WSI Distributed Logic Memories

R.M. Lea and M Sreetharan
Brunel University.

INTRODUCTION

Advances in semiconductor technology, which have led to VLSI, are providing dramatic improvements in speed, cost and reliability of computer hardware components. However, in the future such improvements (viz. faster and larger stores, faster processors etc) will be limited by

1. the basic SISD (Serial Instruction - Serial Data) computer architecture, which has remained relatively unchanged from its original conception by Babbage in 1832 and its engineering specification by von Neumann in 1946, and
2. its means of implementation, which incurs the expensive overheads of printed circuit board layout, assembly and testing.

In view of these restrictions it is expedient to channel semiconductor development towards experimentation with new computer architectures.

Wafer-Scale Integration[1] (WSI) offers the possibility of departing from the von Neumann computer architecture and alleviating its implementation problems. By interconnecting the good chips on an undiced wafer, WSI provides a multiplicity of processing elements and bypasses the expensive stages of chip and printed circuit board manufacture. Whereas VLSI offers low-cost components at the sub-system level, WSI offers low-cost computer systems. Hence, traditional market pressures deter the speculative development of a range of VLSI chips in order to launch a radically new computer structure. However, WSI offers the integration of a new architecture in a single development.

WSI has attracted computer system designers since its inception by Petritz[2] as 'discretionary wiring' in 1967, and various such schemes have been considered since that time. Several techniques for selecting good chips emerge from this work. Discretionary wiring uses a second level of metallisation to interconnect those chips passing wafer probe tests. A model of the ALAP (Associative Linear Array Processor) proposed by Finnila and Love[3], and based on discretionary wiring, has been built and tested by Hughes Aircraft. Another technique proposed

by Elmer, Tchon, Denboer, Frommer, Kohyama, Hirabayashi and Ngina[4], uses fusible links to avoid the expense of preparing a special metallisation mask for each wafer. Other techniques proposed by Catt[5] and Manning[6], use a standard metallisation mask, but each chip comprises fault tolerance logic which enables a chain of good chips to be created. A hardware model of the former has been investigated by Aubusson[7,8] at Middlesex Polytechnic. The common architectural feature of these WSI designs is the construction of a very long segmented shift register.

Initial considerations indicate that WSI could provide the means of implementing the Distributed Logic Memory originally proposed by Lee[9] in 1962, modified by Paull[10] and Gains[11] and extended by Savitt, Love and Troop[12], Kisylia[13], Sturman[14], Lipovski[15], Wright[16,17], Beavan[18], Lea[17] and Lewin[16,18,19]. In view of this possibility, it is interesting to speculate on the influence WSI may have on Computer Architecture.

At Brunel University, the systems, software and application aspects of WSI are being investigated in an ACTP funded contract. A class of distributed logic memories suitable for fabrication with WSI techniques has been identified and a software research vehicle is being constructed for experimental investigations. Two specific distributed logic memories based on a proposal by Catt[20] have been specified for feasibility studies in text compression applications. This paper outlines the design philosophy, structural organisation and operational principles of a general member of the class of distributed logic memories for WSI fabrication.

A DISTRIBUTED LOGIC MEMORY FOR WSI FABRICATION.

The architecture of a machine suitable for fabrication with WSI is described below. The structure of this machine is not designed for a specific application, but is intended to illustrate the architectural potential of WSI distributed logic memories.

The machine is a distributed logic memory, organised as a long string of identical cells (viz. processing elements), interconnected by two communication lines, the 'fast line' and the 'slow line', which at the ends of the string are connected to a host processor as shown in Fig.1.

