

Novel Approach in Computing with Ultra-High Processing Throughput

Son Nguyen, Joan Delalic
Temple University

Bjron Gruenwald, Phil Monson
Hilbert Company

1947 N. 12th Street
Philadelphia, PA 19122
Email: stnguyen@temple.edu
Tel: (215) 204 3382
Fax: (215) 204 5960

This paper will present the creation of a multi-cellular meta-processing unit on a single wafer. Twenty wafers will house about four million computing cells, with a combined throughput of over two billion MIPS (Million Instructions Per Second). The advantage of this new approach lies in extremely high execution capability in processing a given program. The same type of program processed in one minute using this system, would take approximately twenty or more years to execute on currently available processors. Additionally, the system is self-organizing, which means that the multitude of computing cells is transparent to the user. The system can be treated as a single or few input entities, and there is no need to program the individual computing cells, which would be an impossible task. A system containing about four million cells is small in size, and will be fabricated on individual 8" wafers. This revolutionary computing tool could be used in numerous fields such as Medical Research, Space Exploration, Scientific Research, Telecommunications, Finance, Forecasting, Simulation, the Internet, etc. In addition, the individual cells do not need to be lifted from any of the wafers, and the defective cells due to manufacturing process will be bypassed by creative wiring.

Keywords: Meta-computer, ultra-high throughput, self-organizing.

1. Introduction

The vast majority of computers in use today, from mainframe to laptops, use the Neumann single processor architecture. Developed in the 1940s, it has remained the basis of computer design ever since. In this design, one central processing unit (CPU) executes all instructions in a sequential, one at a time, fashion. In the 1970s, CPU evolved into a single chip design –the microprocessor, rigorously maintaining the original sequential characteristics.

To meet the ever increasing demands of computer users, microprocessors have grown larger and more complex as demonstrated by Intel's earliest processors to the more recent ones. Sophisticated operating system software has been developed to control the complex flow of instructions into and out of the processor, and to handle the input/output functions of the computer such as data storage and retrieval devices, printers, keyboards, display, etc.

As microprocessors have grown in size, and complexity, so have their respective operating systems. For example, Microsoft's Window XP is enormous and extremely complex compared to the original DOS operating system. It certainly delivers more capabilities. In spite of computer systems having evolved to deliver increased processing power at each step along the way, the demand for computing power – the throughput, is increasing exponentially. The cliché "the more you have, the more you want", despite its triteness, illustrates the very nature of that.

2. Bottlenecks and limitations

Developing and running a traditional sequential computer program requires a substantial amount of organization and management overhead. Individual instructions cannot act by themselves, and would be lost in the sea of memory and labyrinths of

memory management schemes without the exclusive control of an omnipotent operating system. It is not unusual to find that the system overhead exceeds the actual program by many orders of magnitude. The overhead increases exponentially with the increase in complexity of the task. Traditional computing has become extremely overhead prone, a liability that continues to grow. Overhead consumes time and resources, diminishing system performance.

The deeply rooted design of “one instruction at the time” is the most inauspicious, yet unavoidable bottleneck in all single processor systems, and the logical next step would be to create a computer system that uses more than one processor. Although many attempts have been made to create multiple processor designs, and parallel computers, none of the models have been successful in becoming a mainstream alternative. For proof, just visit the graveyard of the parallel processor companies of the 1980’s and 90’s, including Thinking Machines, KSR, and others.

The major stumbling block is scalability, which is the ability to increase computer power by connecting more processor in parallel. The inherently centralized single processor architecture of today’s computers, causes present computer connectivity schemes to create bottlenecks, eventually choking the overall system throughput. Systems based on these architectures are not conducive to communicating massive amounts of information to large numbers of processors or nodes, as the overhead to manage it will rapidly exceed the information to be communicated.

The large amount of different types of software, making up the multilayered hierarchy of a modern computer, are usually unknown and not transparent to the computer user. Though, the modern computer has become a very useful tool for specific purposes, it has distanced and restricted the innovators from creatively accessing and exploiting their computers as a tool for original and unexplored endeavors. The computer as a commodity tool is severely restricted and limited, because it must be used only as prescribed by the hardware and software manufacturers. Currently, there are enormous amount of available software programs; however, there is a universe out there, both macroscopic and microscopic, ready to be explored, requiring tools that do not at all fit within the narrow realm of commodity standards and other apparent conveniences.

