Journal of Computer Science 3 (9): 754-759, 2007
ISSN 1549-3636

Parallel Algorithms for String Matching Problem on Single and Two Dimensional Reconfigurable Pipelined Bus Systems

S.Viswanadha Raju and A.Vinaya Babu
Gokaraju Rangaraju Institution of Engineering Technology, JNTUniversity,
Hyderabad, INDIA Pin 500072

Abstract: We considered string matching on LARPBS and 2D LARPBS. This has applications such as string databases, cellular automata and computational biology. The main use of this method is to reduce the time spent on comparisons in string matching by using LARPBS. We investigated exact string matching and approximate string matching problems. For these two sub problems, we obtained O (n) bus cycles algorithm and constant bus cycles algorithm. These algorithms have some disadvantages: Reconnecting the sub buses and shuffling the contents .These problems can be solved by 2D LARPBS.

Keywords: Hamming Distance, LARPBS, PRAM, Reconfigurability, string matching

INTRODUCTION

String matching mainly deals with the problem of finding all the occurrences of a string in a given text. String matching had been one of the most extensive problems in computer technologies during the past two decades. Pattern matching is widely implemented in information retrieval, web search engine, DNA sequencing [13,14,16], artificial intelligence and several other fields. Two sub-problems for the string matching problem considered, the first is the exact string matching and the second is approximate string matching with k- mismatches. Experimental results of well known sequential algorithm can be found in [6,9].Since the VLSI (Very Large Scale Integration) technology has been developed rapidly, hardware approaches have also been proposed [5,12]. But most of the algorithms that can be found in the literature multiphase algorithms; they need preprocessing phase in order to retrieve special structures in either the pattern or the reference string. It is then difficult to build special purpose hardware for just string matching tasks. We propose two kinds of parallel algorithms for both string matching problems without any preprocessing phase. First propose on-line solutions; it means that they process the inputs in turn, without knowledge of whole pattern at the beginning of the computation; its elements will be treated one by one. Then we give solutions that require a constant number of steps. Several computational models that have been considered for parallel string matching algorithms the PRAM model, mesh structure [17] and etc. The parallel string matching algorithm is often said to be optimal if its cost is O (nm). The first optimal O (log m) time string matching algorithm was introduced by Galil [3].

The text string of length n and a pattern of length m, Galil presented a optimal constant time complexity parallel algorithm with n processors for the exact string matching problem [3] on a conceptual CRCW PRAM model but this algorithm requires additional space and needs to preprocess the text string. Recently, on a linear systolic array, Park and George proposed an architecture based on data flow for both sub problems [15], solving them in O (n/k+ α) where k is the number of streams and $1 \leq \alpha \leq m$ with k*m processors. In this algorithm the approaches were based on the computation of the well-known Hamming distance. The string matching problem solving on the LARPBS model, introduced by Pan and Li in [7], it has known a growing success since the end of two thousands as well as all optical bus based models. Many authors have been discussed for sorting or matrix multiplication or selection [2, 8, 11].The used pipelined optical bus, instead of an electrical one. The pipelined optical bus utilizes optical fibers to transmit information.

Larpbs Model: The linear array with a reconfigurable pipelined bus system (LARPBS) consists of three waveguides. It is presented by Pan and Li in [7]. It is based on the ideas of using pipelined bus systems and processor array reconfiguration. This is called the one dimensional parallel processing optical model. In such

Corresponding Author: S.Viswanadha Raju, Associate Professor, Gokaraju Rangaraju Institute of Engineering and Technology, JNTUniversity, Hyderabad, India 500072.

model, a pipelined optical bus system uses optical waveguides instead of electrical signals to transfer the several messages which can circulate at the same time among processors through the bus in a pipelined fashion. The communication time for the LARPBS model is measured based on the number of bus cycles used. A bus cycle is the end-to-end propagation delay on the bus .The time complexity of an algorithm is determined in terms of time steps, where a single time step comprises one bus cycle and one local computation which is the time taken for an optical signal to propagate through the entire bus. The properties of optical bus as follows: Unidirectional propagation and predictable delay per unit length, in addition to the high propagation speed of light. These advantages of using waveguides enable synchronized concurrent access of an optical bus in a pipelined fashion.