The host processor acts as the source and sink of information (viz. instructions and data) for the communication lines. Information flows unidirectionally along the string and is processed bit-serially in in transit without recourse to an external processing unit. Thus,

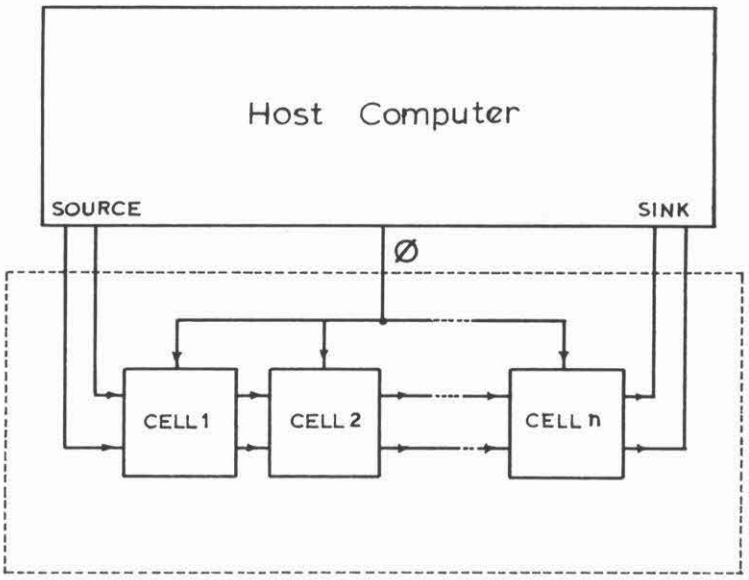


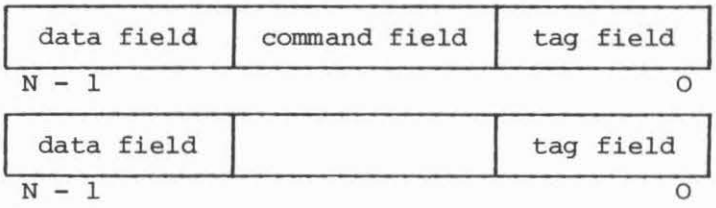
Fig 1. WSI Distributed Logic Memory

each cell acts as a sink for information flowing from upstream cells and a source for downstream cells relative to its position in the string. Thus, the slow and fast line inputs (S_i and F_i) of each cell are coincident with the slow and the fast line outputs (S_o and F_o) of its upstream neighbour.

Information on the lines is represented by fixed-length (viz. N-bits) instruction and data words. Each cell comprises storage for one data word in the slow line whereas the fast line supports a relatively unimpeded flow of instruction words. Thus, for each data word supported by the slow line, the fast line appears to support a sequence of instructions as indicated in Figures 2 and 3.

A key feature of the machine architecture is its content-addressing capability and the instruction and data words are formatted accordingly. Each N-bit word comprises three major fields, namely the tag field, the command field and the data field. Optional fields and special bits may be used to increase the operational capability of the machine.

Instruction word :



Instruction execution is preceded by comparison of the tag fields of the instruction and data words. If there is a match in cell i, the operation specified in the command field of the instruction word is executed in that cell on the operands in the data fields of the instruction and data words and the result is sourced to the downstream cell i+ 1. If the tags mismatch, then the instruction is not executed in cell i. Thus, instruction execution is conditional on the content of the tag field of data words as illustrated in Fig. 2.

Each cell comprises two major functional units.

1. the Information Flow Network (IFN)
2. the Local Control Unit (LCU)

as shown in Fig. 4.

