An application specific TLM array processor

D.Stothard, S.C.Pomeroy.
Department. of Electronic and Electrical Engineering
Loughborough University
Loughborough. LE11 3TU

1 Introduction

The study of wave propagation in two dimensions using the transmission line matrix (TLM) method is well documented[1]. The region under consideration is mapped to a Cartesian mesh of transmission lines which are joined where they cross, forming nodes. Voltage impulses reaching a node are scattered from it according to eqn (1),

$$ {}_{k+1}V^r_x = \frac{1}{2}\left[\sum_{m=1}^{4} {}_kV^i_m\right] - {}_kV^i_x \qquad (1) $$

where V^i and V^r represent incident and reflected voltages respectively and k is the iteration counter. Large TLM models give rise to lengthy run times as (1) must be calculated for the four branches of each node in every iteration. The repetitive nature of TLM makes it apparently suited to solution using a single instruction, multiple data-stream (SIMD) architecture, however such approaches[2-4] are impractical and make inefficient use of the resources available. An application specific processor (ASIC) can overcome this inefficiency. By optimising the design to perform only the TLM algorithm, removing the need for instruction fetching and decoding, a very high throughput can be achieved using a simplified processing element.

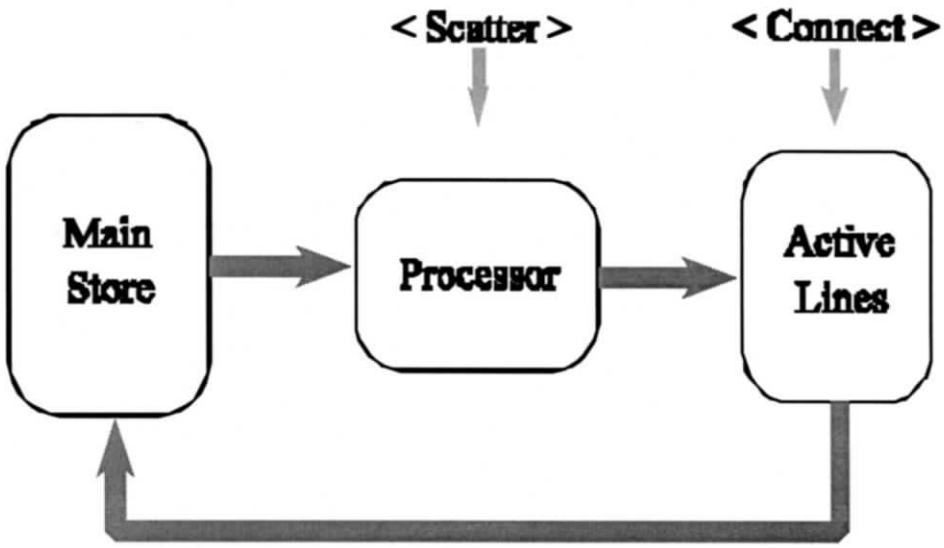
Previous attempts to design an ASIC for TLM[5] have been limited by high pin counts, fixed word lengths and constraints incurred through the use of logic synthesis; these limitations have prevented the integration of the ASIC in to a complete system. The development of a bit serial processing node has removed many of these limitations, allowing a complete dedicated TLM array processor to be developed. This paper gives an overview of the processor and discusses it's capabilities and potential applications.

2 System Architecture

A block diagram of the system is shown in *fig.1*. Data flows in a continuous loop through the three main components, the main store, the processor and the active lines.

Each node in the array is represented by four data words held in the main store. These words, which may be either 32 or 64 bit binary integers, are the four impulses which will be incident upon that node in the next iteration. The main store therefore holds a picture of the impulses propagating through the array at any given time.

The processor and the active lines respectively perform the scatter and connect phases of the TLM process. Both these processes are mapped directly to hardware for maximum efficiency, eliminating the need for an external program.



The processor consists of a small number of bit serial TLM processing elements, each of which is capable of performing a two dimensional scattering event at a single node.

The active lines are a small, fast memory which hold a current copy of three rows of the model, as data scattered from the processor will, in the next iteration, be incident upon a node in either the same row of the array as the scatterer or in one of the two adjacent rows. The active lines and the processor are linked by a circuit which performs the TLM connection process directly in hardware, writing data to the required location in one of the three rows. This data is then used to update the main store.

3 System Operation

Before processing can begin the location and type of all targets and boundaries within the array must be determined and stored in the memory. Any impulses launched at t = 0 may also be written to the main store.