The optical bus of a LARPBS processes three distinct waveguides. The message waveguide is used for sending data and the select and reference waveguides has been used for sending address information. The message waveguide is similar to reference waveguide. The shape of the bus is U structure, as per the structure of bus; the upper half of the bus is used for transmission and lower half for reception. Each processor is connected to the bus through the directional couplers, for transmitting and the other for receiving. Hence, the bus is divided into two segments, the transmitting segment and the receiving segment. The receiving segments of the reference and message waveguides have an extra segment of fiber between every pair of consecutive processors. The receiving segment is used to introduce one unit propagation delay in these two waveguides. Hence the propagation delays on the receiving segments of the reference and select waveguides not the same. In addition the select bus has switch controlled conditional delay between every pair of consecutive processors is placed on the transmitting segment segments of the select waveguide. The conditional delays can be implemented using 2x2 optical switches. The switch can function in two different states. If set to cross, a unit time delay is introduced On the other hand, if the switch is open the messages can pass-through without any delay. Hence many divide and conquer algorithms fit naturally into this model. The main features of the model is scalability and several basic data operations such as broadcast, multicast, split, binary prefix sum and minimum finding.

Reconfigurability: The importance of this model in fact that it can support dynamic reconfiguration facility on the bus. The separate set of optical switches exists on each waveguide of bus. That's why some more optical switches inserted on each segment of the bus. Half of the optical switches on both transmitting and receiving sides (one per waveguide). If all these switches set to straight, the bus system operates only the select and reference waveguide represented [4]. The message waveguide is similar to the reference waveguide. P_1 and P_4 have their switches set to cross. P_2, P_3, P_5 have their switches set to straight (Fig:2.1).

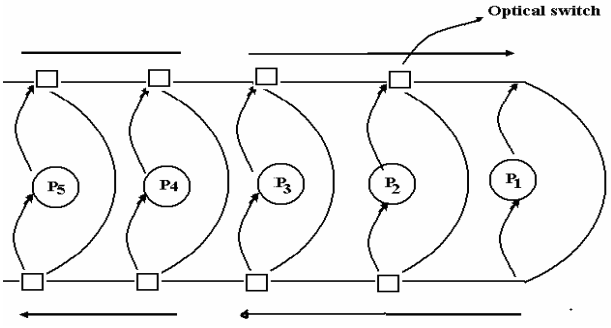

Fig 2.1: A LARPBS of size 5 with two sub arrays.

A regular pipelined bus system connecting all the processors. However, if the switches at processors P_i set to cross, the bus is split into two separate buses, one connecting processors $P_1P_2\ldots P_i$ and the other connecting processors $_{Pi+1\ldots Pn}$. Thus the whole is split into two separate LARPBS structures (only one waveguide is shown) . These two sets can work independently as two LARPBS.

Addressing: An address technology is called coincident pulse technique. The coincident pulse technique is the most common and flexible way of communication among the many methods. That addressing is done using the coincident pulse technique [11].The coincident pulse technique helps in addressing b manipulating the relative time delay of select and reference pulses on separate buses so that they will coincide only at the required receiver. The coincident pulse technique uses frames for writing and addressing information. Each processor processes a select and reference frame that has N slots for the N processors on the LARPBS. If processor P_i wants to send message to processor Pj, P_i transmits a message frame on the message waveguide and a pair of select and reference pulses on the select and reference waveguides. The technique requires the select and reference waveguides of delays between the select and reference waveguides in order to make the pulses coincide at processor P_j. With such method, it is possible to route messages or to broadcast messages on the optical bus. However when more than one message arrive at the same processor in

the same bus cycle, it accepts only the first message, the others automatically ignored. It is due to the fact that an electrical processor cannot match the transmission speed of an optical bus [7]. Why the LARPBS is powerful model, and basic operations broadcasting , multicasting, binary prefix sum computation and compression[6,9].

