VLSI ORIENTED COMPUTER ARCHITECTURE AND SOME APPLICATIONS

Daniel Tabak[*]
ECSE Depart., Boston University
110 Cummington Street
Boston, MA 02215

ABSTRACT

The paper surveys the particular problems, arising in the architectural design of computing systems, realized on VLSI chips. Particular difficulties due to limited on-chip density and power dissipation are discussed. The difficulties of the realization of on-chip communications between various subsystems (between themselves and between other off-chip systems) are stressed. A number of design principles for the realization of on-chip communication paths is presented. Two design philosophies for the instruction set design in a VLSI environment are brought up: (a) The large microcoded instruction set, (b) The Reduced Instruction Set Computer (RISC) approach, based on the Streamlined Instruction Set Design.

A survey of the author's research group work in this area is presented. This includes the ZT-1 single chip microcomputer, RISC computing space studies, applications to a distributed traffic control and a la rge scale, reconfigurable communications system.

INTRODUCTION

Very Large Scale Integration (VLSI) technology is being implemented in practically all areas of modern science and technology. There exists already a vast amount of literature on VLSI system design and its impact on the architectural issues of computing systems [1-5]; only a small sample of the existing literature is quoted here. The VLSI realization of many engineering systems permits significant minituarization, compactness, increased reliability, reduced power consumption, reduced cost and many other 'fringe benefits', depending on the system properties and the nature of the implementation. Many systems, such as some airborne or human-implanted ones, could never have been implemented if

[*] Abrams-Curiel, Professor of Computer Eng., Ben Gurion Univ., Beer Sheva, Israel (on leave). From Sept. 1985: George Mason Univ. Fairfax, VA 22030.

not for the VLSI technology. Naturally, communications, telemetry, signal processing, C^3 systems, are some of the many engineering areas where VLSI is currently applied.

When we speak about VLSI implementation, we usually mean the use of computers, realized by VLSI chips, in conjunction with the system under consideration. Of course, systems other than computers (filters, modulators, demodulators) can be realized in VLSI, however this discussion will be limited to computing systems. The VLSI computing system can either be imbedded and tightly coupled with the whole system, or only loosely interconnected to it.

The current article briefly surveys some of the issues invoked by the VLSI realization, particularly concentrating the discussion on the architectural design problems of the VLSI-realized computing system. The concept of computer architecture requires an explicit clarification [6]. It can be defined (and there exist numerous definitions in the literature) as the image of the computing system, as seen by a machine-language programmer or a compiler writer. This image includes, among other things, the list of all of the CPU (Central Processing Unit) registers, accessible by the programmer, all of the data and instruction formats, all of the addressing modes, the whole machine instruction set, memory size and its hierarchy, I/O (Input/Output). The initial establishment of the above (and other associated features) is a part of the architectural design of the computing system. An example of an item belonging to computer organization, rather than architecture: the data width of the system bus. For instance, in the Intel iAPX432, which is a 32-bit machine, the data bus width is 16-bits only (meaning, a 32-bit item has to be transmitted in two bus cycles) [6]. Or for the iAPX88, utilized in the IBM PC, the data bus width is 8 bits, although it is a 16-bit system [7]. The iAPX86 and iAPX88 have the same architecture (same image to the programmer), but a different organization (different data bus).

The second part of the article contains a brief survey of some of the work done in this field by the author and his research group, consisting of his former graduate students. The work involved both basic development of VLSI-oriented computing systems as well as some potential applications, to be discussed subsequently.

VLSI ARCHITECTURAL DESIGN ISSUES

When we discuss VLSI chip realizations, the following figures of merit should be considered [1-5]:

a) Propagation delay (measured in nanoseconds, ns) - a time delay of a signal through a logic gate.

b) Speed-power product (measured in picojoules) - product of the propagation delay of a gate and its power dissipation.
c) Gate density (gates /mm^2) and bit density (bits/mm^2) - number of equivalent gates and binary bits per millimeter squared of the chip area, respectively.
d) Cost per gate (cents/gate) and cost per bit (cents/bit) - for a product that has reached high-volume (thousands and up) production levels.

The propagation delay directly influences the computing speed of the system, realized on the VLSI chip. Naturally, the designer strives to minimize this delay.

