DEVELOPMENT OF AN FPGA BASED SMART DIAGNOSTIC SYSTEM FOR SPIROMETRIC DATA PROCESSING APPLICATIONS

Shubhajit Roy Chowdhury[1], Dipankar Chakrabarti[2], Hiranmay Saha[3],
[1,3]IC Design and Fabrication Centre,
Department of Electronics and Telecommunication Engineering,
Jadavpur University, Kolkata-700032.
[2]Om HealthNet Telemedicine Private Limited, Kolkata-700055
Email: [1]shubhajit@juicentre.res.in, [2]dc_ohnet@gmail.com, [3]hsaha@juiccentre.res.in

Abstract- The paper describes the development of an FPGA based fuzzy processing system for pulmonary spirometry applications predicting the approaching obstructive or restrictive pulmonary disorder of the patient before criticality actually occurs. The system employs a smart agent that accepts the Peak Expiratory Flow Rate (PEFR), Forced Expiratory Volume in 1 second (FEV1) and Forced Vital Capacity (FVC) data of patients. In order to speed up the computation process, hybrid parallel data processing architectures with dynamic scheduling mechanism have been employed leading to a speed up of approximately 12 times. The processor implemented on the FPGA can perform fuzzy inferencing at a speed of approximately 5.0 MFLIPS. The whole system is realized on Altera Cyclone EP1K6Q240C8 FPGA chip requiring 5,865 logic blocks. The system has been designed to be inexpensive, portable and user friendly for occupational health care applications in developing countries. Using the system, approaching pulmonary disorder of patients has been predicted with an accuracy of 95.83%.

Index Terms: Spirometry, fuzzy processor, hybrid parallel data processing architectures, smart agent

I. INTRODUCTION

Spirometry is a method of measuring various lung volumes and airflow rates in and out of lungs. It is a very effective method of detecting and following up various lung disorders. Spirometry started with very simple mechanical devices and gradually some very complicated and sophisticated equipment came into the market, which require specialized units as well as trained physicians. But, simple and inexpensive spirometers are not capable of computing long list of spirometric parameters. However, the usage of such spirometers can be boosted up even in absence of physicians, by making use of smart agent based diagnostic processing system, that uses approximate reasoning techniques to prognosticate the approaching critical pulmonary condition of a patient at an early stage.

Occupational health hazards involving respiratory system is a grave concern in modern world. Pulmonary function studies [1-6] can show restrictive, obstructive, or mixed patterns and range from normal to severe impairment. Spirometry is becoming more and more relevant for the general population with rising threat of environmental pollution. Pulmonary function tests (PFTs) objectively quantify lung function and impairment, and are used to evaluate persons with chronic lung disease. Spirometry is a mandatory procedure in these situations for early detection of various lung disorders and studying their prognoses.

Wide usage of spirometry in periphery in the third world is prevented by introduction of highly sophisticated spirometers, which are not only costly, but also require special expertise in usage and interpretation of its complicated results. To promote wide usage of spirometry in a meaningful and effective way in the developing and underdeveloped countries, use of simple spirometers – specially the mechanical one may need be encouraged and spirometric interpretations should be made easier. Furthermore, the need of the hour, especially for prevention of occupational health hazards affecting respiratory system, is to design instrument which can help the physicians as well as any person, in interpreting the present and the future of a worker / patient. Early detection of respiratory tract disorder in normal workers and of deterioration of functional status of their lungs in affected individual can prevent precipitation of catastrophic pulmonary disorder causing morbidity and, even mortality and thus can elevate the health status of the workers in hazardous industries of our country.

The current work aims at developing an FPGA based smart processing system that can accept spirometric data and can indicate the current pulmonary status of the patient. The system accepts the Peak Expiratory Flow Rate (PEFR), Forced Expiratory Volume in 1 second (FEV1) and Forced Vital Capacity (FVC) data of patients. The system employs a smart agent based on fuzzy logic that can predict the approaching obstructive or restrictive pulmonary disorder of the patient before criticality actually occurs. Fuzzy logic technology provides a simple way to arrive at a definite conclusion from vague, ambiguous, imprecise and noisy data (as found in medical data) using linguistic variables that are not necessarily precise. In order to achieve this, a study of a knowledge base system for the management of diseases was undertaken.

A number of works have been carried out in the development for hardware for fuzzy systems. The pioneer in this investigation field was Togai and Watanabe [7], who proposed the first hardware for fuzzy logic. Lim and Takefuji first proposed of implementing fuzzy rule based systems on silicon [8]. A large number of contemporary fuzzy systems use general purpose processors to process fuzzy inferences [9, 10]. Although, the solution is a highly flexible one, it is only appropriate where the inference speeds required are not excessively high, i,e. in the order of a few KFLIPS. The performance can be improved by suitable algorithmic modification [11]. However, with very high values this approach is quite inadequate, especially in high performance applications.

In high performance fuzzy systems, therefore, it is necessary to use hardware architecture capable of performing fuzzy computations [12, 13]. Manzoul and Tayal in [14], and Jaramillo-Botero and Miyake in [15] proposed the implementation of high speed fuzzy controllers using multiprocessor based parallel computing architectures. Multiple processors entail for costly solution of the system. The cost of multiple chips can be minimized by using single chip multiprocessor architectures (MP SoC). Using a multiprocessor system-on-chip architecture is a crucial step to optimize performance, energy and memory constraints at the same time, as has been reported by Orsila et al in [16]. Aranguren proposed the implementation of a pipelined fuzzy processor with a single rule unit [17]. Samoladas and Petrou propose two parallel architectures in SIMD mode [18]. Salvador presents a systolic architecture of 4 antecedents and 70 rules for consequences [19]. Although such a design is not very flexible as in the case of general purpose processors, it gives better results in terms of performance. The benefit of hardware implementation of a processor is also explained in [20] by Raychev, Mtibaa and Abid. However, the design and implementation flexibility is considerably improved by mapping the design into reconfigurable architectures like FPGA. The authors in their previous works [21, 22, 23] have also developed efficient architectures for realizing fuzzy processors.

In order to speed up the computation process, hybrid parallel data processing architectures have been employed leading to a speed up of approximately times. The current work comprises of VHDL modeling of the smart processing system and its FPGA based hardware realization. The present work aims at defining the architecture of a processor, which fully exploits the parallelism inherent in the fuzzy inferences. The processor also exploits the fact that only parts of the rules have a positive degree of validation, to reduce the number of rules processed which directly follows from the way the rules are stored. The method presented here considerably reduces the amount of memory required and simplifies the process of detection of active rules. The processor is implemented on an FPGA. The implementation of the processor on a reconfigurable architecture makes it suitable for further modification of functional logic of the processor with minimum programming effort.

The paper is organized as follows. Section 2 focuses on the occupational health hazards affecting respiratory system. Section 3 discuses on the principle of pulmonary spirometry. Section 4 illustrates the methodology used for fuzzification, inferencing and defuzzification of patients' data. The architectural design of the smart processor which forms the core of the proposed computing equipment has been explained in section 5. Section 6 describes the FPGA based implementation of the system. The detailed analysis of results obtained has been given in section 7.

II. OCCUPATIONAL HEALTH HAZARDS AFFECTING RESPIRATORY SYSTEM

Occupational health (often termed occupational and environmental health) is a very important aspect of modern medicine. Awareness of physicians, employees, employers and general population about these conditions is an international issue in order to reduce morbidity and, even, mortality. Aerosols and gases / vapors are the offenders.

Aerosols, which include both dusts and mists, are inhaled by subjects. They get deposited in respiratory tract – in airways and / or lung parenchyma by gravitational settlement (as in larger airways), and / or by inertial impaction (as by change of direction of flow in nose), and / or by interception (as for asbestos fibers which bypass larger airways, but are caught at bronchioles – specially at their bifurcations), and / or by diffusion (as for very small particles bombarded in atria or alveoli). Particles of the range of 20 microns and some of 5 microns are filtered in nose. Smaller particles, particularly in the range of 1 to 7 microns are deposited in lungs, while bulk of those in the range of 0.5 to 1 micron is exhaled again. Dusts may be inert, without any effect causing health hazards. These are nuisance dusts.

Many factors influence deposition of particles. The most important of them is respiratory minute volume, which is the product of tidal volume and rate of respiration per minute. This is increased, increasing the risk, during exertion. Heavier particles and insoluble ones are usually deposited in larger airways. As these airways are lined by ciliated columnar epithelial and mucus secreting cells, those particles are trapped by mucus and are swept upwards for being coughed up. Smaller particles, which are deposited beyond ciliated epithelium lined airways or in lung parenchyma, tend to deposit for long time or permanently, unless they are trapped by macrophages and are drained to lymphatic system or to proximal airways.

