AUTOMATIC GENERATION OF A VLSI PARALLEL ARCHITECTURE FOR QRS DETECTION

A. Koulouris, N. Koziris, T. Andronikos,
G. Papakonstantinou, P. Tsanakas

Digital Systems and Computer Laboratory,
Computer Science Division
Department of Electrical and Computer Engineering
National Technical University of Athens
Zographou Campus, 15773, Athens, Greece
Tel +30 1 7722494; fax +30 1 7721533
e-mail: papakon@dsclab.ece.ntua.gr

ABSTRACT

QRS is the dominant complex in the Electrocardiogram (ECG). Its accurate detection is of fundamental importance to reliable ECG interpretation and hence, to all systems analyzing the ECG signal (e.g. Heart-monitoring). Syntactic methods are a very powerful tool for QRS detection, since they can easily describe complex patterns, but their high computational cost prevents their implementation for real time applications. In this paper, we present a VLSI architecture for ECG signal processing, automatically derived using a nested loop parallelization method. This architecture detects the QRS complex by parsing the corresponding to the ECG signal input string, based on an attribute grammar describing it.

1. INTRODUCTION

Electrocardiogram (ECG) remains the simplest non-invasive diagnostic method for various heart diseases. The good performance of computerized ECG processing systems relies heavily upon the accurate detection of the QRS complex. Three different classes of algorithms for the automatic detection of this complex can be found in the literature. Non-syntactic [3]-[5], syntactic[1],[6],[7] and hybrid [8]- [10].

The main advantage of the syntactic methods is focused on their ability to use the structural information of the ECG signal and describe more complex relations than the common non-syntactic methods. However the computational cost of these methods is in general high. The average complexity is equal to the time complexity of a context free grammar (CFG) parsing ($O(n^3)$) [11],[12].

The computational cost of the parsing algorithms can be tackled by their parallel implementation in special purpose parallel hardware like systolic arrays. The parallel implementation of the Earley's parsing algorithm was first considered by Chang & Fu in [13], where they proposed a 2-D architecture. Later Cheng & Cheng used this architecture for parsing shape Grammars [14].

In this paper we first transform the Earley's parsing algorithm into a perfectly nested loop [2],[15],[16], with loop carried dependencies. Then we apply the automatic method proposed by Andronikos and Koziris in [2] to build a VLSI architecture. This method performs an optimal mapping of the loop iterations, using the less possible processing elements. We thus implement Earley's algorithm into a one-dimensional hardware architecture, consisted of O(n) processing elements, reducing significantly the hardware complexity.

The proposed VLSI architecture is suitable for parsing the QRS attribute grammar defined by Papakonstantinou et. al [1].

2. THE ATTRIBUTE GRAMMAR FOR QRS COMPLEX

The QRS attribute grammar we use, describes any string of line segments representing an ECG signal. An piecewise linear approximation algorithm [19] has been used for transforming the ECG, to string of line segments, defined by their slope. Specifically we have a set of four terminals { LP, SP, LN, SN } each one representing a line segment with large positive, small positive, large negative and small negative

slopes respectively. The attributes of the terminal symbols are the start of each line segment and its duration (i.e. sample points comprising it), denoted as (s,d). These attributes are holding some useful measurements on QRS complexes. In figure 1 we give an example of an ECG and its encoding with a string of line segments. Our goal is to design a VLSI architecture (implementing a parallel version of Earley's algorithm), which takes as input the string of line segments and recognizes the QRS complexes from the other complexes (i.e. T, P) as well as to measure some useful characteristics like their on-set, offset and duration.

Coding: SP(1,47) SP(48,63) LP(111,4) LP(115,2) LN(117,3) LN(120,4) . . . .

Fig 1: An example of an ECG and its encoding with string of terminals and attributes which can be parsed by the QRS Attribute grammar proposed in [1]

3. EARLEY'S PARSING ALGORITHM

Earley's parsing algorithm, modified by Graham et. al in [12], calculates a table T, whose elements t[i][j] are sets of dotted rules of the grammar. The formal description of Earley's algorithm given by Ghang& Fu [13], is :

Algorithm 1 (Earley's parsing algorithm)

FOR j=1 to n do
  $t_{j-1,j}=Y\otimes\{a_j\}$
FOR j=2 to n do
 begin
  FOR i=0 to j-2 do
   $t_{ij}=t_{ij}\otimes\{a_j\}$
  FOR k=j-1 down to 0 do
   FOR i=k-1 down to 0 do
    $t_{ij}=t_{ij}\cup t_{ik}\otimes t_{kj}$
end
IF there is a rule $S\rightarrow\alpha\bullet$ in $t_{0,n}$ **accept** the input string

In [13] it is proved that the above algorithm can correctly parse any CF grammar and in consequence our QRS attribute grammar. With a_j is symbolized an element of the input string (in our case it will be any of the four terminals which the ECG signal is coded to), and with $\otimes$ the main operator of the algorithm. This operator is basically consisted of simple operations between sets (like union, section, e.t.c.).