Power dissipation, which the designer also tries to minimize, is a limiting factor in VLSI design. An increase in power dissipation can bring to a breakdown of the device in the limit. Moreover, increases in applied voltage between closer located interconnections, increases the electric field strength within the chip. This in turn may cause breakdown phenomena in semiconductors. Thus, as the geometry is scaled down, supply voltage must be reduced. However, scaling down the geometry, usually reduces propagation delays and increases the speed of operation. Therefore power and delay (influenced by size) effects are related, and for this reason a speed-power product figure of merit is also used. Naturally, the designer should attempt to minimize it as much as possible.

The designer is interested in producing a chip as small and as computationally powerful as possible. He has to squeeze numerous equivalent componnents on a very small area of the chip. He would like to achieve a high gate or bit density, as possible.

Cost is really the dominant consideration in any design. This is particularly true for designs intended to create devices, used in mass production. Understandably the designer wants to minimize, as much as possible, the cost per gate (or cost per bit) figure of merit.

The expression ‘as much as possible’, used above, is not just a figure of speech. It does have a very definite meaning. Basically, all of the design goals of minimizing or maximizing certain figures of merit are limited by various laws of nature (the speed of light being the ultimate limit). Sometimes, improving one figure of merit (decreasing propagation delay) may ‘spoil’ another (increased power dissipation). The designer has to make many compromising decisions, trade-offs. Eventually, a limit will be reached. An interesting estimate of a possible limit was given by F. Faggin in [8], p. 612. According to this estimate, offered for MOS technology, for a VLSI chip operating with a supply voltage of 400 mV, with a minimum internal line width of 0.25 µm, with a power dissipation of 1W at 100 MHz of operating frequency, having a size of about 5 x 5 cm, a complexity of 10^8 equivalent gates will possibly be achieved. The current technology upper limits are between 0.5 x 10^6 to 10^6 equivalent gates. For instanpe, the density of the

NEC V-60 and V-70 32-bit microprocessors approach 10^6 transistors on a single VLSI chip [9].

The figures of merit (or specifications, or design goals) mentioned above, are to be achieved, while conforming to certain practical limitations, called constraints. Some of the major constraints in VLSI design are [10]:

(i) Upper limit of allowed power dissipation.
(ii) Upper limit of external pin count. A current practical upper limit in use is 114 (Motorola MC68020).
(iii) Upper limit of tolerated communication delays.
(iv) Difficulties in layout arrangements of the chip subsystems.
(v) Limited silicon area.

The power dissipation limit follows from physical considerations. As already mentioned earlier in this section, an excess in power dissipation can cause a complete breakdown of the device. But even before breakdown occurs, strong deviations from establishd modes of operation may also occur.

One of the major causes of chip faults is any kind of external pin breakdown. The reason for this is that it is an external interconnection, subject to continued mechanical motion of insertion and extracton from the sockets. The the more pins we have on a chip, the higher the probability of a chip breakdown, or the lower the reliability. In fact, the number of external pins is one of the major factors affecting chip reliability. Moreover, a multiplicity of pins creates a difficulty of interconnections layout between the pins and the chip subsystems to which they are supposed to be connected. These considerations limit the number of pins allowed on a chip. An addition of pins to a chip requires an addition of extra driver circuits and interconnections, thus consuming valuable and significant chip area.

A VLSI chip, intended to serve a as a computing device, is required to perform its tasks as fast as possible. On the other hand there is always a delay of signals propagating through the chip subsystems and their interconnections. The delays reduce the speed of computation. Therefore, in order to meet prescribed speed specifications, the designer, should keep the signal communication delays below appropriate limits, according to the computing speed requirement. This constraint presents to the designer a very difficult problem of wire and interconnections management with the chip. Some of the approaches of dealing with this type of constraint are:

1) Shorten, as much as possible, the length of interconnection paths. Make long paths as narrow as possible.

2) Integrate, whenever possible, wiring and logic elements.
3) Interconnect subsystems of the chip by abutting (placing to have a common boundary) them with their neighboring subsystems. This eliminates wiring and reduces interconnection paths length.
4) Avoid routing frequently repeated signals from one corner of the chip to another.

The above measures are often termed as the Localization Approach in VLSI system design [2, 3].

The current state of the art of the VLSI realization of commercial computing system is such that we have a 32-bit CPU along with a 256 bytes Cache Memory on a chip (Motorola MC68020 [11]). In addition, the 68020 chip contains over 20 32-bit CPU registers, a sophisticated Memory Management unit and other logic subsystems. On the other hand, there are 8-bit, single-chip microcontrollers containing a CPU, 8K bytes ROM, 232 bytes RAM, and five 8-bit I/O ports (Intel 8396).