Vapors (gaseous state of matters below their boiling points) and gases can cause damage by asphyxiation (e.g., methane, nitrogen, carbon monoxide, carbon dioxide, hydrogen cyanide,hydrogen sulphide), and / or local irritation of airways and lungs (e.g., methylene chloride, various chloroethanes and chloroethylenes), and / or tissue injury after diffusion into blood through alveoli.

Pneumoconiosis is the general term used for all such non-neoplastic conditions affecting pulmonic tissues. Inflammatory conditions, viz., rhinitis, chronic bronchitis, bronchiolitis, etc. are effects on airways. Asthma is a group of condition, which requires special mention. It is caused by inflammation of lining mucosa of terminal airways and by intermittent spasm of smooth muscles encircling them. Hypersensitivity is main feature in these conditions, caused by local sensitization and degranulation of mast cells followed by release of triggering chemical. Malignancy, like bronchogenic carcinoma (of lungs), mesothelioma (of pleura) are other ominous examples of occupational airway diseases. Systemic effects are caused by ultrafine particles or toxic gases / vapours cause extra-pulmonary effects.

III. PRINCIPLE OF SPIROMETRY

Only three parameters are taken out of the long list of spirometric parameters available in the sophisticated spirometers. Simple and inexpensive mechanical spirometers can be used for these parameters in field work in the periphery or in small scale hazardous industries. These three chosen parameters are :

a) Forced Expiratory Volume in One Second : $\boldsymbol{FEV_1}$,
b) Forced Vital Capacity : ***FVC***, and
c) Peak Expiratory Flow Rate : ***PEFR***.

Standard equations are used to calculate the normal values of FEV_1, FVC and PEFR from inputs of age, sex, weight and height of subjects. Lowest ranges are also computed by subtracting 2 Residual Standard Deviation.

The equations used are :

1. PEFR :

For 10 – 18 years :

PEFR = - 617.5 + 9.7 x Age + 4.78 x Height + 2.66 x Weight……….…(1)

For 19 – 60 years :

Males : PEFR = 2.924 x Age + 3.38 x Height…………..………..(2)

Females : PEFR = 2.8 x Age + 3.05 x Height……………………….(3)

2. FEV_1 :

Below 18 years :

Males: $(0.812 \times \text{Height})^{(2.77)}$……………………...……………....(4)

Females: $(0.788 \times \text{Height})^{2.73}$……………………………………….(5)

Height in meters.

18 years and above :

Males: 0.04071 x Height – 0.02147 x Age – 2.59946 …………...(6)

Female: 0.04071 x Height – 0.02147 x Age – 2.56958………..….(7)

Height in centimeters.’

3. FVC :

Below 18 years :

Males: $(1.004 \times \text{Height})^{2.72}$……..………………………………..(8)

Females: $(0.946 \times \text{Height})^{2.61}$……………………………………….(9)

Height in meters.

18 years and above:

Males: 0.06584 x Height – 0.02954 x Age – 5.12451…………(10)

Female: 0.05557 x Height – 0.00793 x Age – 4.89036……….…(11)

Height in centimeters.

4. Lowest Range of Normal=Normal Value – 2X Residual Standard Deviation……………………………………………………………………………………………(12)

Each test value of a subject is compared to the normal values and their lowest limits to find out whether any of the three parameters is / are worse than the respective lowest range.

It is to be noted that the reference value formulae used are mostly relevant to Indian population. Ethnicity is an important criterion in finding the reference values. This is largely dependant on the body shape, height, weight etc. European Respiratory Society has recommended a reduction of Caucasian reference values by 10% for North Indians and by 12 to 13% for South Indians.

Computed normal values and the test values of the subjects are used to compute the percentage predicted values of FVC, PEFR and FEV_1/FVC.

$$\text{Percentage Predicted Value} = \frac{\text{Test Value}}{\text{Predicted Normal Value}} \times 100 \quad \ldots\ldots\ldots(13)$$

The current lung condition of patients is inferred based on the percentage predicted values of the spirometric parameters. Table 1 shows the inferences based on the percentage predicted values.

Table 1. Inferences on the current lung condition of patients is inferred based on the percentage predicted values of the spirometric parameters

PEFR (% predicted)	**FEV_1/FVC (% predicted)**	**FVC (% predicted)**	**Inference**
>70	>80	65	Normal
<70	>85		Mild Obstruction
<70	>65 but <85		Moderate

			Obstruction
>70	>90	>65	Mild Restriction
>70	>90	>45 but <65	Moderate Restriction
>70	>90	<45	Severe Restriction
<70	<90	<80	Mixed Cases

Combination of these values, give important inferences, viz., identify subjects with normal lung volumes; type of lung disorder – obstructive type or restrictive type; level / site of lung disorders – indicating outline of provisional diagnoses; and severity of lung disorder – mild, moderate or severe. An important group of chronic asthma patients, who, otherwise, fall within normal spirometric group, can be identified through the fuzzy logic used in the equipment under discussion.

Over and above the aforesaid diagnostic functions, the equipment under discussion can store data of subjects and can compare subsequent data of the said subjects to assess their prognoses. It can alarm about deterioration of the subject(s) in question and also can alert about impending catastrophe. It goes beyond saying that this feature adds value to the designed system extending its usability not only to prevention of occupational health hazards, but also to the well-being of common people in the face of rising environmental pollution.

In fact, for occupational health, follow up statistics of spirometric data are more important that a single set of test values. Most important aspect in occupational health is research. Unexposed workers of a particular industry / factory should be utilized for finding out an 'internal control' for the said industry / factory. This allows derivation of the risk factor involved in a particular department / section of the industry. A baseline of essential spirometric parameters should be set for each industry / factory before employment. Comparison of test values with this set of 'internal' reference as well as the calculated or published set of reference values is important to eliminate the possibility of extra-occupational effect on respiratory tract.

IV. FUZZIFICATION OF PATIENTS' DATA

Since doctors are more interested in knowing whether the pathophysiological risk parameters of patients are high, moderate or low, and also the trend of physiological parameters of patients, it would be more useful, to represent the pathophysiological risk parameters of patients as linguistic variable rather than ordinary variable and use fuzzy logic to build a predictive model, to predict the fuzzy set (low, moderate or high) in which the particular risk parameter of the patient is to lie in the next reading of patient data. For this purpose, triangular and trapezoidal fuzzy operators have been used.

The fuzzy logic of the diagnosis of lung diseases involves fuzzification, inference and defuzzification. The basic flow of information of the fuzzy logic mechanism is illustrated in Figure 1.

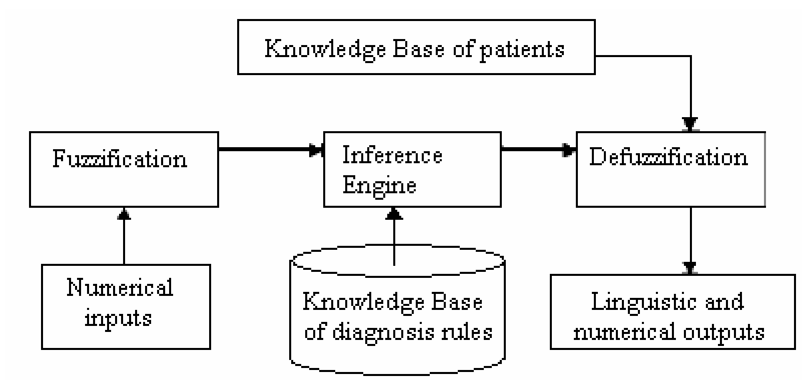
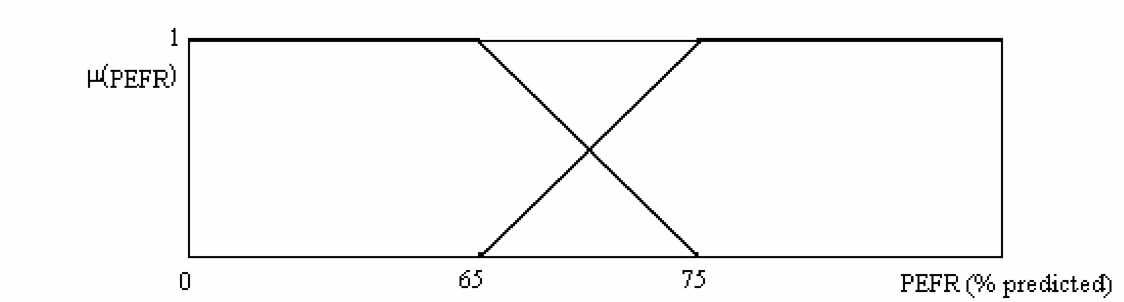


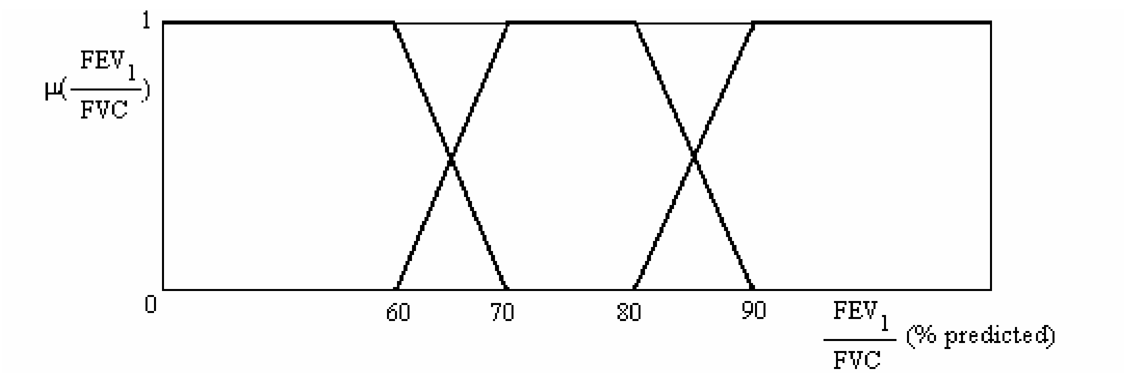
Figure 1. Empirical model of fuzzy logic of diagnosis of lung disease

The physiological variables viz. PEFR, FEV_1 and FVC that are computed from height, weight, age and sex data are fuzzified in cognitive frame of reference and the numerical values are transformed into linguistic values suitable for approximate reasoning.