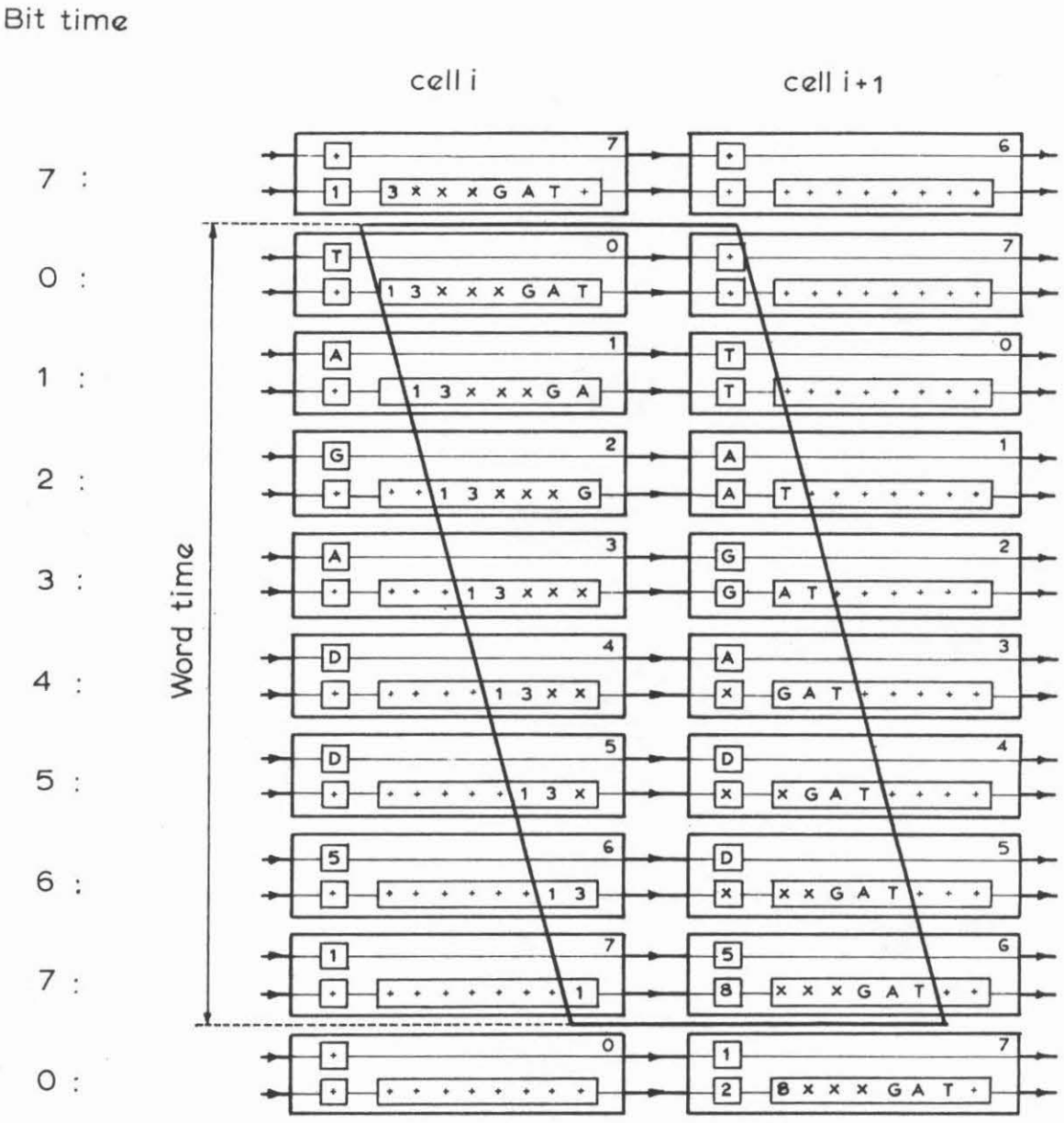


Fig 2. Data flow on the slow line

