their. User-to-Network Interface (UNI) Network-to-Network Interface (NNI) Inter-Carrier Interface (ICI) The UNI exists between a single end user and a public ATM network, between a single end user and a private ATM switch, or between a private ATM switch and the public ATM network of an RBOC. The NNI exists between switches in a single public ATM network. NNIs may also exist between two private ATM switches. The ICI is located between two public ATM networks All of these interfaces are very similar. The major differences between these types of interfaces are administrative and signaling related. The only type of signaling exchanged across the UNI is that required to set up a VIRTUAL CHANNEL for the transmission. Communication across the NNI and the ICI will require signaling for virtual-path and virtual-channel establishment together with various exchange mechanisms for the exchange of information such as routing tables, etc.

To carry out the desired goal we have to design the project with various modules and their interfaces. In this project we are using 64- bit Fuzzy Inference Processor just to speed up the execution of process. In our project there are four different modules has to be designed with the help of Xilinx and then system is implemented with the help of FPGA kit to get desired results.

Modules to be implemented

1. Architecture of fuzzification unit with the various methods used to realize fusiliers.
2. The defuzzification Unit.
3. The inference engine Unit.
4. The rule-base system.

For an instance before developing 64-bit fuzzy processor here we are doing the same thing with 8-bit fuzzy processor. So for designing fuzzifier, defuzzifier, and inference engine and rule base 8 bit fuzzy processor is designed which has simulates in Xilinx as shown below.

With the 8-bit fuzzy processor initially we have to give the crisp value to Fuzzifier which has the fuzzy sets for two membership functions as low membership function & High membership function which is again having their own sub functions as low, medium and high. As a result of this fuzzifier will give the fuzzified output in terms of 0, 1 and 2 which are having their own format. After this rule engine will check the combination from the rule table and give the corresponding output for 0, 1 and 2. Now this output is applied to the defuzzifier which converts linguistic output variable into the crisp value corresponding to the variable value. Now this crisp value is given to the ATM module which has three inputs as the average cell arrivals start, the second input is average cell arrivals previously and the last third input is allowed traffic for the network. So by checking all these three inputs and by taking the appropriate decision based on rule base ATM module gives corrected traffic at the output which is shown in fig. 2 the simulation results in the Model sim for different combination of inputs as below.

IV. CONCLUSIONS

By designing the fuzzifier, defuzzifier, inference engine and rule base for an 8-bit fuzzy inference processors we can get the corrected traffic at output of ATM module. The same thing can be achieved with 64-bits for more combination of inputs such that the system can be is designed in a real time application so that the speed of the process increases to a very far level due to which the overall efficiency must be increased so that it should be very beneficial for user.

REFERENCES

[1] H. Ikeda, Y. Hiramoto and N. Kisu Electronics Research Laboratory, Central Engineering Laboratories, Nissan Motor Co., Ltd. 1 Natsushima-Cho. Yokosuka 237, a Fuzz Y Inference Processor R with an "Active - Rule-Driven" Architecture.

[2] J. Fattaruso, S. S. Mahant-Shetti, J. B. Barton A Fuzzy Logic Inference Processor, Semiconductor Process And Design Center Texas Instruments Inc. Pob 655474 Ms 446 Dallas, Tx 75265

[3] Herbert Eichfeld*, Martin Klimke', Manfred Menke*, Jurgen Nolles" And Thomas Kunemund*, A General-Purpose Fuzzy Inference Processor, *Siemens Ag, Corporate Research and Development, D-8 1730 Munich, Germany "Siemens Ag, Semiconductor Group, D-81617 Munich, Germany.

[4] J.S. Ho, M.H. Liiii And K.T. Lau School Of Electrical And Electronic Engineering Nanyang Teclnological University, Singapore ,Simplified Architecture Of A Fuzzy Inference Processor 0-81864260-2/9$30 3.00 0 1993 IEEE

[5] Ansgar P. Ungering, Karl Goser University Of Dortmund, Ls-Be, Emil-Figge-Str. 68,44227 Dortmund, Germany , Architecture Of A 64-Bit Fuzzy Inference Processor, , 0-7803-1896-X/94 $4.00 01994 IEEE

[6] Herbert Eichfeld, Martin Klimke, Manfred Menke, Jurgen Nolles, Thomas Kunemund Siemens Ag A General-Purpose Fuzzy Inference Processor, 0272-1732/95/$04.00 8 1995 IEEE.