A VLSI device, composed of a number of different subsystems, would require a major design effort for the layout of each subsystem and for the design of all of the interconnection paths in between. Therefore, if the chip is composed of a large number of identical subsystems the layout design task will be significantly alleviated. In this case, after designing one subsystem, we can copy its layout many times on the chip. Subsystems which appear many times on the same chip are sometimes called regular structures [10]. This brings us to the definition of the concept of the Regularization Factor [12]: the total number of devices on the chip, excluding ROMs, divided by the number of drawn (designed) devices. In other words, it is the ratio between the total number of elements on the chip and the number of different elements on the same chip.

For instance, taking some Intel chips as an example, the regularization factor on the 8086 was 2.8, while on the 432 system chips it ranged between 5.2 and 7.9 [12].

The strife for higher regularity on a VLSI chip, along with higher speed of computation, brings us necessarily to the implementation of Parallel Processing in any VLSI design of computing systems (13]. Indeed, parallel processing is being implemented in numerous VLSI design projects [2, 3, 13, 14].

One of the crucial points of the Architectural Design of VLSI-based computing systems is the Instruction Set Design aspect. We have two major design options [2]:

1) The large micro coded instruction set approach, or the so called Complex Instruction Set Computer (CISC). In this case one may attain better flexibility for the assembly language programmer. However, the control unit subsystem will be more complicated

(larger size micro code, more control signals), it may take up to 50% of the chip area, leaving less space for other features (such as on-chip CPU registers).

2) The Streamlined Instruction Set Design or the Reduced Instruction Set Computer (RISC) approach, involving a computing system with a relatively small (about 32 instructions) instruction set [2, 15, 16]. Some of the RISC advantages:

   a) Reduction of design time and of design errors.
   b) Reduction of instruction execution time (faster access to a smaller microcode).
   c) Smaller control area on chip (6%) compared to about 50% on CISCs - hence possibility of greater fast-access on-chip storage.
   d) Higher regularity factor (25), compared to 12 on the MC68000, for instance.
   e) Fixed length instructions, small number of formats, single-cycle execution.

The development of RISC-type computing systems is one of the most active areas both in Universities as well as in industry [2, 15, 17]. There exist already marketed samples offered by Ridge (Santa Clara, CA) and by Pyramid Technology (Mountain View, CA) [18]. Large companies, such as IBM and NCR also engage in this venture [18, 19].

SOME APPLICATIONS

Some projects, involving the application of both off-the-shelf as well as projected-into-the-future VLSI chips, will be briefly surveyed. The projects covered are restricted to the ones in which the author was personally involved.

a) A Parallel-Processing, Reconfigurable, Single-Chip Microcomputer.

The system proposed [14] capitalizes on projected future developments of the VLSI technology. It is still unrealizable to its full extent. However, if Faggin's prediction of 10^8 equivalent gates on a chip is accepted, it could be the single-chip microcomputer of the future.

The proposed single-chip microcomputer, called ZT1, contains a CPU and a Multiport Main Memory on the same chip, interconnected by a number of buses. There is, of course, an interconnection from one (or more) of these buses to the outside through the pins. The CPU contains a number of control and arithmetic sub units. The control system is designed in such a way that the whole system can be reconfigured into a variety of parallel processing configurations (SIMD, MIMD [13]). A reconfiguration may occur at each separate instruction and on the level of particular units such as registers, buses or ALUs (Arithmetic Logic Units). In this way it is an extension of the Kartashevs' Dynamic Architecture [20], termed Flexible Architecture.

The system can be used to process a number of information channels simultaneously (in a signal processing system, for instance), using a small system - a single-chip microcomputer, with a relatively small power consumption and cost. The ZT1 can provide on-site, sophisticated data processing in remote telemetering systems, where for various practical reasons, the installation of a regular size compter would be unrealistic. The ZT1 is somewhat similar to the recently announced INMOS TRANSPUTER (CPU and Main Memory and several buses on chip) [21]. In a way, the ZT1, developed in a thesis in 1980/81, can be regarded as a prediction of the TRANSPUTER, which was developed independently by INMOS Co.

The ZT1 has memory extension auxiliary chips and can be interconnected into a distributed network of multiple chips of the same type [14].

b) RISC-type Systems.

Stretching the RISC idea (of reducing the number of instructions in a computer) to the extreme, we arrive at the very limit of a Single Instruction Computer (SIC). Such a system, using MOVE or Conditional MOVE (CMOVE), as the single instruction, has been proposed by Lipovski [22, 23] and later researched, designed and constructed by Azaria and Tabak [24-27].