4. MAPPING EARLEY'S ALGORITHM ONTO 1-D VLSI ARCHITECTURE

In order to extract all the inherent parallelism of Earley's parsing algorithm we rewrite it in an equivalent form of perfectly 3-nested loops [17]

Algorithm 2 (Modified Earley's Algorithm)

FOR i=1 TO n
  FOR j=i-1 TO 0
    FOR k=i-1 TO j
  S1: $t(i,j)=t(i,j)\cup t(k,j)\otimes a(i-1,k)$
  S2: $t(i,j)=t(i,j)\cup t(k,j)\otimes t(i,k)$
    END
  END
END
IF there is a rule $S\rightarrow\alpha\bullet$ in t_{n0} **accept** the input string

The new algorithm builds the recognition matrix row by row so the latest element that will be computed is t[n][0] instead of t[0][n]. We give also the input string as a matrix a(i,j). The form of this matrix is the following:

$$a(i,j)=\begin{pmatrix} a_1 & 0 & \dots & 0 \\ 0 & a_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_n \end{pmatrix}$$

This form preserves the correct correspondence of the input symbols with the elements of the recognition matrix *t*, in the above computations. Finally the two statements (S1, S2) can easily be proved to be independent. Thus they can be computed in parallel.

The key for the efficient parallelization of the above algorithm is to find the set of all possible dependence vectors (called Dependence Set, and

symbolized DS), which relate different loop iterations. Any equivalent parallel algorithm should preserve these dependencies. This dependence set is:

$$DS=\{d_0(0,0),\ d_i(i-k,0)\ |\ 1\le i-k\le n,\ d_j(0,j-k)\ |\ 1-n\le j-k\le 0\ )$$

and is graphically shown in figure 2 (We have drawn only the first two dependencies (i.e. d(k, 0) and d (0,-k) , k=1,2) in order to avoid a very complex figure).

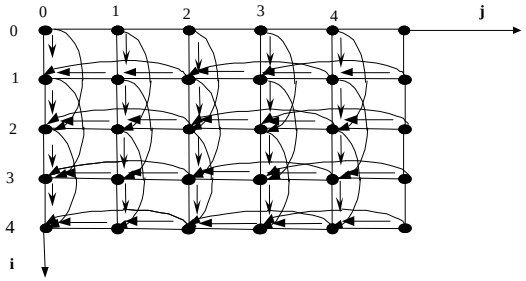

Figure 2 The Index Space and the dependence vectors for algorithm 2

We apply the method proposed by Andronikos et al. in [2], to construct the VLSI architecture. This method partitions the loop index space into the less possible disjoint chains of computations. After the partitioning phase, it assigns each chain to a different processing element, while preserving the dependence relations between the loop iterations. The resulting schedule is proved to be optimal both in terms of time and processor utilization.

	Processing Elements				
Time	P_1	P_2	P_3	P_4	P_5
1	t(1, 0)	t(2, 1)	t(3, 2)	t(4, 3)	t(5, 4)
2	t(2, 0)	t(3, 1)	t(4, 2)	t(5, 3)	-
3	t(3, 0)	t(4, 1)	t(5, 2)	-	-
4	t(4, 0)	t(5, 1)	-	-	-
5	t(5, 0)	-	-	-	-

Table 1 Time–scheduling example for n=5. Each row of the matrix t is assigned to a different processor

In table1 an example of the derived time-scheduling is given. It is clear that the above mapping, ensures the proper flow of the data. For example, the computation of element t[4][0] will start at time unit 4. For the computation of the element t[4][0], the values of the elements t[i][k], t[k][j] (j≤k≤i-1) are needed. At this time instance all these elements (i.e. t[1][0], t[2][0], t[3][0], t[4][1], t[4][2], and t[4][3]) have already been computed and sent to processor P_1

The derived architecture is given in figure 3

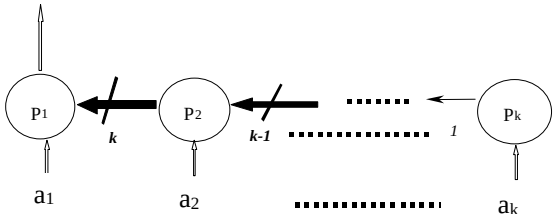

Figure 3 One dimensional VLSI-architecture for parallel execution of Earley's algorithm. The result of parsing is obtained in the first processor after n-time units

The input string is feeded in parallel to the VLSI cells and the result is obtained from processor 1, using exactly n time units. Each cell is capable to execute the main operation ⊗ as well as the evaluation of the attributes.