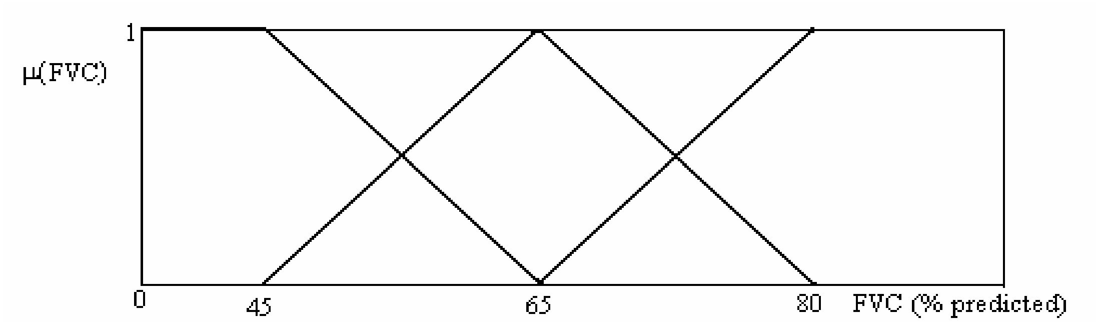
Figure 2 depicts the cognitive frames used for fuzzy modeling PEFR, FEV_1/FVC and FVC data of patients.



(a)



(b)



(c)

Figure 2. Plots of the membership functions:

(a) PEFR (b) FEV_1/FVC (c) FVC

In figure 2 (a), the membership function plot of PEFR indicates two fuzzy sets, the left hand side trapezoid indicating the low fuzzy set and the right hand trapezoid indicating the high fuzzy set. In figures 2(b) and (c), the membership function plots of FEV_1/FVC and FVC indicates three fuzzy sets – low, moderate and high respectively.

The algorithm for diagnosis computes the time- weighted mean of the membership functions of the patient's pathophysiological data. The possibility that the next pathophysiological data will low or moderate or high (for PEFR only low and high) is computed as:

$$P_R(x) = \frac{\sum_{i=1}^{n} i\mu(x)}{\sum_{i=1}^{n} i} \quad \text{..(14)}$$

where the summation is done from i=1 to n, and the value of n is the sequence number of the time instant at which the current pathological data of the patient is taken and R ϵ {low, moderate, high}for FEV_1/FVC and FVC and R ϵ {low, high} for PEFR. $\mu(x)$ is $\mu l(x)$, $\mu m(x)$ or $\mu h(x)$ accordingly as the membership function concerned refers to low, moderate or high fuzzy set respectively. For predicting the fuzzy set in which the next state input of a certain pathophysiological parameter is going to lie, the value of P(x) is considered for which P(x) ≥PR(x).

The knowledge base of patients store information about pathophysiological state of patients at different instants of time. The inference engine infers about the current physiological condition of the patient and predicts whether the patient is approaching a critical condition or not. Inferencing involves giving a decision whether the patient is in normal condition or heading towards a moderately critical condition or a severely critical condition. Inferencing is done by taking the possible next state output of the diagnostic algorithm at different instants of time. Typical rules for inferencing are:

R1: if ($PEFR$ is *high* and $\left(\frac{FEV_1}{FVC}\right)$ is *high* and FVC is *moderate*) then condition is *normal*

R2: if ($PEFR$ is *low* and $\left(\frac{FEV_1}{FVC}\right)$ is *high*) then condition is *mild obstruction*

R3: if ($PEFR$ is *low* and $\left(\frac{FEV_1}{FVC}\right)$ is *moderate*) then condition is *moderate obstruction*

R4: if ($PEFR$ is *low* and $\left(\frac{FEV_1}{FVC}\right)$ is *low*) then condition is *severe obstruction*

R5: if ($PEFR$ is *high* and $\left(\frac{FEV_1}{FVC}\right)$ is *high* and FVC is *high*) then condition is *mild restriction*

R6: if ($PEFR$ is *high* and $\left(\frac{FEV_1}{FVC}\right)$ is *high* and FVC is *moderate*) then condition is *moderate restriction*

R7: if ($PEFR$ is *high* and $\left(\frac{FEV_1}{FVC}\right)$ is *high* and FVC is *low*) then condition is *severe restriction*

R8: if ($PEFR$ is *low* and $\left(\frac{FEV_1}{FVC}\right)$ is *high* or $\left(\frac{FEV_1}{FVC}\right)$ is *moderate* and FVC is *high*) then condition is *mixed cases*

and so on.

V. ARCHITECTURAL DESIGN OF THE FUZZY PROCESSOR

The architectural design of the processor is shown in figure 3.

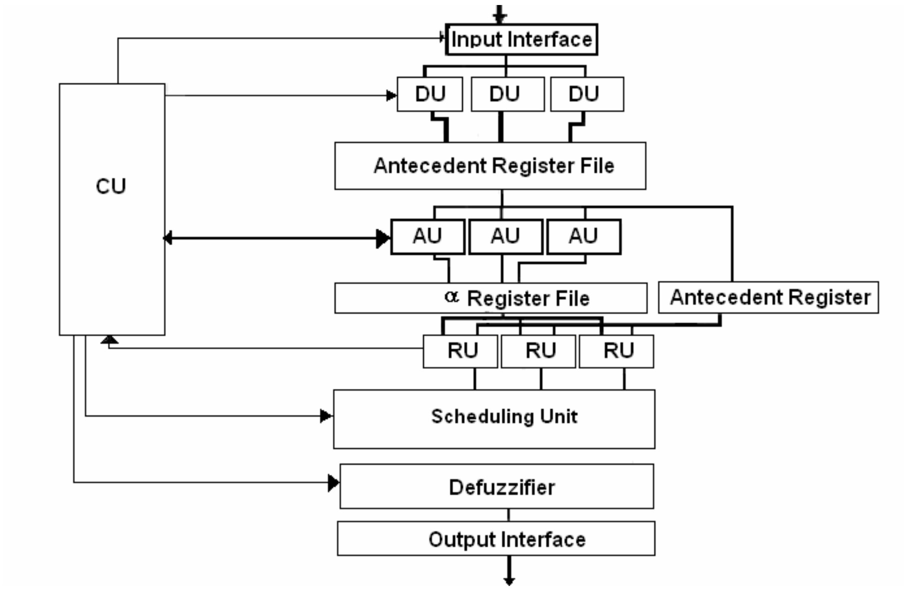


Figure 3. Architectural design of the fuzzy processor

It comprises of the following: an input interface, an output interface, 3 detection units (DU), 3 antecedent units (AU), 3 rule units (RU), a defuzzifier, a scheduling unit and a controller (CU). The input interface acquires the fuzzy sets of inputs. The DU have the task of detecting antecedents with a positive degree of truth. The AU calculates the degree of truth of positive antecedents. The RU calculates the degree of activation of active rules alone. The scheduling unit schedules the active rules to be used for defuzzification. The defuzzifier defuzzifies the output using Yager levels [24] methods. The CU issues control signals to coordinate the activities of other units.

Information about the fuzzy sets with non null degree of truth is stored in the antecedent register file. The DU, AU, RU and the defuzzifier are pipelined. The proposed structure assumes that the maximum number of antecedents, n_A is 16. Each rule is made up of one consequent. The maximum possible number of rules (n_R) that can be stored is 256. The maximum number of elements in the variable term set, n_T is 15. The universe of discourse is made up of d_U=256 values. The number of membership degrees n_L is 64. It is assumed that the fuzzy sets are convex. The following sub-sections focus on the architectural details of the different units.