Processing follows a repeated cycle of events. The main store passes data in blocks to the processor. Each block contains the four incident data values at one node for each processing element (PE). This data is presented to the processor bit serially via an array of shift registers. Using this technique a small number of processing elements can process an array of any given size. Each PE outputs its four scattered data words and the total energy incident upon the node in that iteration. The scattered data from the processor is routed to the correct location within the active lines depending upon it's source and the presence of local boundaries. When the processor begins operating on a new row, one row held in the lines leaves the scattering window, ie. it is no longer effected by the localised scattering of the processors. This row can be written back to the main store where it forms the incident data values for the next iteration.

The total energy incident upon the node can be output to a host system for visualisation purposes or for further processing. There is no computational overhead associated with the production of the total energy data, therefore the whole array may be output for visualisation.

Target boundaries are specified through the inclusion of a three bit code which is added to the beginning of each data word. This allows the arbitrary placement of targets within the array. The system can perform boundaries with $\rho = 0$, 1 or -1. The system may also perform free space boundaries to simulate propagation within an infinite medium. This is done using the condensed space-time extrapolation method (CSTE), a hardware optimised variation on Higdon's 2nd order space-time extrapolation method[6].

4 Implementation

It is intended that the whole system will be mounted on a single printed circuit board. This board will contain the three units described above and all peripheral circuitry. The processor on each board will contain 8 processing elements, thus the main store must provide 32 bits (8 nodes x 4 words) of data in each clock cycle. Boundary data is held in a small memory separate from the main store. Data is loaded from the main store in to 64 bit wide parallel in, serial out shift registers. A switchable routing system is used to correctly place the boundary data depending upon the word length. Data is then fed out bit serially to the processor, see *fig.2*.

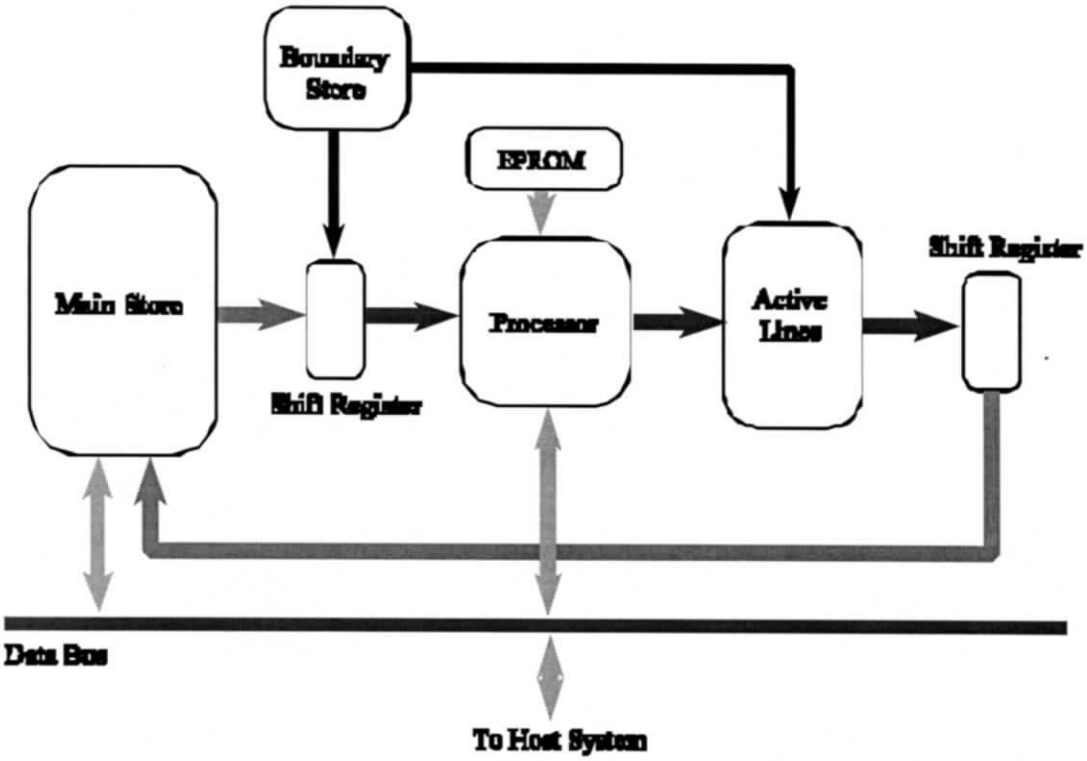
Each board will contain sufficient memory to process a 1024 x 1024 node array using 64 bit data. The actual array need not be square, any rectangular array may be processed as long as the number of nodes does not exceed 1024^2. (With 32 bit data, an array twice this size may be stored). The only condition imposed upon the array is that in one dimension it's length must be a multiple of 8 nodes.