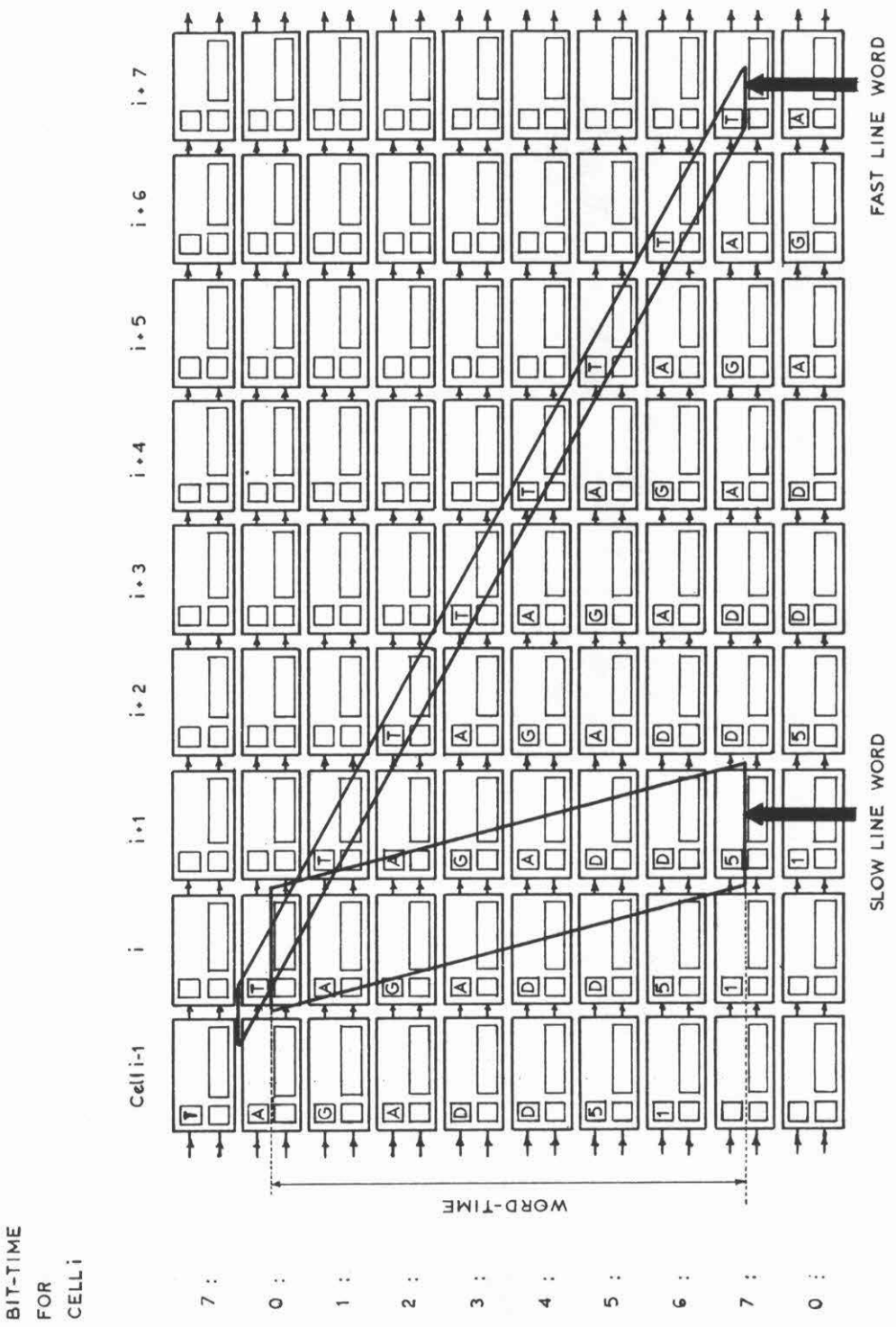
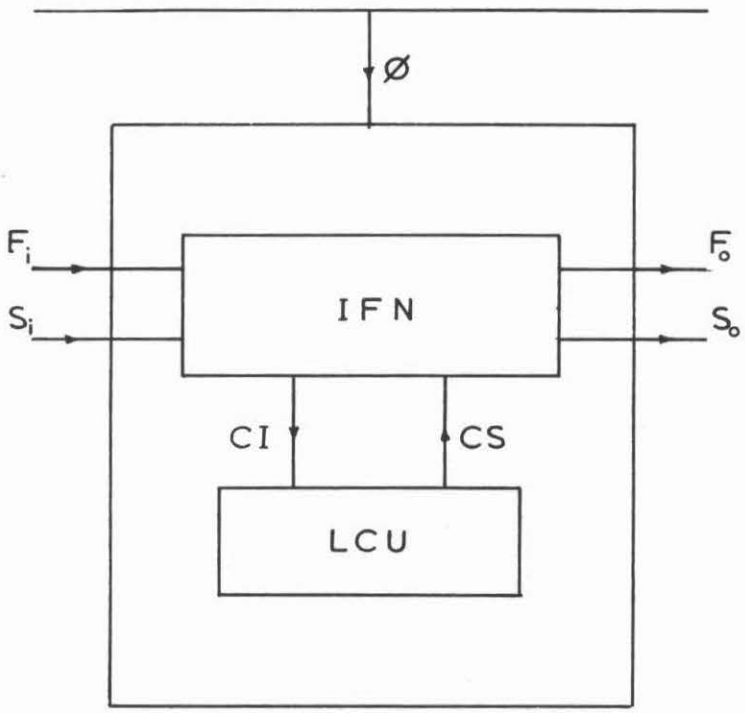


