The VLSI Implementation of the Σ Architecture

S.R. Williams and J.G. Cleary

Computer Science Department, University of Calgary
Calgary, AB, Canada T2N 1N4

Abstract

A description is given of a parallel computer architecture called Σ and its implementation as a full custom design in 2μm VLSI technology. The architecture is highly parallel, consisting of many simple processing elements heavily interconnected. The processing elements perform threshold computations on thousands of inputs. This architecture was inspired by research under the "neural network" banner and retains the highly interconnected nature of such systems. However, it differs from them in some key areas. The Σ architecture is digital, it provides greater functionality with respect to the type of threshold comparison done, the connection weights remain static for the duration of a problem, and its processing is deterministic. Communication between units is in single bit values which are heavily multiplexed to reduce the amount of physical interconnect and pinout.

THE Σ ARCHITECTURE

The Σ architecture was motivated by the need for a system suited to a number of areas that cannot be efficiently executed on conventional serial, or von Neuman, computers. Neural network research heavily influenced this architecture but rather than attempt to duplicate all their features, the Σ architecture focuses on achieving a large number of highly interconnected processors. Each processor is very simple, so that alone it is incapable of performing much meaningful computation, but collectively many such processors can be made to perform complex tasks.

A complete Σ system is composed of a conventional host computer and a number of identical chips, containing processing elements (σs). Each σ sums its inputs, and its output is a programmable linear comparison function of this sum against a programmable threshold. Each input is selected for summing by a weight of 0 or 1. The comparison performed can be either $>$, $\geq$, $<$, $\leq$, $=$ or $\neq$ to the threshold. Within this scheme all the standard logic functions can be obtained and in addition so can more complex functions not easily expressed in boolean logic, for example "any three of some large set of inputs are true".

To achieve the degree of interconnection between the σs that is required by this architecture, each chip has 1000 inputs. This input pool is shared by all the σs on the chip. This allows any σ to compute results on anywhere from 1 to 1000 inputs. The inputs available

will depend on how the system is physically wired together – chip inputs originate as outputs from other chips, or occasionally from an external source, such as the host.

A σ weights the chip input set by a locally stored weight vector to arrive at a subset of inputs used in its calculation. The vectors form a connection matrix that prior to computation is programmed to described the connections between σs. With this programmable scheme, communication is configurable to a structure matching application requirements, thus accommodating a wider range of problems than a hard-wired pattern.

The host provides programming and control for the system including downloading programs for execution. Programming is done for each σ by setting the weights attached to its inputs, the form of threshold comparison to be done, the value of the threshold and the weight vector. The host also provides a way of injecting inputs into the system for processing. To intercept the final outputs of the system there is an interface back from the Σ system to the host.

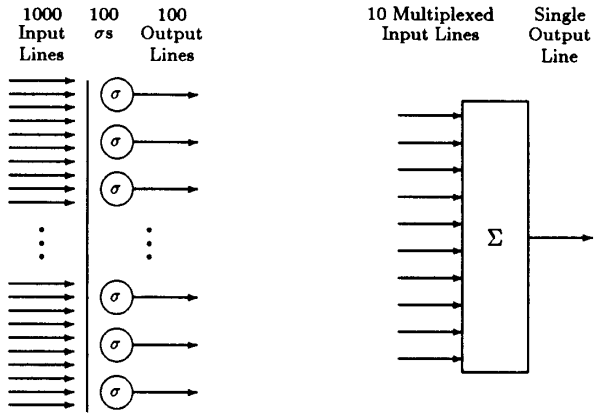

Figure 1 Multiplexing of Inputs and Outputs

TOWARDS A VLSI IMPLEMENTATION

The architecture lends itself well to implementation in VLSI circuit technology. The repetition of the basic processing element hundreds or thousands of times immediately reduces the VLSI design complexity (see Fairbairn 1982) – a key consideration in managing the design of chips in excess of 100,000 transistors. In addition, due to the scale of integration possible, a VLSI implementation allows many of the σs to be integrated onto a single chip thereby minimizing system cost.