A. The Detection Unit

The task of the detection unit is to detect antecedents with a positive degree of truth. This can be achieved by checking whether each antecedent has an intersection with its input. A necessary and sufficient condition to check this intersection is to see whether the respective supports of the fuzzy sets intersect. Detection starts as soon as the support of an input variable has been acquired.

Let F_{ik} be the k^{th} element of the term set of variable X_i ($i=1,2,\ldots,n_A$), it has with it an associated bit a_{ik}, which indicates whether the k^{th} fuzzy set F_{ik} of the term set of the input variable X_i has a non null intersection with the input fuzzy set X_i.

Since the fuzzy sets are assumed to be convex, hence the support of each fuzzy set F_{ik} is represented by a closed interval $[B_{ik}^L, B_{ik}^R]$. Thus a_{ik} is obtained as follows:

$$a_{ik} = \begin{cases} 0, IF\left[B_i^{L*}, B_i^{R*}\right] \cap \left[B_{ik}^{L}, B_{ik}^{R}\right] = \phi \\ 1, IF\left[B_i^{L*}, B_i^{R*}\right] \cap \left[B_{ik}^{L}, B_{ik}^{R}\right] \neq \phi \end{cases} \quad \ldots\ldots(15)$$

While the processor is detecting the antecedents with a positive degree of truth, it acquires the information concerning the fuzzy input. Figure 4 shows the block diagram of the detection unit (DU).

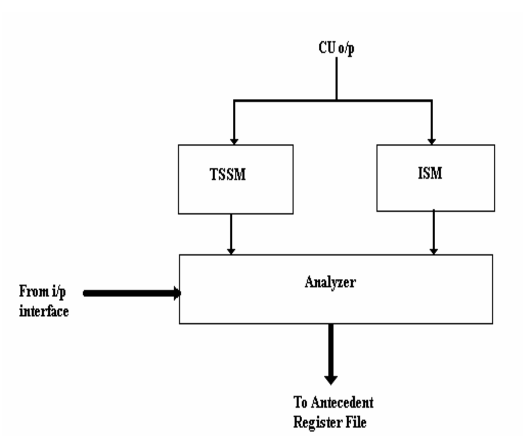


Figure 4. Architectural design of the detection unit (DU)

As evident from figure 2, the detection unit consists of one pipe stage. It comprises of a term-set support memory (TSSM) and an input support memory (ISM). The TSSM stores information concerning the support of the fuzzy sets of the term sets of the variables. As the fuzzy sets are convex, it is sufficient to store the pair of end points which delimit them. The TSSM therefore contains n_T*n_A words of 2*$\log_2 d_u$ bits. In the proposed architecture, n_T=15, n_A=2 and d_u=256. Therefore, TSSM contains thirty 16 bit words.

The ISM contains supports for the fuzzy inputs, so it contains n_A words of 2*$\log_2 d_u$ bits. Under the assumptions made, these modules contain eight 16 bit words. The analyzer block computes a_{ik} and stores the value in the antecedent register file.

B. The Antecedent Unit (AU)

The task of the antecedent unit is to calculate the degree of truth of antecedents which have a non null intersection with the respective inputs. The membership functions have been represented by means of triangular or trapezoidal functions.

Given any truth value α, if segment number ni which crosses this value and the abscissae x_{Lni} and x_{Rni} are known, the end points of the α cut are given by:

$$x_L(\alpha) = x_{L\min} + p_L * \alpha \quad (16)$$

$$x_R(\alpha) = x_{R\max} - p_R * \alpha \quad (17)$$

where $x_{L\min}$ refers to the abscissa of the leftmost point of the fuzzy set in its universe of discourse and $x_{R\max}$ refers to the abscissa of the rightmost point of the fuzzy set.

p_L and p_R are defined as follows:

$$p_L = x_{L\max} - x_{L\min} \quad (18)$$

$$p_R = x_{R\max} - x_{R\min} \quad (19)$$

The degree of truth of an antecedent is obtained by calculating the maximum value for which the α cut of the antecedents and inputs have a non null intersection. The calculation of the degree of truth can be separated into two phases: identification of segments of membership functions which have intersections and calculation of the intersection in the interval of the truth space containing these segments. These two phases are pipelined: while the segments which intersect for a given antecedent is found, the maximum of the intersection for a previous antecedent is calculated with the intention of doubling the throughput of the system.

Figure 5 shows the architectural design of the antecedent unit. It consists of four pipe stages. It consists of a segment calculator (SC), whose task is to identify the segments which intersect. The antecedent calculator (AC) calculates the maximum value of α for which there is an intersection. The antecedent memory (AM) and the input memory (IM) respectively store the fuzzy sets of the antecedents and the inputs. The two memories store in each word, the lower end point of the segments which define a fuzzy set. The width of each memory word is 16 bits. The AM contains 32 words whereas the IM contains 8 words.

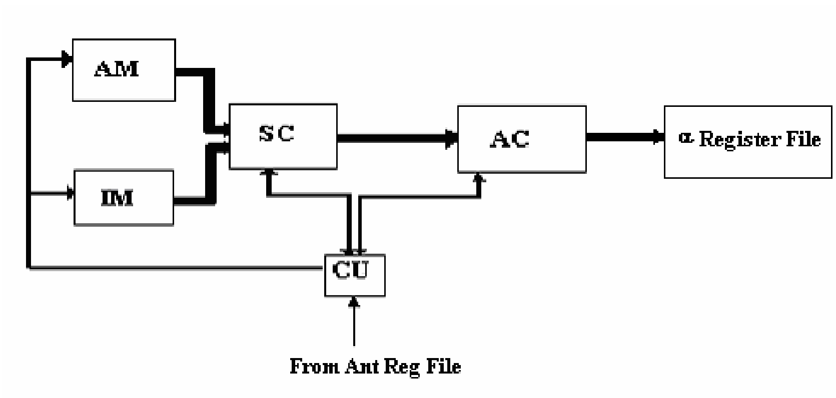


Figure 5. Architectural design of the Antecedent Unit

C. *Rule Unit (RU)*

The rule unit calculates the degree of activation of the rules. A typical fuzzy application consists of groups of rules that share the common antecedent. This suggests not storing all the antecedents but only the terms which vary; the constant terms for each group are stored only once. In the actual implementation, two separate memories are used. One contains constant terms and the number of rules in the group. The other contains the other antecedents in each rule. This leads to a huge saving in memory space for rule storage.

The rule memory can store a maximum of 512 rules. The number of memory locations can be sufficiently increased to store much more than 512 rules. There can be more than 512 rules in principle. However, for our particular application of medical of medical diagnosis, we do not require more than 512 locations in the rule memory. The proposed rule unit (RU) does not calculate the degree of activation for all the rules, but only for active ones. This is achieved in the following way: As a group features the presence of constant terms, if their degree of truth is null, it is useless processing the rules in this group. The unit can thus pass on to the rules in the next group. To calculate whether the constant terms of the rules in a group are active, it is sufficient to perform an AND operation between the associated a_{ik} bits. Only if the result is one, the group is processed. In this case, partial degree of activation obtained by calculating the minimum of the degrees of truth of the constant terms is only calculated once, stored in a register, and reused for all the rules in the group. Each rule is processed sequentially, so if there are terms with null degrees of truth among other antecedents, the RU suspends processing of the rule and goes on to another one. The architectural design of the rule unit is shown in figure 6.

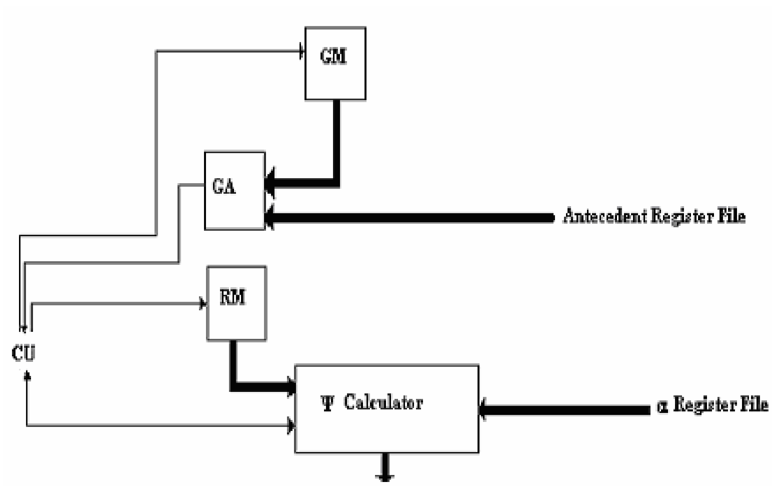


Figure 6. Architectural Design of the Rule Unit

The rule unit (RU) acquires the information of each group of rules from the group memory (GM). The GM contains 16 words. Each word contains 4 bits to indicate the variable and 4 bits to indicate the element of the term set and 4 bits to indicate the number of rules in the group. Before processing of the group starts, the group analyzer

(GA) checks whether the constant terms are active. If they are no active rules, the next group is read. If the checking yields positive results, the Ψ calculator calculates the degree of activation for the rules in the group.