5. CONCLUSION - FURTHER WORK

In this paper, we presented a new automatically derived hardware architecture for ECG signal processing. This architecture can detect the QRS complex and is based on an attribute grammar [1] and on the Earley's parsing algorithm. Most possibilities of parallelization of the algorithm were investigated, using the theory of dependence vectors, and the results have been used for the efficient utilization of the hardware according the method proposed in [2]. Future work includes the implementation of the presented methodology, in a real FGPA VLSI chip, using the VHDL tool and the environment presented in [18]. We intend also to apply this method to grammars that describe the T and P complexes of the ECG signal.

REFERENCES

[1] G. Papakonstantinou, E. Skordalakis, F. Gritzali, "An Attribute Grammar for QRS Detection, " *Pattern Recognition,* vol 9, no 4, pp: 297-303, 1986.

[2] T. Andronikos, N. Koziris, Z. Tsiatsoulis, G. Papakonstantinou, P. Tsanakas, "Lower Time and Processor Bounds for Efficient Mapping of

Dependence Algorithms into Systolic Arrays," *Journal of Parallel Algorithms and Applications,* vol 10, no 3-4, pp. 177-194, 1997.

[3] J. Pan, W.J. Tompkins, "A Real-Time QRS Detection Algorithm," *IEEE Trans. Biomedical. Eng.,* vol 32, no 3, pp. 230-236, 1985.

[4] F. Gritzali, "Towards a Generalized Scheme for QRS Detection in ECG Waveforms," *Signal Processing,* vol 15, no 2, pp. 183-191, 1988.

[5] G.M. Friesen, T.C. Jannet, M.A. Jadallah, S.L. Yates, S.R. Quint, H.T. Nagle, "A Comparison of the Noise Sensitivity of Nine QRS Detection Algorithms," *IEEE Trans. Biom. Eng.,* vol 37, no 1, pp.85-98, 1990.

[6] P. Trahanias & E, Skordalakis, "Syntactic Pattern Recognition of the ECG," *IEEE Trans. Patt. Anal. Mach.. Intell.,* vol 12, no 7, pp. 648-657, 1990.

[7] A. Koski, M. Juhola and M. Meriste, "Syntactic Recognition of ECG Signal by Attribute Finite Automata," *Pattern Recognition,* vol 28, no 12, pp. 1927-1940, 1995.

[8] K.P. Birman, "Rule-Based Learning for More Accurate ECG Analysis," *IEEE Trans. Patt. Anal. Mach. Intell.,* pp. 369-380, 1982.

[9] E. Giakoumakis & G. Papakonstantinou, "A Prolog Programm for QRS Detection and Classification," *IEEE Computer in Cardiology,* 1988.

[10] E. Giakoumakis, G. Papakonstantinou, M. Sideri, "A hybrid approach to the QRS detection in ECG waveforms," *IEEE EMBS 9th Annual Conference,* 1987.

[11] J. Earley, "An efficient context-free parsing algorithm," *Commun. of the ACM,* vol. 13, pp. 94-102 1970.

[12] S. L. Graham, M. A. Harrison and W. L. Ruzzo, "On line context-free languages recognition in less than cubic time," in Proc. *8th Ann. ACM Symp. Theory of Comput.* , May 1976.

[13] Y. T. Chiang and K. S. Fu, "Parallel parsing algorithms and VLSI implementation for syntactic pattern recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-7, no 3, pp. 302- 313, 1985.

[14] H.D.Chen, X..Chen, "Shape recognition using VLSI Architecture," *The Int. Journal of Pattern. Recgn. and Art. Intell.,* 1993.

[15] Lee, P.Z and Kedem Z. M, "Mapping Nested Loop Algorithms into Multidimensional Systolic Arrays," *IEEE Trans. Parallel Distrib. Syst.*, vol. 1, no. 1, pp. 64-76, 1990.

[16] T. Andronikos, N. Koziris, G. Papakonstantinou and P. Tsanakas, "Optimal Scheduling for UET/UET-UCT Generalized n-Dimensional Grid Task Graphs," *Proceedings of 11th IEEE/ACM International Parallel Processing Symposium* (IPPS 97), pp. 146-151, Geneva, Switzerland.

[17] D. I. Modlovan, "Parallel Processing: From Applications to Systems," Callifornia Morgan Kaufmann 1993

[18] N.Koziris, G.Papakonstantinou, P.Tsanakas, "Automatic Loop Mapping and Partitioning into Systolic Architectures," 5th Hellenic Conferene on Informatics, Athens, 1995.

[19] S.Tzafestas , G.Papakonstantinou, "Microprocessor - based Data Reduction and Compres Systems," *Microprocessor -Based Control Systems,* pp.369-399, 1986.