The decision of implementing the Σ architecture as a VLSI chip is not without cost. In a VLSI implementation where a large number of processors are interconnected, anything other than single wire interconnection is prohibitive. The VLSI design for the Σ architecture allows only single bit datum to be communicated between the σ units, and single bits are used as input weights. Thus the current design includes no provision for excitatory and inhibitory

connection levels, instead the information in the connection matrix stores only whether there is or is not a connection between σs. This binary data limitation is not one of concept but rather a question of chip area.

The distributed nature of the system and the inherent large communication bandwidth is also a problem in VLSI since fewer I/O pins exist than inputs and outputs. This pinout problem is solved by heavily multiplexing the input and output signals on wires. Multiplexing also has the advantage of reducing the amount of interchip wiring. Multiplexing signals, particularly the inputs, influenced much of the chip design.

The current realization of the Σ design has 10 physical inputs to each chip, see Figure 1. The number of inputs seems right at 10, this is sufficient for most problems including mesh wiring patterns where an input can be received from each of 8 neighbors as well as a feedback from the Σ chips own output and an input for control and data from the host. An example of a complete Σ system is given in Figure 5.

Each physical input has 100 virtual bit streams multiplexed onto it for a total of 1000 inputs available to each chip. Each chip has a single physical output line. So that it can serve as an input directly to other chips it also has 100 output lines multiplexed onto it. The natural size for a chip would then be to have 100 σs each generating one of the virtual outputs. However, area limitations on the MOSIS process for which the design was targeted limited the number of σs to 20 per chip. The design however allows five of these chips to be combined to form a single Σ unit and to generate a single output line, see Figure 2.