D. Scheduling Unit:

The scheduling stage is to discard the non-active rules and, then, schedule the active rules if necessary. Figure 7 gives the logic diagram of the scheduling unit, which scans the weights of the rules processed in the rule unit.

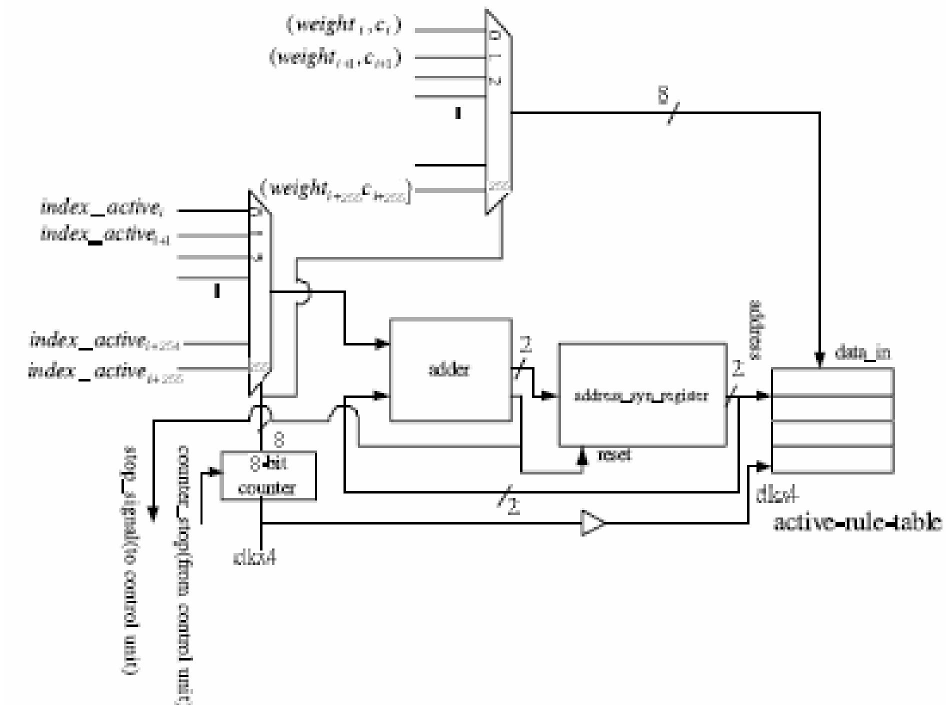


Figure 7. Block diagram of the scheduling unit.

For a rule R_i, $weight_i$ is the degree of activation of the rule. An active rule Ri can be identified by an OR gate, whose inputs are the bits of $weight_i$. We define the output bit of the OR gate as the active value $index\_active_i$. It is obvious that the rule R_i is active if and only if the active value $index\_active_i$ is 1. In each pipeline stage cycle, 255 active values are scanned sequentially by a 256-to-1 multiplexer. We design an active-rule-table to store four active rules. In the active-rule-table, each active rule R_i is represented by $weight_i$ and c_i, where c_i is the index of consequent membership function. A two-bit register $address\_syn\_register$ points to the first available address of the active-rule-table. In fact, this register also denotes the number of active rules. In every clock cycle of the clock signal *clkx4*, the rule *Ri* is stored into the first available address, and the register $address\_syn\_register$ is added with the active value $index\_active_i$. When a carry-out of the addition occurs, the active-rule-table is full. If the active-rule-table is full but a new active rule is found, the dynamic scheduling mechanism is invoked. The basic idea of our dynamic scheduling mechanism is to stop both the Ψ calculator and new active rule insertion in the remaining time of the present pipeline stage cycle. Therefore, the *stop* signal is generated when one of the following two conditions occurs: (1) the active-rule-table is full but a new active rule is found; or (2) the 8-bit counter counts up to 0 (i.e., from 11111111 to 00000000). The *stop* signal is cleared at the beginning of every pipeline stage cycle.

E. Defuzzifier

The defuzzifier performs the following computation:

$$z_0 = \frac{\sum_{i=1}^{N} \Psi_i X_i}{\sum_{i=1}^{N} \Psi_i} \quad \text{(20)},$$

where z_0 is the defuzzified output of the fuzzy processor. A multiplier performs the task of computing $\Sigma_i \Psi_i x_i$ and $\Sigma_i \Psi_i$ for each output variable. The Each unit receives the Ψ_i value from the Ψ calculator. The divider is implemented using the available dividers of the FPGA chip. The basic reason of using Yager's defuzzification method instead of the more commonly used Mamdani's method is evident from equation (20). For defuzzification using *N* active rules, we need *2N* additions, *N* multiplications and one division.

On the other hand, using Mamdani's method with centre of area procedure, we have,

$$z_0 = \frac{\sum_{i=1}^{m} z\mu_{C'}(z)\Delta z}{\sum_{i=1}^{m} \mu_{C'}(z)\Delta z} \quad \ldots\ldots(21),$$

where z_0 is the defuzzified output of the fuzzy processor, Δz is the discretization value for universe Z, m is the number of discretization steps and $C' = \bigcup_{i=1}^{N} C_i$ [24]. Therefore, using Mamdani's method would require a total of $2m$ additions, $3m$ product operations and one division. It is reasonable to assume that $m>N$, and so it is easy to see that Yager's method is more efficient as far as computational requirements are concerned. In particular if z_i is the symmetric axis, then $m_i=z_i$ for Yager's method.

F. The Control Unit

The control unit is modeled as a hardwired control unit. The implementation of the control unit is done by considering the controller as a finite state machine. The state diagram of the control unit is shown in figure 8.

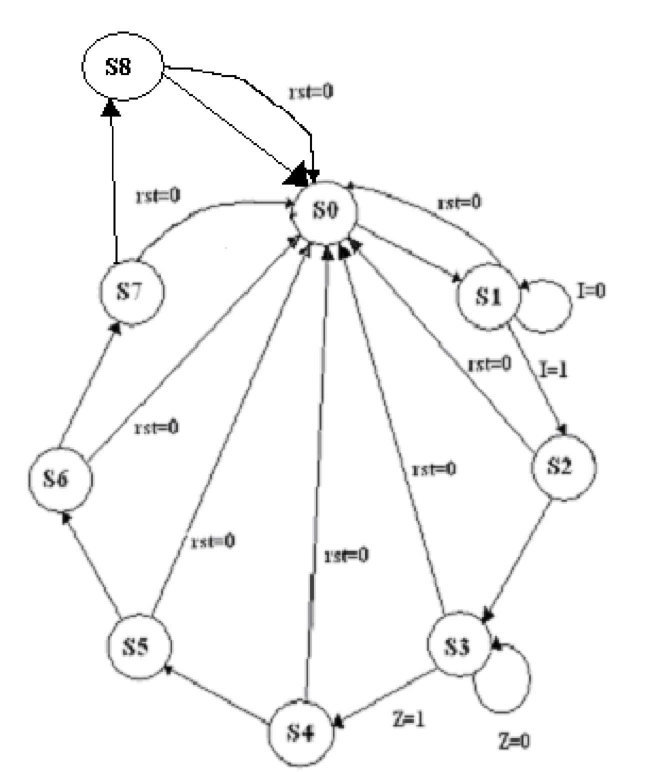


Figure 8. State diagram of the controller

Initially the controller starts from state S0. At state S0, the fuzzy inputs from the input interface are received by the detection unit and a_{ik} values are stored in the antecedent register file. The controller then switches to state S1. In this state, the controller provides control inputs to the segment calculator to iteratively search the interval containing α. A new set of fuzzy inputs is also accepted from the input interface, since the entire architecture is pipelined. So long the search continues, a single bit output of the segment calculator viz. intersection tester I is zero and the pipelined is stalled. When I become equal to 1 at the end of the search process, the controller switches to state S2. In state S2, the maximum value of α is determined for which the sets of antecedents and inputs have a non-null intersection. When the controller is in state S3, the output of the α register file is enabled and selected by the Ψ calculator block, which calculates the minimum between the truth value of the current antecedent and the current degree of activation. The Ψ calculator also checks whether the result of the minimum operation is null. If null, a control signal is sent to the control unit, which suspends the calculation and goes on to the next rule. The final degree of activation is stored in the Ψ register within the Ψ calculator. In state S4, the mid point $x_M(\Psi_i)$ of the Ψ_i level set is calculated. Next the controller switches to state S5 in which the scheduler schedules the active rules. In state S6, the multiplier of the defuzzifier calculates $\Psi_i X_i$. In state S7, summation $\Sigma_i \Psi_i X_i$ is performed by the adder block within the defuzzifier. In state S8, the divider is enabled by the controller.