3. Living Systems

As we search for a better design models to overcome the inherent limitations, we need look no further than inside ourselves. Living systems are truly the most successful and sophisticated systems known to human. The natural phenomenon upon which living systems are built, consists of enormously large numbers of independent processes, together making up what we observe and experience, but almost never understand. The millions of dynamic processes that sustain the various life forms on this planet are in no sense analogous to the traditional computer’s centrally controlled sequential operations. To the contrary, living processes operate autonomously and concurrently, ostensibly in a random fashion. Though true understanding of living things is still quite elementary, the one observation that stands out is the incredible success of operation and survival in many different environments.

All living systems exhibit self-similarity in their makeup, in which the components and subcomponents of their structures are small replicas of the entire structure, or the whole is a replica of its constituent parts. From it arises emergent properties, which defines the concept of adding known things together and ending up with something that could not be predicted, calculated, or in any other logical way derived. For example, the union of the gaseous elements hydrogen and oxygen, an extremely combustible and dangerous combination, turns into a safe liquid called water with new emergent properties, among others to extinguish fire. There exists no knowledge to predict the emergent properties from its lower level constituent of hydrogen and oxygen. So far, we have only been able to observe these facts and learn them by heart. Living systems exhibit billions of emergent properties, intelligence probably being the most conspicuous and mysterious one [1-2].

4. Paradigm Shift

We have arrived at a new era of thinking. It differs from the old Greek model of analysis in which everything is broken down into smaller pieces explaining every step by pure reason and logic to subsequently deduce predictions. We have arrived at a model based on the combination of both reason and the “unreasonable” emergence of properties. New properties are not created, they are revealed.

Many experts believe that a paradigm shift in computer systems architecture will be the best if not the only way to achieve truly dramatic, orders-of-magnitude increase in processing power, or

throughput, needed to meet the rapidly growing demand. This research embodies the paradigm shift, by introducing a totally new concept in the evolution of computing machines. This evolution is needed and it is timely to enable the natural next step in the race of faster computers, able to do more in less time, and in particular, to enable a deeper understanding of our world and ourselves. To imitate some of the capabilities of living systems and their remarkable performance requires the notion of machines that can do many simultaneous processes, contrary to the present breed of computing machines that function totally on the old concept of one large centralized set of preconceived procedures [3].

This paper discusses the new computer technology, which is based on the utilization of a very large number of small computing cells. Each cell produces an individual process. Multiple cells are self-organizing processes that inherently are self-similar. This means that each computing process, with some added information content, is constantly replicated. During each replication, the processes gain information content and become part of a higher computing process, which look very similar to its smaller constituent processes, and so on. This is analogous to nature's way of creating the most intricate functions all the way up to intelligent life.

5. The New Architecture

The overall objective of this research is to assemble a large number of small processors organized in self-similar configurations, together capable of executing many trillions of instructions per second, and having the potential to reveal new emergent properties.

First, the actual mechanics of the self-similar nature, which unleashes unprecedented scalability, will be demonstrated through simulation. It will show the bottleneck problems in previous designs are eliminated in the developed architecture, and that the connectivity issues between the different processes are doable independent of the number of processors involved.

In this research, the creation of thousands and subsequently millions of self-similar processes on multiple 8" silicon wafers are being developed, through the process of wafer scale integration. A set of twenty wafers, each containing more than two hundred thousand five-bit processors, totaling over four million computing cells, with an individual processor speed of five hundred MIPS, would have a

throughput of more than two quadrillion (10^{15}) instructions per second, or two billion MIPS. The entire silicon wafer system will occupy a space of approximately 8"x8"x8", which is relatively little real estate considering the throughput. Power supply, casing, etc. will add to the overall dimensions.

The virtually unlimited scalability of processors arranged in this self-similar configuration will give the computer a new and inexhaustible means to keep up with the ever increasing demands of tomorrow in diverse areas of business, industry, science, and the military. Large scale network management and security, world wide databases, data warehousing, transaction processing, information on demand, and pattern recognition, are just a few examples.

A large number of operating processes may give rise to new emergent behavior, which will occur as a natural consequence of the process interaction, in a similar way as it does in living systems. The research, development, and management of emerging behavior and their properties present exciting new frontiers that can open up new and unexplored areas in all fields of human endeavor. They will subsequently become a very important aspect of mainstream computing [4].