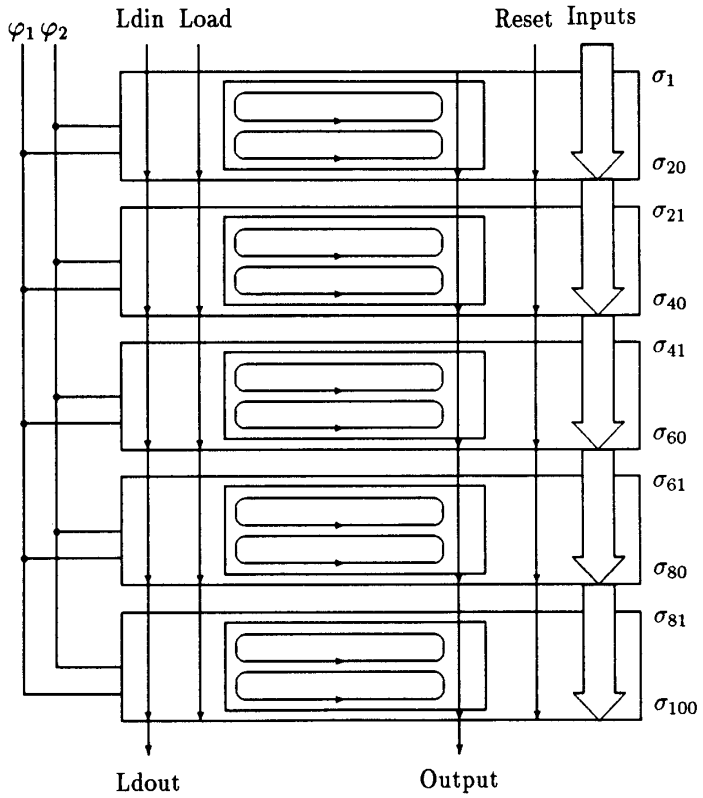

Figure 2 A Σ as 5 Cascaded Chips

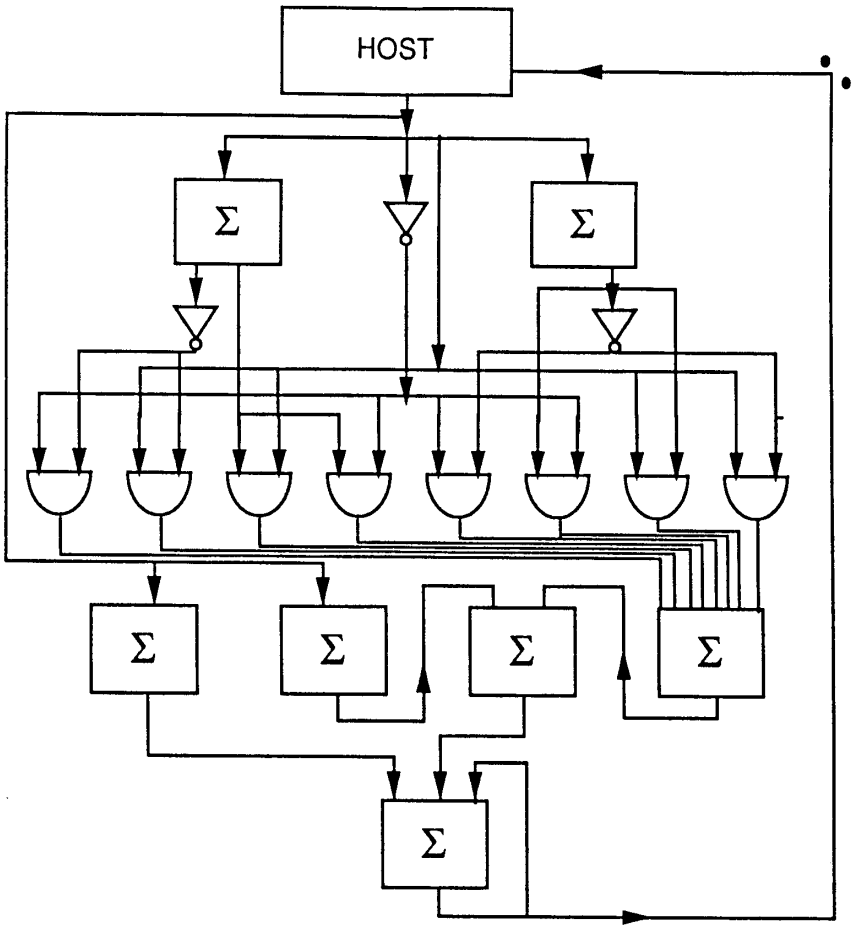

Figure 3 A Complete Σ System for Picture Thinning

APPLICATIONS

There exist many examples where a high degree of concurrency can be arranged by matching the processing elements to the natural structure of the data. The Σ architecture supports such processing through its many processing elements and a programmable connection matrix. For problems that need parallelism and the high degree of connectivity, they are primitive properties of the system and do not need to be grafted on top of an existing structure. The processing elements compute simple threshold functions on their inputs; sufficient processing power so that collectively they can solve interesting problems in areas of cellular automata, image processing and pattern recognition. Some examples are given by Sahebkar (1987) and Cleary (1986, 1987) including the n-queens problem; rule-based reasoning; string searching; shortest path; completion of a jigsaw; parallel thinning; playing tic-tac-toe and the game of life. Such

applications exhibit properties of conflict detection, constraint propagation, pattern matching, decision making, feature extraction and reducing the search space.

Figure 3 shows how a complete Σ system can be configured. This example does parallel thinning for black-white pixel images (for the original description of the problem see Holt, Stewart *et al* 1987). It operates on a 10x10 pixel field. Sahebkar (1987) includes a more elaborate mesh design which can be scaled to any size field of view. The system makes use of some simple gates to do primitive operations on all the bits in a multiplexed line – one gate does the work of 100. The resulting system can do one parallel thinning operation on the entire field in 40μsec when clocked at 10 MHz.

VLSI DESIGN

The chip layout was done in the Electric CAD system. (Figures 4 and 5 give the floor plans of a σ and a Σ). Further details of the design are given by Williams (1990). The design obeys MOSIS's (conservative) 2μm double-metal process design rules. The number of transistors is in excess of 200,000 covering a 9200μm x 7854μm area. Fortunately, the overall structure of the design is more like that of a RAM than a CPU – inherent in the design is a great deal of repetition. This regularity makes the design viable as a full custom design, which was essential in achieving the layout density required for a single chip of this size.