VI. FPGA BASED IMPLEMENTATION OF THE PROCESSOR

The processor is implemented on a Altera Cyclone family EP1C6Q240C8 device. The whole system is realized by configuring the FPGA as a processor which handles the task of computing and interfacing the peripherals. The LED 7 segment displays are driven by BC 545 PNP transistors. The 7 segment displays indicate the possibilities of low, moderate and high values of the different pathological parameters at the next physiological state of the patient. Since, there are four seven segment displays for output in the final system, and there is only one port available for display, hence a four bit output called SCAN (0 to 3) is used. Actually the different bit lines of the SCAN are connected to cathodes of different common cathode LED 7 segment displays so as to select the 7 segment LED in time shared mode. The display codes corresponding to the 7-segment display has been stored in a ROM. The system can be reset at any point in time by a reset input which has been implemented using a push button switch. The FPGA receives its configuration information from a EPCS1 configuration PROM chip. The whole system receives its clock signals from a piezoelectric oscillator. Figure 9 shows the picture of the instrument.

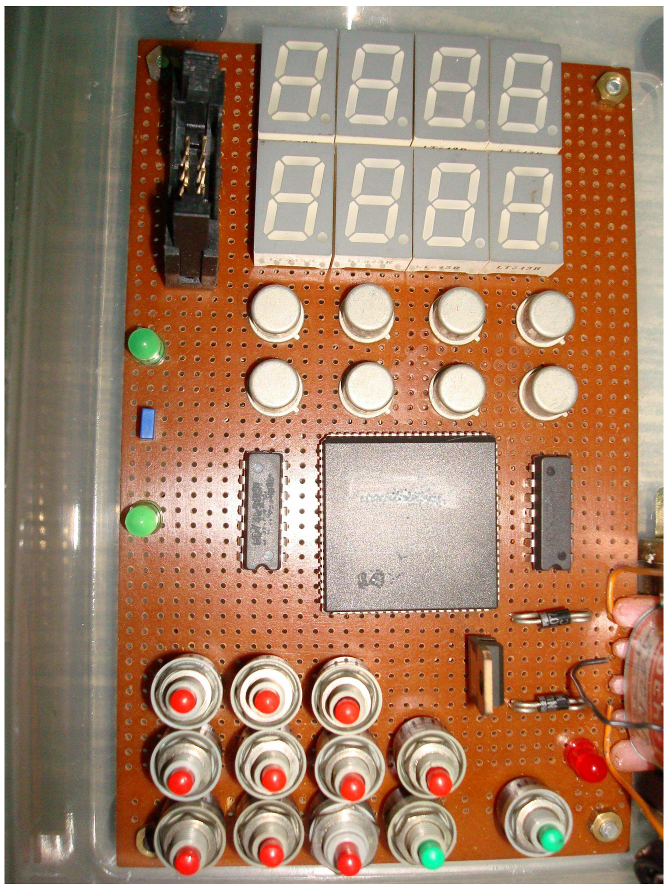


Figure 9. Picture of the FPGA based system on board

The technology schematic of the processor as realized on the FPGA is shown in figure 10.

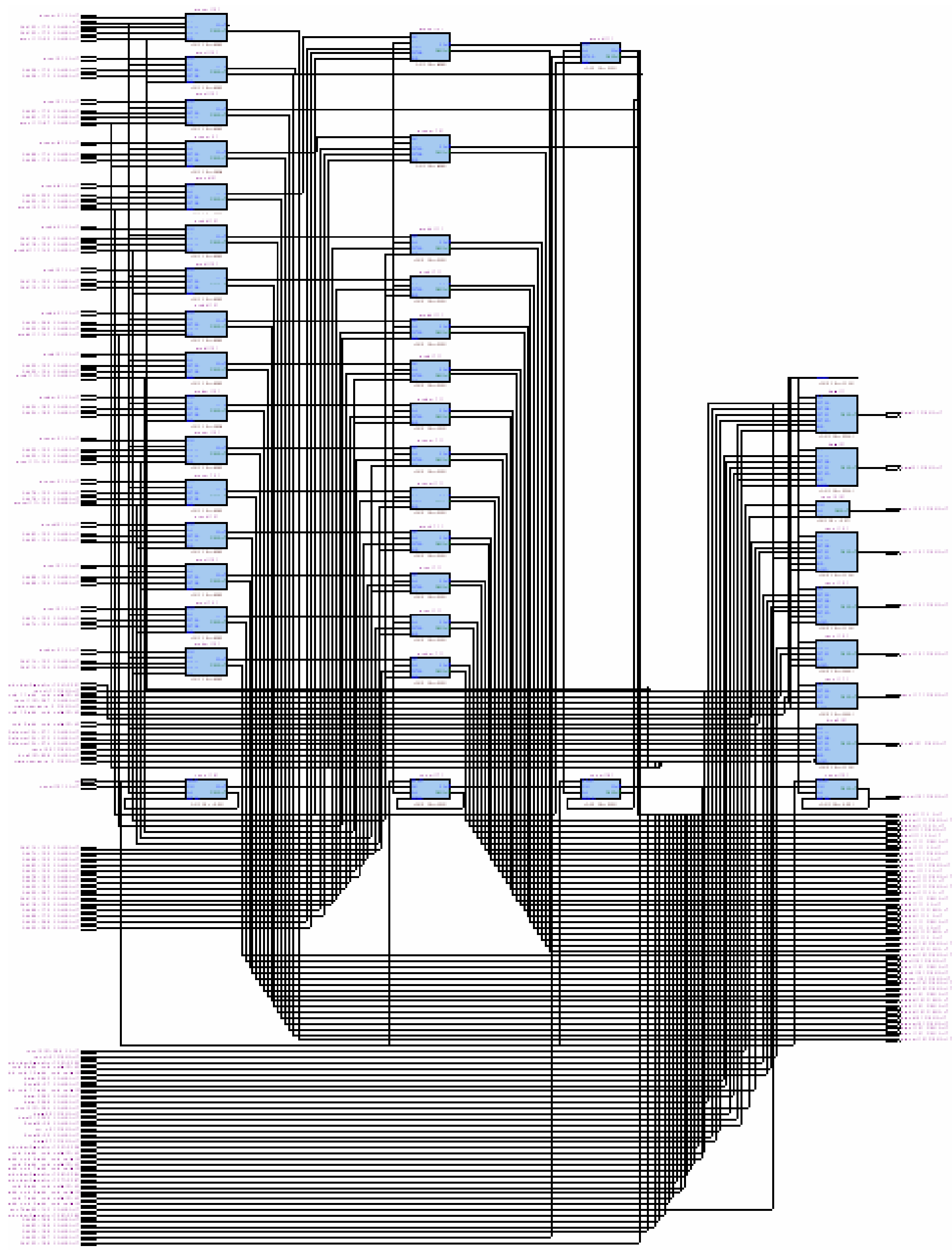


Figure 10. Technology schematic of the processor

The technology schematic shows how the different LABs in the FPGA have been occupied to realize the processor. The resource utilization summary of the processor on the FPGA chip is given in Table 2.

Table 2. Resource Utilization Summary

Resource	**Usage**
Logic Cells	5,865 / 5,980 (98 %)
Registers	3,235 / 6,523 (50 %)
Total LABs	579 / 598 (97 %)
Logic elements in carry chains	2353
User inserted logic cells	0
I/O pins	27 / 185 (14 %)
Virtual pins	0
Clock pins	1 / 2 (50 %)
Global signals	2
M4Ks	15 / 20 (75 %)
Total memory bits	4,096 / 92,160 (4 %)
Total RAM block bits	46,084 / 92,160 (50 %)
Global clocks	2 / 8 (25 %)
Maximum fan-out node	Clk
Maximum fan-out	1047
Total fan-out	17700
Average fan-out	3.32

The resource utilization summary indicates that there is a very small number of logic cells and logic array blocks (LABs) left unused during implementation of the processor on the FPGA. This implies a minimum wastage of silicon area while implementing the processor on the FPGA. The floorplan of the processor as realized on the FPGA is shown in figure 11:

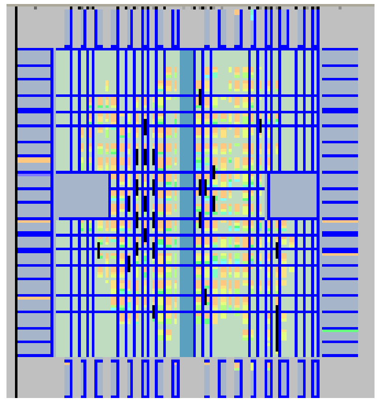


Figure 11. The floorplan of the processor

The colored cells correspond to the logic array blocks on the FPGA chip that have been allocated to the processor.