2D LARPBS: We propose a 2-D model in this section. See Fig 2.3 A 2-D LARPBS has two sets of switches at each row and column intersection: conditional delay switches and reconfigurable switches [10]. It has one extra pair of switches for the convenience of row bus reconfiguration and column bus reconfiguration. Actually, since we do not assume column communication be performed concurrently, each processor can use the same set of switches (including reconfigurable switch pairs and conditional delay switch) for its row bus and column bus if the switching between column bus and row bus can be done in a reasonable time. We classify a communication step into two types: a column communication step and a row communication step. The communication performance of an algorithm designed for this model is calculated by the total number of bus cycles for both row communication steps and column communication steps.

Basic operations on 2D LARPBS:2D LARPBS is more scalable than 1D LARPBS, but communication in a 2D LARPBS is more restrictive. One –to-one communication in LARPBS can be done in constant time, but not all permutation of the one-to-one communication can be done in 2D LARPBS in constant time: in this section some special permutation operations can be done in O(1) time matrix –transpose, ,2D-integer-prefix-sums, and 2d compaction. These basic permutation operations will be used as blocks in the algorithms [10].

RESULTS AND DISCUSSIONS

In results and discussion, designing of algorithms for both problems, such as the exact string matching problem[1] and the approximate string matching with k-mismatches. These algorithms designed based on the Hamming distance. Hamming distance is used for error correction. So it is best of known for finding of differences between two strings. The Hamming distance is a measure of distance between two strings equal is the number of positions for which the corresponding symbols may be different, that need to be changed to obtain one from the other, as per the definition for the following two strings: “rama” and “rimu” have a Hamming distance of 2.

Let $A=a_1a_2..a_m$ and $B=b_1b_2..b_m$ be two strings of length m , The Hamming distance of string A and string B, noted Ham(A,B) , is the number of locations where A[i]!=B[i], $1\leq i\leq m$.How to use this definition in order to solve string matching problem as shown below:

Let us consider text(T)length of n and pattern(P) length of m. suppose there is an occurrence of P in T, it means the text string $t_i,t_{i+1}..t_{i+m-1}$ equal to P, so that $H(t_i,t_{i+1}..t_{i+m-1},P)=0$. It is used for exact string matching problem. Suppose that there is an occurrence of P in T. It means that a factor $t_i,t_{i+1}..t_{i+m-1}$ of R is equal to P with at most k mismatches , hence $Ham(t_i,t_{i+1}..t_{i+m-1},P) \leq k$. It is used for string matching with k-mismatches. A naïve algorithm for solving the string matching problem can proceed as follows: consider the first n-m+1 positions of the text string. Occurrences of the pattern can start only at these positions. The algorithm checks each of these positions for an occurrence of the pattern. The time complexity for the occurrence of pattern is O(mn)

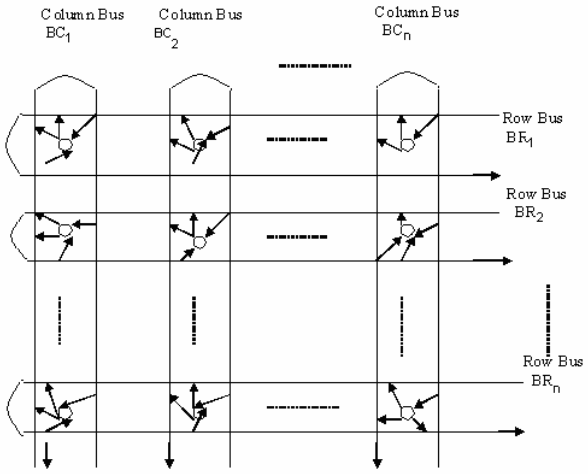

Fig. 2.3: A 2 D LARBPS

At this point, it can easily deduce a naïve algorithm for both problems on a LARPBS.Comparing all subscripts $r_i\ r_{i+1}\ldots r_{i+m-1}$ of the reference string ($1\leq i\leq$ n-m+1) of length m with the pattern , let us denote those factors R_i. This solution requires the sending of m messages ($p_1\ p_2\ldots p_m$) to perform the comparisons for each ri $r_{i+1}\ldots.r_{i+m-1}$ of a R_i. As a processor can receive at most one message per cycle, it takes m cycles to compare the pattern with all the Ri . We must add m more cycles between them to make processor i know Ham (R_i, P) from the m next processors. As it requires to know the pattern, this solution is not on-line. To reduce the number cycles to O(m) and obtain the algorithm on a LARPBS with O(n) processors. First,