The layout of a single σ can be divided into two parts: the storage of its connection weights, and the logic to compute the weighted summation and the threshold comparison. The weights are stored as bits circulating in 10 shift registers, one for each physical input stream. This provides a compact and simple storage scheme and allows convenient access to the values while accumulating the sums.

A serial adder accumulates 10 weighted inputs into a register holding a running total of the weighted sum. As each of the inputs is a 0 or 1 and each of the weights is similarly a 0 or 1, input weighting is equivalent to "and"ing the inputs against a mask of weights. To speed the threshold comparison it is done in parallel, and the weighted sum register is compared with zero. This requires the weighted sum register into which the weighted inputs are accumulated be initialized with the two's complement of the desired threshold. This negative threshold value is stored in a register. Another register, storing the comparison type, selects the σ output from the comparator results.

The calculation by a σ is a cumulative process. On each clock cycle, 10 multiplexed inputs are weighted and accumulated. After 100 cycles all inputs have been processed by the σs to yield an output. The weighted sum registers are re-initialized and the process repeats for the next batch of inputs.

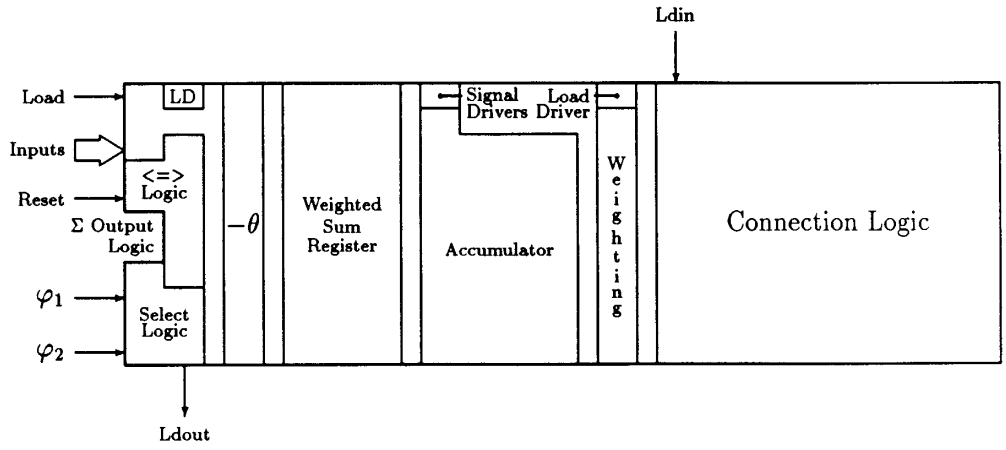

Figure 4 Floor Plan of a σ

The remaining chip circuitry interfaces the σs to the off-chip environment and provides a means to load the chip from an external source. A shift register is used to multiplex the σ outputs onto the output line: σ outputs are latched and shifted off-chip over the subsequent 100 cycles. During loading the connection weight registers, the weighted sum registers, the negative threshold registers, and the output function registers of the σs are linked together to form a load line. The floor plan of a complete Σ chip appears in Figure 5.

The inclusion of testability into any VLSI design is important to assure that no faults exist in individual chips due to fabrication errors. Additional circuitry is not needed to test a Σ chip for fabrication errors. For testing purposes the load line also serves as a scan path. Since all the internal registers are on this scan path the chip is 100% testable: every register can be controlled and observed.

Performance predictions and electrical characteristics for the layout were derived from SPICE simulations of the leaf cells found in the design. Although conservative timing estimates of the internal computation delays result in a clock frequency of 18MHz, the system is to be clocked at a much slower 10MHz rate. The clock speed is not limited by the processor speed, but by the power consumption of the chip. The problem is the number of registers switching at high speeds – the faster elements are clocked the more frequently the transistors switch, the higher the average current drawn, and the more power consumed. Table 1 compares the power consumption calculated for various chip components at clock frequencies of 18MHz and 10MHz. Ceramic chip packaging can dissipate a maximum power of 2.5 watts without special cooling requirements (see Glasser and Dobberpuhl 1985). Hence the clock speed is has been tuned down.