VII. RESULTS AND DISCUSSIONS

The system has been applied to determine the current lung condition of the patient as well as predicting in advance, the approaching critical lung condition of the patient. The height, weight, age and sex information is given as input to the system, from which the system calculates the normal predicted values of PEFR, $\frac{FEV_1}{FVC}$ and FVC. The instantaneous values of PEFR, FEV1 and FVC are given as inputs to the system.

7.1 Analysis of current lung condition of patients:

The system has been tested with the data of 10 patients to test its applicability in determining the current lung condition of the patient. The analysis of data of 10 patients from Om Health-net Telemedicine Pvt. Ltd. is shown in Table 3.

Table 3. Analysis of current lung data of patients

Sl. No.	Patient Name	PEFR (%)	μ(PEFR)		$\frac{FEV_1}{FVC}(\%)$	$\mu(\frac{FEV_1}{FVC})$		FVC (%)	μ(FVC)		Instrumental Decision	Physicians' Decision
1.	Chhaya Rani Ganguly (66)	51	μ_L	1.00	95	μ_L	0.00	56	μ_L	0.45	Moderate Restriction	Moderate Restriction
			μ_H	0.00		μ_M	0.00		μ_M	0.55		

						μ_H	1.00		μ_H	0.00		
2.	Raniprava Chowdhury (65)	46	μ_L	1.00	120.64	μ_L	0.00	58	μ_L	0.35	Moderate Restriction	Moderate Restriction
			μ_H	0.00		μ_M	0.00		μ_M	0.65		
						μ_H	1.00		μ_H	0.00		
3.	Bimal Chandra Majumdar (63)	60.57	μ_L	0.54	94.36	μ_L	0.00	70.88	μ_L	0.00	Mixed Cases	Mixed Cases
			μ_H	0.46		μ_M	0.00		μ_M	0.61		
						μ_H	1.00		μ_H	0.39		
4.	Rajat Narayan Saha (30)	87.78	μ_L	0.00	98.79	μ_L	0.00	101.11	μ_L	0.00	Severe Restriction	Severe Restriction
			μ_H	1.00		μ_M	0.00		μ_M	0.00		
						μ_H	1.00		μ_H	1.00		
5.	Snigdha Pal (44)	65.5	μ_L	0.95	119.79	μ_L	0.00	84.34	μ_L	0.00	Severe Restriction	Severe Restriction
			μ_H	0.05		μ_M	0.00		μ_M	0.00		
						μ_H	1.00		μ_H	1.00		
6.	Sibu Das (25)	34.41	μ_L	1.00	94.86	μ_L	0.00	79.23	μ_L	0.00	Mixed Cases	Mixed Cases
			μ_H	0.00		μ_M	0.00		μ_M	0.05		
						μ_H	1.00		μ_H	0.95		
7.	Bandana Sarkar (54)	89.29	μ_L	0.00	112.32	μ_L	0.00	67.86	μ_L	0.00	Mixed Cases	Mixed Cases
			μ_H	1.00		μ_M	0.00		μ_M	0.19		
						μ_H	1.00		μ_H	0.81		
8.	Arabindo Choudhury (16)	51	μ_L	1.00	64	μ_L	0.60	56	μ_L	0.45	Severe Obstruction	Severe Obstruction
			μ_H	0.00		μ_M	0.40		μ_M	0.55		
						μ_H	0.00		μ_H	0.00		
9.	Mannu Hela (40)	1.84	μ_L	1.00	66.84	μ_L	0.32	48.74	μ_L	0.81	Mild Obstruction	Mild Obstruction
			μ_H	0.00		μ_M	0.68		μ_M	0.19		
						μ_H	0.00		μ_H	0.00		
10.	Kaushik Singh (30)	64.13	μ_L	1.00	175.91	μ_L	0.00	85.71	μ_L	0.00	Severe Restriction	Severe Restriction
			μ_H	0.00		μ_M	0.00		μ_M	0.00		
						μ_H	1.00		μ_H	1.00		

In the above table μ_L, μ_M and μ_H refers to the membership function values of the pathophysiological parameters in the low, moderate and high fuzzy sets. Since only the current pathophysiological state is analyzed, hence, the possibility values discussed in section 4 is same as the membership function values. Using the rules of inferencing discussed in section 4, the system indicates the current lung condition of the patient. The figures within the parentheses in the patients' name column indicate the age. It is clear from table II that in all the ten cases the decision given by the smart computing system agrees with the diagnostic decision given by the physician.

7.2 Analysis of sequence of patient data collected over time:

Data of 120 patients of Aga Khan University, Karachi, Pakistan, has been analyzed to assess the accuracy of diagnosis by the smart agent in predicting the future pulmonary condition of the patient. The analysis of a patient of age 27 years, is shown in table 4.

In table 4, data has been taken at 5 days interval of time. T1 refers to the 0^{th} day and T5 refers to the 20^{th} day. PL, PM and PH refers to the possibility of low, moderate and high values of the pathophysiological parameters respectively. The essence of the system lies in that the system predicts a state of approaching pulmonary obstruction even when the lung condition of the patient is normal, which is later proved to be true. This also elucidates that such type of system when implemented in portable hardware can be deployed in telemedicine environments in rural areas, where the health care professionals often provide support services in absence of the physician.

7.3 Performance analysis of the fuzzy processor:

The advantage of having an FPGA based implementation of the smart diagnostic system has also been analyzed by comparing the performance of the implemented smart diagnostic system with a general purpose processor running the same algorithm. In order to compare the delay of computation by the hardware implementation with that of a software implementation, a sequential version of the algorithm is realized in C and studied on a general-purpose computer (Pentium IV processor 2.0 GHz running Fedora 5.0). The actual running time of the sequential algorithm have been via clock ticks using times( ) function in C. The CPU time for running the sequential algorithm is found to be 6 ms. On the other hand, with the FPGA based smart diagnostic system, the computing time using a single data set has been found to be equal to 0.2μs. This amounts to a speed up of 6000 using the smart system. The difference in delay of computation may stand out to be considerable when the system is redeployed for computationally intensive applications. Thus we have fast data processing architecture in the FPGA chip.

Table 4. Analysis of a typical patient data collected over five instants of time

PEFR (Normal predicted) = 9.68

FEV_1 (Normal predicted) = 3.01L

FVC (Normal predicted) = 3.68L

Time	PEFR	PEFR (%)	FEV_1(L)	FVC(L)	$\frac{FEV_1}{FVC}$(%)	FVC (%)	μ(PEFR)		$\mu(\frac{FEV_1}{FVC})$		μ(FVC)		P_{PEFR}		$P_{\frac{FEV_1}{FVC}}$		P_{FVC}		Instrumental Decision	Adhoc Decision by physician
T1	7.92	81	1.84	2.39	95.06	65.00	μ_L	0.00	μ_L	0.00	μ_L	0.00	P_L	0.00	P_L	0.00	P_L	0.00	Normal	Normal
							μ_H	1.00	μ_M	0.00	μ_M	1.00	P_H	1.00	P_M	0.00	P_M	1.00		
									μ_H	1.00	μ_H	0.00			P_H	1.00	P_H	0.00		
T2	7.98	82	1.81	2.31	96.29	62.77	μ_L	0.00	μ_L	0.00	μ_L	0.11	P_L	0.00	P_L	0.00	P_L	0.07	Normal	Normal
							μ_H	1.00	μ_M	0.00	μ_M	0.89	P_H	1.00	P_M	0.00	P_M	0.93		
									μ_H	1.00	μ_H	0.00			P_H	1.00	P_H	0.00		
T3	8.32	85.95	1.59	2.59	75.79	70.31	μ_L	0.00	μ_L	0.00	μ_L	0.00	P_L	0.00	P_L	0.00	P_L	0.04	Mild Obstruction	Normal
							μ_H	1.00	μ_M	1.00	μ_M	0.65	P_H	1.00	P_M	0.60	P_M	0.79		
									μ_H	0.00	μ_H	0.35			P_H	0.50	P_H	0.06		
T4	7.48	77.27	1.47	2.54	71.44	69.02	μ_L	0.00	μ_L	0.00	μ_L	0.00	P_L	0.00	P_L	0.00	P_L	0.02	Moderate Obstruction	Mild Obstruction
							μ_H	1.00	μ_M	1.00	μ_M	0.73	P_H	1.00	P_M	0.70	P_M	0.76		
									μ_H	0.00	μ_H	0.27			P_H	0.30	P_H	0.21		
T5	7.04	72.72	1.51	2.55	69.13	69.29	μ_L	0.23	μ_L	0.09	μ_L	0.00	P_L	0.08	P_L	0.03	P_L	0.01	Moderate Obstruction	Moderate Obstruction
							μ_H	0.77	μ_M	0.91	μ_M	0.71	P_H	0.92	P_M	0.77	P_M	0.75		
									μ_H	0.00	μ_H	0.29			P_H	0.20	P_H	0.24		

Since the processor is arranged as a cascade of pipelined stages, the slowest stage determines performance. The processor operates at a clock frequency of 40 MHz. This clock frequency is determined by the logic synthesis of the processor, modeled in VHDL and synthesized using Altera Quartus II synthesis tool. A higher frequency would have been chosen but it will add to the power dissipated by the processor. With a clock frequency of 40 MHz, the number of inferences processed per second is given by:

$$N_{INF} = \frac{1}{N_{cycles} * 25 * 10^{-9}}$$

If the number of active groups is 16 and the rules are uniformly distributed in the groups, the latency of the slowest stage (N_{cycles}) is 8 clock cycles. Therefore the performance obtained is 5.0 MFLIPs approximately.