5 System Performance

The processor unit has been fabricated and tested. Current performance figures indicate that it is typically capable of processing a 1024^2 node array using 32 bit data 8 - 10 times a second, c.f. once every 2 - 3 seconds for optimised C code running on a Pentium processor.

Although the board can be operated as a stand alone system, should faster processing or greater storage be required then any number of boards can be linked together to process a model in parallel. As data is only passed between nodes during the connection process, the only data path required between the boards is a single bit transfer between the active lines of adjacent boards. The relationship between the number of processors available and the processing rate is linear, eg. two boards will operate twice as fast or may process a model of twice the size etc.

The system has three operating modes depending upon the size of the model and the number of processors available. If M is the number of nodes in each row of the model and P is the number of processing elements available then the three modes are:



M > P : Each row of the model is processed in blocks of P nodes. If M 1 nP {n = 1,2,3...} then should there be only X nodes left on a row, where X < P, the processor will process these X nodes concurrently with the first (P - X) nodes of the next row.

M = P : The model is processed one complete row at a time. This mode of operation requires a simpler control structure.

M < P : This condition may be written as P 3 nM {n = 1,2,3...}. The processor can then be configured to process n lines simultaneously for maximum performance.

6 Upgrades and Future Developments

At present the processor will only support the basic 2D TLM scattering equation. There are many additions to this procedure[1] such as the use of lossy or inhomogeneous media. It is intended that the processor will in time support these functions. To facilitate this the processor has been fabricated on a Xilinx XC4025 field programmable gate array (FPGA)[7]. This is a reconfigurable logic array which is configured from an EPROM at power up. When a new feature is added to the processor the EPROM is reprogrammed with the new design and the processor is reconfigured accordingly, the layout of the board and its components remain unchanged. This will allow the user to hold a 'library' of different processor configurations from which they can choose the most suitable for their application.

The system must be able to communicate with the outside world. For this, we intend to use a SCSI (Small Computer Systems Interface) port. This will allow the board to connect to any SCSI compatible platform, eg. PC or Macintosh or workstation. The data is then provided to the host platform in a standard format and the system does not need to be aware of the nature of the host. The use of the SCSI port will also allow the system to integrate with CAD software for more efficient design, with the system operating as a 'TLM co-processor'.

It is hoped that the techniques developed here may be similarly applied to the development of a processor for three dimensional TLM in the future.

7 Conclusions

An application specific processor for the solution of two dimensional acoustic or electromagnetic TLM models has been presented. Using only a small number of processors the system is capable of processing large arrays at rates considerably higher than those achieved using serial computers. The system is easily expandable if faster processing or greater storage capacity is required; any number of boards may be operated in parallel to achieve the desired performance. By selecting the correct processor configuration and mode of operation the system can be optimised for specific tasks. Further expansion of the system is facilitated through the use of a reconfigurable architecture. Data is transferred through a SCSI port, providing true portability over a wide range of platforms and allowing the system to integrate with CAD software for more efficient design.

References

1 Hoefer,W.J.R 'Numerical Techniques for Microwave and Millileter Wave Passive Structures', Chapter 8, Ed. T.Itoh, J.Wiley & Sons, New York, 1989.

2 J.L. Dubard, O. Benevello, D. Pompei, J. LeRoux, P.P.M. So, and W.J.R. Hoefer: 'Acceleration of TLM Through Signal Processing and Parallel Computing', Int. Conf. on Computation in Electromagnetics, IEE, 1991, pp.71-73.

3 C.C. Tan and V.F. Fusco: 'TLM Modelling Using an SIMD Computer' Int. Jnl. Numerical Modelling **6** (1993) 299-304.

4 P.P.M. So, C. Eswarappa and W.J.R. Hoefer: 'Parallel and Distributed TLM Computation with Signal Processing for Electromagnetic Field Modelling' Int. Jnl. Numerical Modelling **8** (1995) 169-185.

5 D. Stothard, S.C. Pomeroy and I.P.W. Sillitoe: 'An Application Specific Processor for TLM.' First International Workshop on Transmission Line Matrix (TLM) Modelling - Theory and Application, Victoria (BC) August 1995, pp.277-280.

6 R.L. Higdon: 'Numerical Absorbing Boundary Conditions for the Wave Equation' Math. Comput, **49** (1987).65-917.

7 Programmable Logic data Book, Xilinx Inc, 1994