describe the approach with the exact string matching problem, and will show later how to easily adapt it for the approximate string matching problem with k-mismatches.

Given a string $R=r_1\ r_2 \ldots r_n$ of length n, it requires n-processors LARPBS that contains an element of R on each processor, i.e. processor I contains the i^{th} element of R. for designing of an algorithm require one source processor that sends the elements of the pattern P one by one. The main aim of computation, for a pattern of length m, processor i ($1 \le i \le n-m+1$) has calculated Ham(Ri ,P) , i.e. Ham($r_i\ r_{i+1} \ldots r_{i+m-1}$, $p_1 p_2 \ldots p_m$).

Constant Time String Matching Algorithm on LARPBS: In this section introduce a new constant time string matching algorithm for the LARPBS model. When compared with naïve algorithm in $O(m^2)$ cycles with O(n+m) processors, our algorithm requires more processors and is much simpler in constant cycles. The present algorithm use m*(n-m+1) processors on LARPBS. It uses a different strategy when finding the occurrence of string in a text (reference string). The reference string is decomposed into m sets of length n-m+1($L_1, L_2, .. L_m$). In each set L_i is occurrences of P_j, The j^{th} element of P_j then it is denoted by $r_{k,j}$. if $r_{k,j}$ have m elements such that(k,j)={ (i,l), (i+1,2),…(i+m-1,m)} then $r_i, r_{i+1}, \ldots r_{i+m-1}$ is an occurrences of the pattern. The string matching problem can solve in O (1) bus cycle with a m*(n-m+1) processors on LARPBS.

Algorithm for string matching (text n, pattern m)
Begin

Multicast (Li): Each processor Pi,j , where $1 \le i \le m$ and $1 \le j \le n-m+1$, multicasts its Li
to Pi,j and processor holding the first element of each set also carries an element of
the pattern.

Form sub-buses: processors Pi, where $1 \le i \le n$, set their switches to cross in order to partition the array into m sub-buses. Each sub-bus has n-m+1 processor.

Broadcast: the first processor of each sub-bus broadcasts the element of the pattern it
carries.

Calculate matching array: each active processor Pi,j does the following multicast operation: uses the Ei and Pj to set its select frame , and then multicasts a dummy message within its sub-bus. If processor Pi,j receives a message in this multiple multi cast operation, then Pi,j sets its SUB-BUS(i) = i ; otherwise it sets SUB-BUS(i)=0.

Reconnecting processors: To reconnect all the m sub-buses.

Rearranging the data: the rearranging the order of processors is difficult, so we change their contents without rearranging the bus (i^{th} processor of each sub-bus to be placed in the i th sub-bus in the same order).

Sub array: the array of bus is divided into n-m+1 sub arrays with the length of m.

Compute hamming distance: suppose if there is pattern in text then hamming distance of sub array of i and pattern m is zero otherwise false.(like compute hamming distance for all sub arrays)

End

Theorem: Given two strings text n and pattern m and find the occurrences of pattern in O(1) bus cycles , and in O(1) computation time using m* (n-m+1) processors.

In the above algorithm identified three disadvantages: First, divide the bus into several sub-buses, second, reconnecting the buses, third when we need to move the processors but it is not possible that has been solving by just change their contents. For solving above problem, propose a 2D LARPBS.