Fig 3. Data flow on the fast line relative to the slow line



Legend

IFN	–	Information Flow Network
LCU	–	Local Control Unit
F_i	–	Fast line input
S_i	–	Slow line input
F_o	–	Fast line output
S_o	–	Slow line output
CI	–	Control Information
CS	–	Control Signals
Ø	–	Global clock

Fig 4. Major functional units of a cell

The Information Flow Network (IFN) recognises instructions and where the contents of the tag field of an instruction word matches the contents of the tag field of the stored word, it routes the cell command to the Local Control Unit (LCU) which executes cell operations according to the state of a resettable counter which governs the timing within the cell.

The IFN includes a 1-bit buffer for each of the fast lines and slow line inputs (F_i and S_i) and an N-bit serial-in, serial-out shift register which provides local storage for an N-bit word. The IFN performs,

1. Flexible routing of the different fields of the instruction and data words between the fast line, slow line and the shift register.
2. arithmetic and logic operations on the data-fields of the stored data word and the incoming instruction word.

The data word stored in the shift register of a particular cell can be moved relative to the data words in the shift registers of other cells as follows:

1. Rotation of the data word within the shift register has the relative effect of progressing the word 'upstream'.
2. Transferring the data word from the shift register to the fast line has the relative effect of progressing the word 'down-stream'.

CONCLUSIONS

Wafer-Scale Integration[1-8] (WSI) could provide a means of implementing cost-effective distributed logic memories which have been of interest for nearly 17 years. Although the hardware feasibility of WSI is not yet proven, sufficient work has been done to indicate that innovative wafer assembly and packaging techniques would lead to cost-effective WSI devices. Thus, it is interesting to speculate on the influence WSI distributed logic memories might have on computer architecture.

In contrast to the von Neumann computer architecture, distributed logic memories are data-flow machines. The cpu of the former is specifically programmed to execute a sequence of operations on essentially unordered and implicitly addressed operands. Thus, the conventional machine is based on an ordered instruction-flow to an explicitly referenced single-processor, such that only one operation can be performed on only one or two operands at any given time. However, the distributed logic memory

supports a programmed data flow within which sequentially ordered operands are implicitly addressed and manipulated without reference to specific cells. Thus, the distributed logic memory is based on an ordered data-flow within an implicitly-referenced multi-processor string such that many different operations can be performed on many different operands at the same time. Consequently, WSI distributed logic memories are only suitable for those applications where large data blocks can be processed in transit.

There are three application classes where a WSI distributed logic memory could be inserted in a data stream for 'on-the-fly' information processing.

1. As a front-end processor, WSI distributed logic memories could be incorporated in the communication channels of computer networks for special purpose tasks such as error detection, data compaction, code translation etc.
2. As a main-frame processor, the intrinsic parallelism of WSI distributed logic memories offers considerable benefits for string processing algorithms. Hence, WSI distributed logic memories should be well suited to sampled data processing (viz. signal correlation, digital filtering, fast fourier transform etc) and text string processing (viz. text editing, lexical analysis etc.) However, it is unlikely that WSI distributed logic memories will benefit conventional main-frame processing techniques involving random store accesses.
3. As a back-end processor WSI distributed logic memories could be incorporated in peripheral equipment controllers for file searching updating and maintenance in support of data-base management and information retrieval systems.