The results from the SPICE simulations further help in circumventing several potentially hazardous conditions from arising by ensuring the power and ground supply lines are of adequate width. Sufficiently wide supplies keep current densities below the metal migration limit and ensure that the supply rails have low resistance so that switching transients do not introduce significant noise.

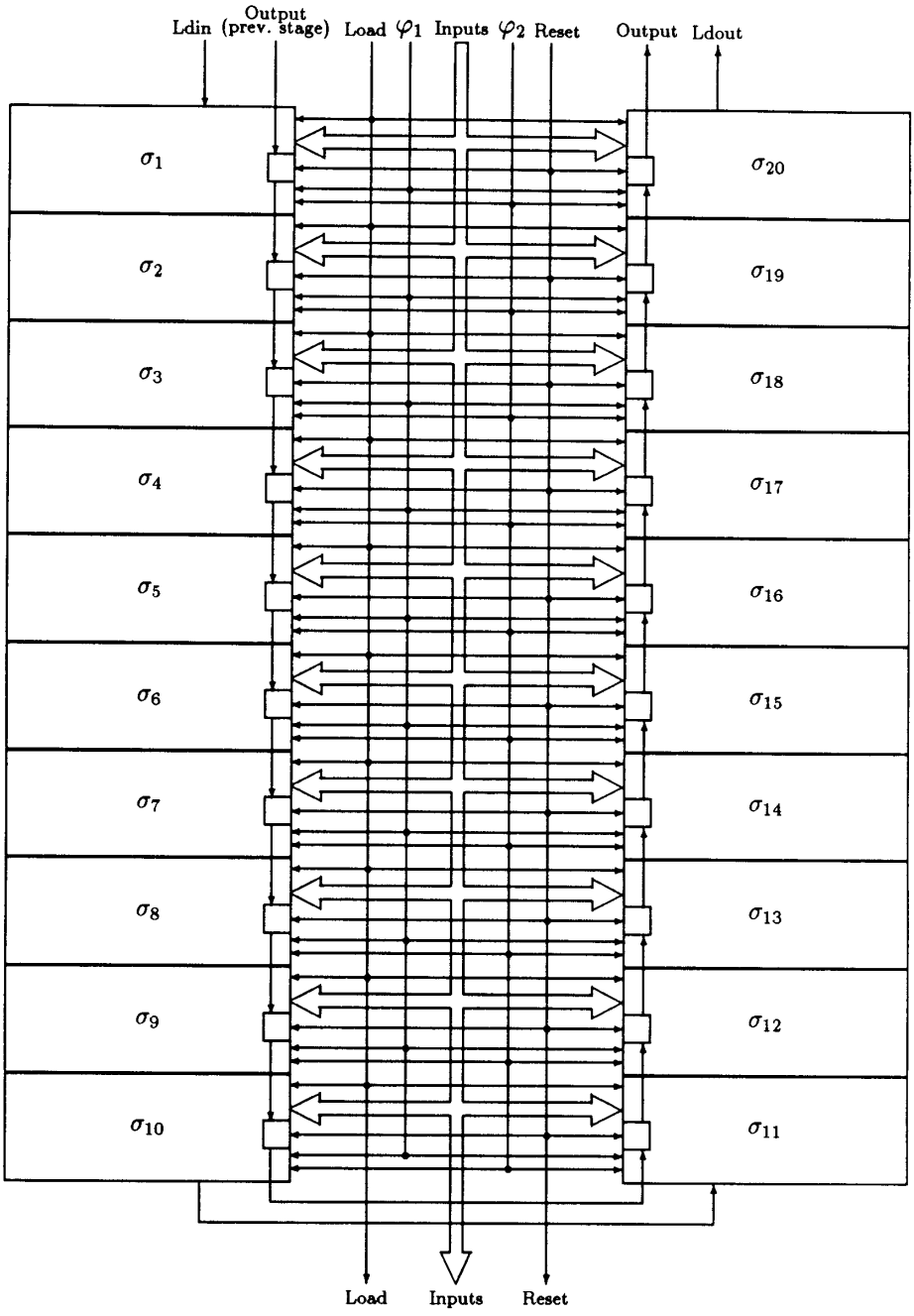

Figure 5 Floor Plan of a Σ

Table 1 Power Consumption.