Constant Time String Matching Algorithm on 2D LARPBS: The algorithm can be extended to 2D LARPBS; here is the 2D LARPBS algorithm. the data on processors have been organized such that they represent the m sets of length of n-m+1 of the text string with m* n-m+1 matrix plus, the first processor of each row segment holding the first element of each set also carries an element of pattern. The process is similar as per above for the remaining m-1 rows.

First show how to find the occurrences of pattern P in text string T on 2D LARPBS with m*(n-m+1) in constant time O (1).

Algorithm for string matching (pattern P, text T)
Begin

Initially: m elements of pattern initially distributed to the m processors on the first column, one processor and $L_{i,j}$ distributed to the m*(n-m+1) processors on the row and column. Where $1 \le i \le m$ and $m \le j \le m*(n-m+1)$

First processors broadcast elements on row buses: each first processor $p_{i,1}$, where $1 \le i \le m$, broadcasts the element $c_{i,1}$ to every processor in the i^{th} row using only row buses .After this multiple row broadcast

communication operations each processor $P_{i,j}$ saves received element as $r_{i,j.}$

Compare and Set results: Each processor $P_{i,j}$ compares $r_{i,j}$ with $X_{i,j,}$ if $r_{i,j} = X_{i,j}$ if sets result=1 otherwise, result=0

Sum up 1's: perform a one-dimensional binary prefix sum operation on each column simultaneously for the value of result. Each processor $P_{i,j}$ where $1 \leq (i,j) \leq m$ stores the binary prefix sums to $b_{i,j.}$

Based on Hamming Distance: If $P_{m,j} = 0$ then the string matching(i.e. exact string matching) otherwise approximate string matching with k mismatches.
End.
Each step in the above algorithm runs in constant time. Thus we have the following theorem.

Theorem: There is a constant time string matching algorithm on a 2D LARPBS that finds the occurrences of pattern in text using m*(n-m+1) processors

Example: Text string T(n)= abcbcd and Pattern string P(m)= bcb

L1 ={ abcb}, L2={bcbc} L3={cbcd}

b	a	b	c	b	<b,a> <b,b> <b,c> <b,b>	
c	b	c	b	c	<c,b> <c,c> <c,b> <c,c>	
b	c	b	c	d	<b,c> <b,b> <b,c> <b,d>	

step1 step2

0	1	0	1		0	1	0	1
0	1	0	1		0	1	0	1
0	1	0	0		0	1	0	0
3	0	3	1					

step4 step3

prefix sum(column)

As per the given example, after step 4 in the matrix $M_{m+1,j}$ values useful for deciding the matching is exact string matching or approximate string matching with the k mismatches.

Scalability: Suppose the number of available processors is p, if $(n\text{-}m\text{-}1) \leq p < m*(n\text{-}m+1)$, then modify algorithm in the following way to achieve good scalability. As per the above information, it is still want to split the processors into m sub-buses in step1. Each group is responsible for one set of m sets and pattern string, matching array. Thus there x=p/(n-m+1) processors in each sub-bus .every processor has three arrays :pattern, reference(Li) and matching array each of which has size m*(n-m+1)/p. the algorithm takes (m*(n-m+1))/p bus cycle and there is no local computation for multicast and contains the (m*(n-m+1))/p computation time. The time complexity for the modified string matching algorithm (m*(n-m+1))/p. in other words , the complexity T1(N)=O(1) , T2(P)=(m*(n-m+1))/p where N=m*(n-m+1) and P=p the scalability for our algorithm is g(N,P)=T2(P)/T1(N) * P/N = 1 . So that the string matching is completely scalability and obtain the following theorem.

Theorem: The given two strings size of text n and size of pattern m. find the occurrences of pattern in text. There is completely scalable on LARPBS. The algorithm runs in O(m*(n-m+1))/P time, where P is the number of processors and $1 \leq p \leq m*(n\text{-}m+1)$.

CONCLUSIONS

We studied both optical interconnection models and efficient algorithms for these models. Mainly we concentrated on a simple but powerful optical interconnection network model-LARPBS. This model extended to 2DLARPBS to improve the scalability.
We achieved a constant O(1) time algorithm for string matching on 2DLARPBS without any preprocessing the pattern string or text string. This is first known optimal algorithm for pattern matching on 2DLARPBS.Our future research on same problem with three dimensions or another mode, which contains minimum communication, and improve the performance.