7.4 Determination of accuracy of diagnosis by Bayesian Analysis:

Bayesian analysis has been carried out on the population of patients from Aga Khan University under study to estimate the reliability of the system. In order to estimate the reliability of diagnosis, the definitions of statistical terms used in [25] have been used. As follows from the application of Bayes' theorem, the predictive value of any diagnostic test is influenced by the prevalence among the tested population, and by the sensitivity and specificity of the test [21, 22, 23]. In our particular case, the total population under study was 120.

Let *a* be the number of patients where the diagnostic test yields a positive result and the patient really has a disease, *b* be the number of patients where the diagnostic test yields a positive result and the patient does not have a disease, *c* be the number of patients where the diagnostic test yields a negative result and the patient really has a disease and *d* be the number of patients where the diagnostic test yields a positive result and the patient does not have a disease.

Hence, $(a + b + c + d) = 40$.

In our particular population under study, $a = 58$, $b = 2$, $c = 3$, $d = 57$.

Therefore, Prevalence of disease, $P = \dfrac{(\mathrm{a + c})}{(\mathrm{a + b + c + d})} = 0.5$

Sensitivity of diagnosis, $Se = \dfrac{a}{(a+c)} = 0.9508$

Specificity of diagnosis, $Sp = \dfrac{d}{(b+d)} = 0.9661$

False positive rate = 1–Sp $= \dfrac{b}{(b+d)} = 0.0339$

False negative rate = 1–Se $= \dfrac{c}{(a+c)} = 0.0492$

Accuracy of diagnosis $= \dfrac{(a+d)}{(a+b+c+d)} \times 100\% = 95.83\% \cdot$

The figures testify to the accuracy of diagnosis by the smart instrument.

VIII. CONCLUSION

The paper proposes the development of an FPGA based fuzzy processing system for pulmonary spirometry applications predicting the approaching obstructive or restrictive pulmonary disorder of the patient before criticality actually occurs. Architectural design of the fuzzy processor has been realized based on pipelined parallel architectures. The system has been implemented on an FPGA. In order to speed up the computation process, hybrid parallel data processing architectures with dynamic scheduling mechanism have been employed leading to a speed up of approximately 12 times. The processor can process several fuzzy rules parallely, can detect only the positive antecedents, precomputes the degree of truths of the antecedents, computes the degree of truth of only the positive antecedents and uses a defuzzification technique which could be cost effectively implemented in hardware. The processor is modeled in VHDL, realized on an FPGA and reaches a inferencing speed of 5.0 MFLIPS. The system has been applied for medical diagnosis of patients in industrial areas and has been found to give results with an accuracy of 95.83%. Further works are going on in achieving an ASIC implementation of the processor that can be more optimized in terms of speed, power and area.

IX. ACKNOWLEDGEMENTS

The authors would like to acknowledge the support lent by Ministry of Communications and Information Technology, Government of India for providing the necessary support to carry out the work. Thanks are also due to Dr. S.C. Mukhopadhyay of Massey University, New Zealand, for inspiring us to carry out the research work.

REFERENCES

[1] Long, R, Maycher, B, Dhar, A, et al Pulmonary tuberculosis treated with directly observed therapy: serial changes in lung structure and function. Chest 1998;113,933-943.

[2] Gaensler, EA, Lindgren, I Chronic bronchitis as an etiologic factor in obstructive emphysema, preliminary report. Am Rev Respir Dis 1959;80,185-193

[3] Snider, GL, Doctor, L, Demas, TA, et al Obstructive airways disease in patients with treated pulmonary tuberculosis. Am Rev Respir Dis 1971;103,625-640

[4] Wilcox, PA, Ferguson, AD Chronic obstructive airways disease following treated pulmonary tuberculosis. Respir Med 1989;83,195-198.

[5] Weiner, H Changes in employment status associated with hospitalization for tuberculosis: analysis of 163 consecutively admitted males. Am Rev Respir Dis 1963;87,17-22

[6] Hnzido, E, Singh, T, Churchyard, G Chronic pulmonary function impairment by initial and recurrent pulmonary tuberculosis following treatment. Thorax 2000;55,32-38.

[7] M. Togai and H. Watanabe, "Expert System on a Chip: An Engine for Real Time Approximate reasoning", IEEE Expert Systems Magazine, Vol. 1, pp. 55-62, 1986.

[8] M.H. Lim, Y. Takefuji, "Implementing Fuzzy Rule Based Systems on Silicon Chips", IEEE Expert Systems Magazine, Vol. 5, No. 1, pp. 31-45, 1990.

[9] L.A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes", IEEE Transactions on Systems, Man and Cybernetics, Vol. 3, No. 1, 1973.

[10] L.A.Zadeh, "Fuzzy logic, Neural Networks and Soft Computing", ACM Transactions on Computing, Vol. 37, No. 3, 1994.

[11] H. Surmann, A. Ungering, "Fuzzy Rule Based Systems on General Purpose Processors", IEEE Micro, Vol 15, no. 4, 1995.

[12] H. Watanabe, J.R. Symon, W.D. Detloff, K.E. Yount, "VLSI fuzzy chip and inference accelerator board system", Fuzzy Logic for Management of Unceratinty, John Wiley & Sons, 1992.

[13] K.Nakamura, N.Sakashita, N.Nitta, K. Shimomura, T. Tokuda, "Fuzzy Inference and Fuzzy Inference Processor", IEEE Micro, Vol. 13, No. 5, pp. 37-48, 1993.

[14] M.A. Manzoul, S. Tayal, "Systolic Array for multivariable Fuzzy Control Systems", International Journal of Systems and Cybernetics, Vol. 21, No. 1, pp. 27-42, 1990.

[15] A. Jaramillo-Botero, Y. Miyake, "A High Speed Parallel Architecture for Fuzzy Inference Control of Multiple Processes", Proceedings of IEEE 1994 World Congress on Conputational Intelligence, Vol. 3, pp. 1765-1770, 1994.

[16] H. Orsila, T. Kangas, E. Salminen, T.D. Hamalainen, M. Hannikainen, "Automated memory-aware application distribution for Multi-processor System-on-Chips", Journal of Systems Architecture, Vol. 53, No. 11, pp. 795-815, 2007.

[17] G. Aranguren, M. Barron, J.L. Arroyabe, G. Garcia-Carreira, "A pipeline fuzzy controller in FPGA", Proceedings of IEEE Conference on Fuzzy systems, Vol. 2, pp 635-640, 1997.

[18] V. Samoladas, L. Petrou, "Special Purpose Architectures for Fuzzy Logic Controllers", Microprocessing and Microprogramming, Vol. 40, No. 4, pp. 275-289, 1994.

[19] L. de Salvador, J. Gutierrez, "A Multilevel Systolic Approach for Fuzzy Inference Hardware", IEEE Micro, Vol. 15, No. 5, pp. 61-71, 1995.

[20] R. Raychev, A. Mtibaa, M. Abid, "VHDL Modeling of a Fuzzy Coprocessor Architecture", Proceedings of International Conference on Computer Systems and Technologies, pp. I. 2.1- I. 2.6, 2005.

[21] S. Roy Chowdhury, D. Chakrabarti, H. Saha, "FPGA realization of a Smart Processing System for Clinical Diagnostic Applications using Pipelined Datapath Architectures", Microprocessors and Microsystems, Vol. 32, No. 2, pp. 107-120, 2008.

[22] S. Roy Chowdhury, D. Chakrabarti, H. Saha, "Medical diagnosis using Adaptive Perceptive Particle Swarm Optimization and its hardware realization using Field Programmable Gate Array", Journal of Medical Systems, Available online.

[23] S. Roy Chowdhury, H. Saha, "A high performance generalized Fuzzy Processor Architecture and realization of its prototype on an FPGA", IEEE Micro, Vol. 28, No. 5, pp. 38-52, 2008

[24] M. Figueiredo, F. Gomide, A. Rocha, R. Yager, "Comparison of Yager's Level Set method for Logic Control with Mamdami's and Larsen's Methods", IEEE Transactions on Fuzzy Systems, Vol. 1, No. 2, pp.156-159, 1993.

[25] M. Kukar, "Transductive reliability estimation in medical diagnosis", Artificial Intelligence in Medicine, Vol. 29, pp 81-106, 2003.