	Power	
Component	18 MHz	10 MHz
Connection Weight	164.3 mW	91.3 mW
Threshold Logic	27.7 mW	13.4 mW
σ	192.0 mW	106.7 mW
Σ Output Logic	1.9 mW	1.1 mW
Σ	3.84 W	2.13 W
Pad Frame	90.0 mW	50.0 mW
Chip Total	3.93 W	2.18 W

SCALING WITH TECHNOLOGY

As current state-of-the-art technology becomes more readily available and the fabrication technology further scales down this will open many avenues for design experimentation with the Σ architecture. First order scaling of the linear dimensions increases circuit density in an inverse square relation. Estimates indicate that it would be feasible to squeeze the full 100 σs of a Σ onto a single chip in a fabrication process of approximately 0.9μm (the current 20 σ design is for a 2μm process). The dimensions of the system are not cast in stone. Other designs may implement a different combination from the 1000 inputs and 100 σs presented here. The underlying architecture is flexible and easily extensible. Table 2 shows the comparison of the current Σ design with an ideal system that awaits further advances in technology. A VLSI design for such a scaled version shares many of the same principles and design ideas presented in this paper.

Table 2 Scaling of Dimensions for Different Σ Implementations

	Current Design	One Chip Design	Ideal Design
Fabrication	2 μm CMOS	0.9 μm CMOS	0.35 μm CMOS?
Multiplexing	100	100	500
Physical Input Lines	10	10	10
Virtual Input Lines	1,000	1,000	5,000
Number Connections	20,000	100,000	2,500,000
Number of Transistors	200,000	1,000,000	25,000,000

CONCLUSIONS

The arrival of VLSI technology has removed a fundamental constraint from computer architecture. Designers are no longer rigidly bound by the cost of processing logic. Ideas that were once impractical and only reasoned about are now within reach and can be built. Of the numerous possible parallel architectures, a system composed of simple processors such as the Σ is ideal for implementation using current VLSI technology.

ACKNOWLEDGEMENTS

This work was supported by the Natural Sciences and Engineering Research Council of Canada, and by the Alberta Microelectronics Centre. The authors also thank Masoud Sahebkar for many enlightening discussions and gratefully acknowledge his effort in programming a simulator and implementing a number of applications.

REFERENCES

Cleary, J.G. "Connectionist Architectures", Research Report 86/234/8, Department of Computer Science, University of Calgary, 1986.

Cleary, J.G. "A Simple VLSI Connectionist Architecture", *IEEE First International Conference on Neural Networks*, San Diego, pp 419-426, 1987.

Fairbairn, D.G.. "VLSI: A New Frontier for Systems Designers", *IEEE Computer*, vol. 15(1), pp. 9-24, January 1982.

Glasser, L.A. and Dobberpuhl, D.W., *The Design and Analysis of VLSI Circuits*, Addison-Wesley, Reading, MA, 1985.

Holt, C.M., Stewart,V., Clint, M. and Perrot, R.H., "An Improved Parallel Thinning Algorithm", Communications of the ACM, vol. 18(2), pp. 255-264, 1987.

Sahebkar, M., "An Analysis of Algorithms for a Connectionist Architecture", Project Report Department of Computer Science, University of Calgary , 1987.

Williams, S.R. "The VLSI Implementation of a Fine-Grained Parallel Architecture", MSc Thesis, also available as Research Report 90/391/15, Department of Computer Science, University of Calgary, 1990 .