ACKNOWLEDGEMENTS

We would like to thank our colleagues in the Department of Computer Science and Computer applications for their advice and helpful discussions. We thank Prof P.S.Raju, Director, and Dr.J.N.Murthy, Principal, GRIET for his moral support and providing Laboratory. We grateful to the British Library, JNTUniversity Library for their services.

REFERENCES

1 A.V. Aho, J.E. Hopcroft and J.D. Ullman, 1974"The Design and Analysis of Computer Algorithms", Addison-Wesley Publishing Company.,

2 A.Datta, 2002, "Efficient graph-theoretic algorithms on a linear array with a reconfigurable Pipelined bus system", the journal of supercomputing, 23, , no.2, 193-211.

3 Z.Galil, 1995, "a constant-time optimal parallel string matching algorithm", journal of the of the ACM, 42, no.4, 908-918.

4 Z.Guo, R.Melhem, R.Hall, D.chiarulli,and S.Levitan, 1991,"Array processors with pipelined optical busses", Journal of parallel and distributed computing, 12, no3, 269-282.

5 M.Isenman and D.shasha, 1990 , "Performance and architectural issues for string matching", IEEE Transactions on computer, vol.39, no.2, pp.238-250.

6 T.Lecrop, 1995, "Experimental result on string matching algorithms", Software-practice and experience,vol.25, pp.727-767.

7 K.Li and Y.Pan, 1999 , " Linear array with a reconfigurable pipelined bus system: Concepts and applications",information sciences an international journal, 11, no .7, 237- 258.

8 K.Li, Y.Pan, and S.Zhen, 1998 , " Fast and processor efficient parallel matrix multiplication algorithm in a linear with a reconfigurable pipelined bus system",IEEE Trans Parallel and Distributed systems 9, no 8, 705-720.

9 P.D.Michailidis, 2002, K.G.Margaritis, "On-Line String Matching Algorithms: Survey and Experimental Results", International journal of computer mathematics,79, No.2,867-888.

10 A.G.Bourgeois and J.L.Trahan, 2000 ,"relating two dimensional reconfigurable meshes with optically pipelined buses", International journal of foundations of computer science,vol.11, 553-571.

11 Y.P.J.L.Trahan, A.G.Bourgeois and R.Vaidyanathan, 2000 ,"optimally scaling permutation routing on reconfigurable arrays with optically pipelined buses", Journal of parallel on Distributed computing, 60, no.9, 1125-1136.

12 S.Viswanadha Raju and A.Vinayababu, 2004, "Performance in the design of Parallel Programming", Proc ObComAPC-2004, Allied Publications, pp.380 to 392.

13 S.Viswanadha Raju, A.Vinayababu and M.Mrudula , 2006, "Backend Engine for Parallel string Matching using Boolean Matrix" , Proc PARELEC-2006, IEEE Computer Society , 281-283.

14 S.Viswanadha Raju, S.R.Mantena, A.Vinayababu and GVSRaju, 2006, "Efficient Parallel String Matching Using Partition Method", Proc PDCAT-2006,IEEE Computer Society, 281-284.

15 S.Viswanadha Raju, A.Vinayababu, S.P.Yanaiah and GVSRaju, 2006 "Parallel Approach for K String Matching", Proc NCIMDiL-2006, Indian Institute Of Technology, Kharagpur , 5-10.

16 S.Viswanadha Raju and A.Vinaya Babu, 2006, " Optimal Parallel String Matching Algorithm on Body Centered Hypercube", International Journal of Mathematics and Computer Science,**1** No.4, 473-484.

17 S.Viswanadha Raju and A.Vinaya Babu, 2006 ," Optimal Parallel algorithm for String Matching on Mesh Network Structure", International Journal applied mathematical Sciences, **3** No.2, 167-175