[7] Ctp Song, Sf Quigley, S Pammu, School Of Electronic And Electrical Engineering, University Of Birmingham, Birmingham, B 15 2tt, England: A Novel Cmos Analogue Fuzzy Inference Processor, 0-7803-4455-3/98/$10.00 0 1998 IEEE

[8] Masahiro Sakamoto, Daisuke Hamano And Mititada Morisue, Faculty Of Information Sciences Hiroshima City University, Hiroshima, 73 1-3 194 Japan, 1999 IEEE International Fuzzy Systems Conference Proceedings, August 22-25, 1999, Seoul, Korea, A Study Of A Radix-2 Signed-Digital Fuzzy Processor Using The Logic Oriented Neural Networks ,

[9] Giuseppe Ascia And Vincenzo Catania, Istituto Di Informatica E Telecomunicazioni, V.Le A. Doria, 6-95125-Catania-Italy, 1999 IEEE International Fuzzy Systems Conference Proceedings August 22-2j, 1999, Seoul, Korea, A High Performance Processor For Applications Based On Fuzzy Logic.

[10] Giuseppe Ascia And Vincenzo Catania Universith Di Catania Istituto Di Informatica E Telecomunicazioni V.Le A. Dona, 6-95 125-Catania-Italy, A Pipeline Parallel Architecture For A Fuzzy Inference Processor, 0-7803-5877-5/00/$10.Q0 0 2000 IEEE

[11] Yu Ning Mei* Chen Jingjindepartment Of Electron Engineering, Xi'an University Of Technology, Xi' An 710048,China, ,An Analog Fuzzy Processor Using Neuron-Mos Technology, 0-7803-85u-X/04/$20.00 ©2004 IEEE

[12] Shubhajit Roy, Chowdhury, Hiranmay Saha Jadavpur University. Published By The IEEE Computer Society A High-Performance Fpga-Based Fuzzy Processor Architecture For Medical Diagnosis, 0272-1732/08/$20.00 G 2008 IEEE.

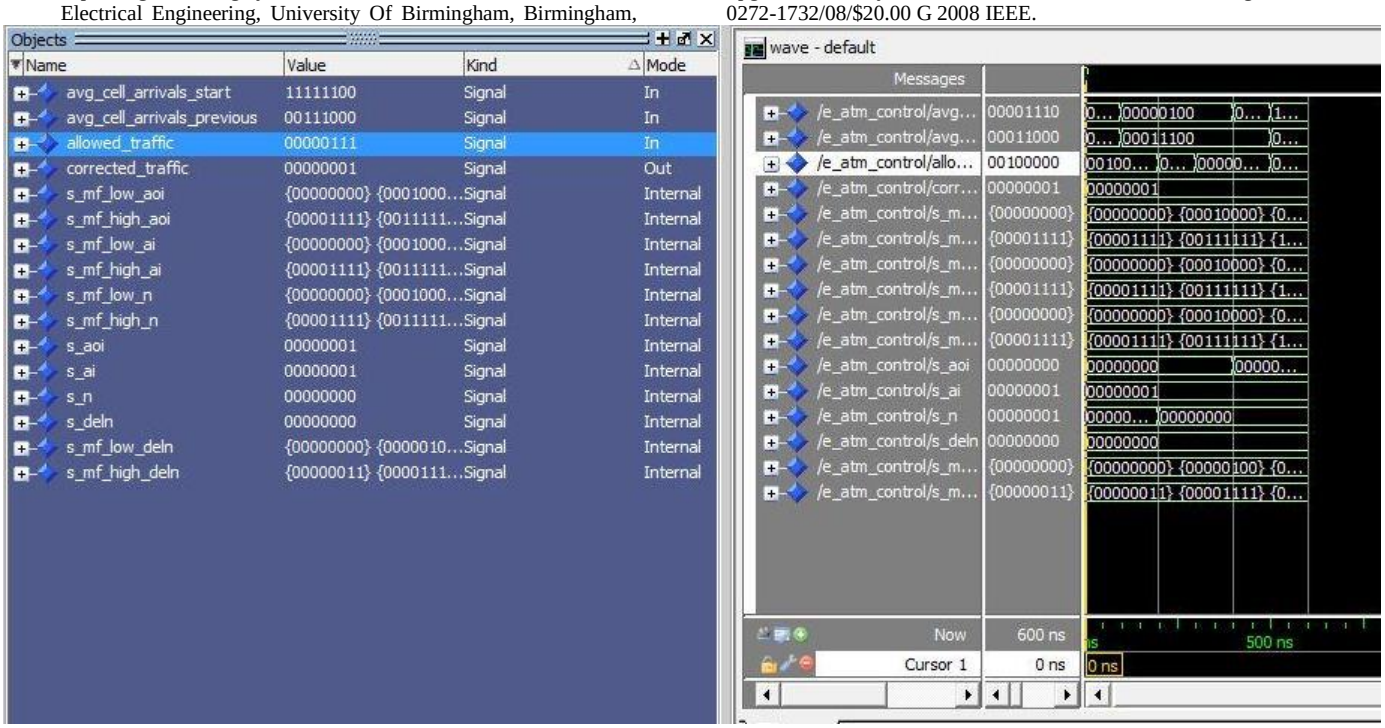

Fig. 2 Simulation result