6. Architecture and Design Implementation

The researched system is based on the interconnection of many very small computing cells. Each cell has multiple (as illustrated in Figure 1) independent, high speed, serial I/O channels, each operating at approximately 60 MHz. A cell has a five-bit word length and contains 32 words of Read Only Memory, and 32 words of Random Access Memory as well as a set of internal five-bit registers. Each cell has its own internal clock, which consists of a 500 MHz free running ring oscillator. Each cell runs at a slightly different speed as there is no synchronization between any of the cells

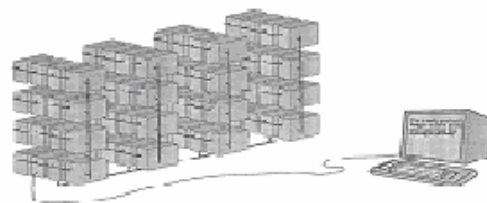


Figure 1: Fourth order system –generic configuration.

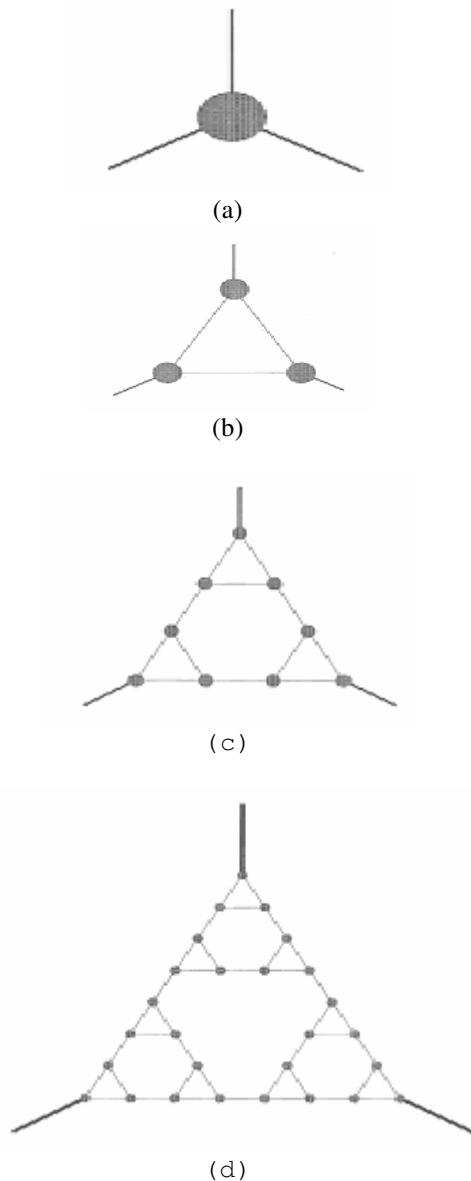


Figure 2: The particular implementation and scalability aspects of a third order system, starting from a single cell:

- a) Single third-order cell.
- b) Third-order system, 3 cells.
- c) Third-order system, 9 cells.
- d) Third-order system, 27 cells.

Additionally, cells are stack oriented, and each has a native instruction set of approximately 15-20 primitive threaded instructions for stack, register, and memory operations. These include stack operations such as DUP, DROP, SWAP, OVER, STORE, and FETCH instructions, basic comparison: XOR, Conditional Jump, Call, Return, Add and Carry, Left and Right Shift, Jump on 0, Jump on

Carry, Register Operations, etc. [5] as well as some specifically designed instructions for this particular cell such as node pointers, and others to accommodate interconnectivity between its neighbors. Each cell contains its own independent operating environment, allowing it total operational independence.

The topology of the researched system is based on the interconnection of many very small five-bit computing cells in self-similar, multi-order arrays in the following way:

Figure 1 shows the generic configuration of a fourth order system, while Figure 2 the particular implementation and scalability aspects of a third order system, starting from a single cell. The order is determined by the number of individual I/O channels of each cell. At any level, the entire sub-system looks like any other lower level sub-system all the way down to the elementary cell (Figure 2a), with only three I/O connections.

Each cell in an n -order system is connected to $n-1$ peers (neighbors at its own level) and one to the next higher level. Not every node is connected as in a traditional n -order array. Each cell has the ability to asynchronously receive or send input or output messages, or tokens, via its serial I/O connections. It has the ability: 1) to execute the token if it knows it, 2) to ignore it, or 3) to accept more information enabling it to express new tokens in terms of its previously stored information. This last alternative is determined by the neighboring cells. This will cause the cells and groups of cells, in time, to gain different sets of information and become more and more dissimilar. Therefore, the reproduction of higher and higher levels is not identical but self-similar because the internal information differs.