Although the application environment of WSI distributed logic memories is well developed the structural organisation and programming philosophy of such devices are still matters of research. Accordingly, software simulations, written in Pascal, of specific WSI distributed logic memory structures are under investigation at Brunel University. In addition, the feasibility of WSI distributed logic memories for on-line text compression, for efficient data communication and storage systems, is being investigated, as part of an ACTP funded contract.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of ACTP (Department of Industry) and the many fruitful discussions with R.C. Aubusson, I. Catt and E.A. Newman.

REFERENCES

1. R.C. Aubusson and I. Catt, "Wafer-Scale Integration A New Approach". ESSCIRC Digest 77, pp. 76-78, Sept. 1977.

2. R.L. Petritz, "Current status of LSI technology", IEEE. J. of Solid-State Circuits, vol. SC-2, pp. 130-147, Dec. 1967.

3. C.A. Finnila and H.H. Love, "The Associative Linear Array Processor", IEE Trans. on Computers, vol. C-26, No. 2. pp. 112-125, Feb, 1977.

4. B.R. Elmer et al., "Fault tolerant 92160 but multiphase CCD memory", IEEE Int. Solid-State Cir. Conf. Dig., pp. 116-117, Feb, 1977.

5. I. Catt, "Improvements relating to digital integrated circuits", British Patent Specifications 1 377 859, Dec, 1974.

6. F.B. Manning, "An approach to highly integrated, computer-maintained cellular arrays", IEE Trans. on Computers, vol. C-26, pp. 536-552.

7. R.C. Aubusson and I. Catt, "Wafer-Scale Integration - A Fault-Tolerant Procedure", IEEE J. of Solid-State Circuits, vol. SC-13, pp. 339-344, 1978.

8. R.C. Aubusson and R.J. Gledhill, "Wafer-Scale Integration - Some Approaches to the Interconnection Problem", Microelectronics, vol.9, No.1, pp. 5-10, 1978.

9. C.Y. Lee, "Intercommunication cells - bases for a distributed logic computer", Proc. AFIPS (FJCC), 22, pp. 130-136, Dec, 1962.

10. C.Y. Lee and M.C. Paull, "A content addressable distributed logic memory with applications to information retrieval", Proc. IEEE, 51, pp. 924-932, June 1963.

11. R.S. Gains and C.Y. Lee, "An improved cell memory", IEEE Trans. on E.C., 14, pp. 72-75, 1965.

12. B.A. Savitt, H.H. Love and R.E. Troop, "ASP - a new concept in language and machine organisation", Proc. AFIPS (SJCC), 30, pp. 87-102, 1967.

13. A.P. Kisylia, "An associative processor for information retrieval", Co-ordinated Science Lab. (Illinois University), Report R-390 (AD 675310), Aug, 1968.

14. J.N. Sturman, "An iteratively structured general purpose digital computer", IEE Trans. EC-17, pp. 2-9, 1968.

15. G.P. Lipovski, "The architecture of a large associative processor", Proc. AFIPS (SJCC), 36, pp. 385-396, 1970.

16. J.S. Wright and D.W. Lewin, "A draft specification for a symbol processor", IEE Conf. Computer Science and Technology, (IEE Pub. No. 55) pp. 282-295, 1969.

17. R.M. Lea and J.S. Wright, "A Novel memory concept for information processing", Datafair Research Papers, 11, pp. 413-417, 1973.

18. P.A. Beavan and D.W. Lewin, "An associative parallel processing system for non-numerical computation", The Computer Journal, 15,4, pp. 343-349, 1973.

19. D.W. Lewin, "Introduction to Associative Processors", Proc. NATO Advanced Study Institute on Computer Architecture 12-14 Sept, 1976, St. Raphael, France, pp. 217-234.

20. I. Catt, "Specification of Property 1a invention", Aug, 1978.