For example if 0.5 micron wafer scale integration is used, approximately 200,000 computing cells can be packed on one 8" wafer, and each cell can execute at least 500 MIPS. It seems likely that 42 of these five-bit cells can be pack in a 100mil x 100mil die. The number of cells per die is in a 3x3x3 configuration (see Figure 2d). There are no requirements for pads on each die because no wires are being connected to outgoing connector pins, as is the case for a chip, where the external chip pins have to be connected to the die via its pads. All die connections are internal on the wafer. Also traditional cutting space between dies is eliminated as there is no need to cut the wafer into dies. All this allows for higher density. Each cell or group of cells has only three connections which in a third order

system do not cross anywhere. This allows for the dies to be packed on a wafer adjacent to each other. The die area would then be 0.006429 sq. in. A wafer of 8" diameter, having a surface area of 50.27 sq. in, allows for 7,819 dies. This amounts to 211,113 computing cells per wafer. At 500 MIPS per cell, the throughput of one wafer system totals 105 million MIPS. The total number of cells distributed over 20 wafers, occupying a space of 8"x8"x8" equals 4,222,260. At 500 MIPS per cell, the throughput of a 20 wafer system totals 2.1 billion MIPS. The 20 wafers will be spaced approximately 1cm, allowing for either forced or natural cooling. The spacing can readily be increased to accommodate more cooling. These issues must be determined empirically after the systems are built.

7. Simulation

The simulation was done to show the idea of the parallel processing of huge amount of information via the calculation of the pixel value of a Mandelbrot display. The Mandelbrot pixel values (complex number) were calculated by a group of thousands of computing cells. Each computing cell was assigned a number of registers, a certain amount of memory, and set of instructions. The number of pixels that a cell has to compute is determined by the total number of pixels of the display divided by the number of computing cells. All the computing cells are supposed to work simultaneously.

For the Mandelbrot pixel value calculation following traditional way (sequential processing as there is only one or two processors available), the pixel values will be calculated and displayed sequentially from left to right and either from top down or bottom up. On contrary, with parallel processing of the computing cells, almost all the divided groups of pixels were calculated and displayed at the same time. So instead of seeing the Mandelbrot displayed slowly either from bottom up or bottom down, the whole Mandelbrot could be seen almost all at the same time depending on the number of computing cells.

8. Comparison with Other Computing Approaches and Conclusions

A number of different multiple processor architectures and massive processing designs have been implemented in the past several years, yet none has caught on as a mainstream alternative. Most of these, including shared memory and mesh type architectures are still based on a sequence of predefined instructions that are executed

consecutively by means of several processors to improve upon speed. Many clever schemes have been deployed to hasten the execution of these instructions. In essence, parallel processing as it is known today exhibits the same maladies and limitations as the single processor systems, because it is essentially based on similar principles. They function on the antiquated concept of one or a few large centralized sets of procedures. Systems increase in complexity each time a new revision is made, making it more vulnerable to malfunction, subject to failure, and eventual collapse.

The proliferation of the software throughout a massive parallel system is a non-trivial task, if not an impossible task in the present day architectures. It is the dilemma of the communication and management of massive amounts of different preprogrammed instructions of large numbers of processors or nodes in which the overhead rapidly exceeds the actual program. Many inhibitors to acceptance of the machines exist including high entry price, inability of the systems to run existing software, difficulty of developing new software, poor reliability, significant complexity, and more.

The need for intelligent single unit operation weighs against conventional computing whether using top down algorithms or a bottom up neural net-like device for sifting data. Intelligence is dependent upon learning which requires both top down and bottom up approaches. Neither the conventional single processor architectures nor the current parallel or multiprocessing schemes, are able to exhibit this intelligence because of their fundamental dependence upon a single known and predefined instruction sequence, barring the very occurrence of emergent behavior.

Simulation shows simultaneous data processing using a large number of small computing cells enhances the speed of computing process. This means that very huge amount of data can be processed in fraction of time compared to the currently available computing systems.

References

- [1] Anon., A probing look at strategic technologies that will significantly affect industry. Special report on: "Nested Cellular Expansion," Futuretech, No. 39, August 10, 1987.
- [2] Gruenwald, B., J., (1995), "The Emergent Properties of Asynchronous Processes" *Proc. 1995 Roc. Forth Conf. on Emerging Technology*.

- [3] Ken Baskin, "Of ants, Einstein and super computers," Technology Council News, Vol. 2, No. 9, May 1986.
- [4] Bjron J., Gruenwald "Information and Intelligence within Contractile Structures," Paper presented in a Seminar for the Physiology and Biochemistry Departments of the Albert Einstein College of Medicine, Yeshiva University, Scarsdale, NY, October 26, 1978.
- [5] Moore, Charles H., and Leach, G. C. (1970) "Forth –a language for interactive computing." Amsterdam, NY: Mohasco Industries Inc..