Contrary to ZTI, the SIC should be used whenever only a very simple an short program is to be executed in the particular implementation. if the RISC, in general, significantly simplifies the instruction decoding process, the SIC avoids it entirely (single instruction - no decoding necessary). Arithmetic and logic operations in SIC are performed in hardware pre-designed I/O units just by moving the operands into them, and storing (again by a MOVE) the result in memory or transmitting it out (from the I/O unit where it is obtained). The operands can be fed in directly into the I/O units from outside sensors, transducers and communication links. A system can contain even hundreds of such I/O units, distributed physically wherever needed [26]. Such a system could be particularly effective in monitoring and data collecting of large, distributed processes where at each point only a very simple data processing operation is necessary. Thus, it would be more cost-effective to put an I/O unit of a SIC system at each point, instead of a full-fledged microprocessor with 100 instructions (99 of which will never be used).

An experimental computing system MODHEL [17], where groups of instructions are microcoded in separate PLA units, which can be selectively activated, has been developed. In this way, one can use MODHEL as a RISC-type computer with different combinations of reduced instruction sets for R and D purposes.

c) A Distributed Traffic Control System.

A distributed micro computer - based control system of a large set of signalized, urban intersections has been developed [28]. The whole system has been configured in a hierarchical fashion, involving a central controller, which utilizes a mainframe computer (CDC CYBER 170 for instance), regional controllers and local controllers at each intersection. The regional and local controllers are realized by a single-chip microcontroller Intel 8051 (4K ROM on chip). Because of cost considerations, the 8031 (without the on chip ROM) was used in an experimental prototype.

d) A large Scale Reconfigurable Communications System.

A complex, reconfigurable, large-scale, distributed communication system has been developed and investigated [29, 30]. The system has been proposed to be realized by VLSI off-the-shelf components. The Intel iAPX432 system was used as an example, but other, similar systems can be implemented in the future.

SUMMARY

The particular aspects involved in the architectural design of computing systems, realized by VLSI chips have been discussed. Several examples involving the use of special purpose VLSI projected chips as well as some off-the-shelf chips, have been mentioned and some potential applications pointed out.

REFERENCES

1. Mead, C.A., and Conway, L., 1980, Introduction to VLSI Systems, Addison-Wesley, Reading, MA.
2. Hennessy, J.L., 1984, VLSI Processo r Architecture, IEEE Trans. on Computers, Vol. C-33, No. 12, pp. 1221-1246.
3. Seitz, C.L., 1984, Concurrent VLSI Architecture, IEEE Trans. on Computers, Vol. C-33, No. 12, pp. 1247-1265.
4. Ullman, J.D., 1983, Computational Aspects of VLSI, Computer Science Press, Rockville, MD.
5. Kung, H.T., Sproull, B. and Steele, G., Eds., 1981, VLSI Systems and Computations, Computer Science Press, Rockville, MD.
6. Myers, G.J., 1982, Advances in Computer Architecture, 2nd ed., Wiley, NY.
7. Liu, Y.C., and Gibson, G.A., 1984, Microcompter Systems: The 8086/8088 Famly, Prentice-Hall, Englewood Cliffs, NJ.
8. Siewiorek, D.P., Bell, C.G., and Newell, A., 1982, Computer Structures: Principles and Examples, McGraw Hill, NY.

9. Rhim, H., and Iwasaki, J., 1985, Making a new CMOS Microprocessor Part of an Extended Family, Compuer Systems Equipment Design, pp. 17-21.
10. Fairbairn, D.G., 1982, VLSI: A New Frontier for Systems Designers, Computer, Vol. 15, No. 1, pp. 87-96.
11. MC68020 32-Bit Microprocessor User's Manual, Prentice Hall, Englewood Cliffs, NJ, 1984.
12. Lattin, W.W. et al, 1981, A Methodology for VLSI Chip Design, Lambda, Second Quarter, pp. 34-44.
13. Hwang, K., and Briggs, F.A., 1984, Computer Architecture and Parallel Processing, McGraw Hill, NY.
14. Zager , E. , and Tabak, D., 1983, Flexible Architecture Microcomputer Design, Microprocessing and Micrprogrammiug, Vol. 11, No. 3/4, pp. 177-186
15. Patterson, D.A., 1985, Reduced Instruction Set Computers. Comm. ACM, Vol. 28, No. 1. pp. 8-21.
16. Patterson, D.A., and Sequin, C.A., 1982, A VLSI RISC, Computer, Vol. 15, No. 9, pp. 8-21.
17. Azaria, H.and Tabak, D., 1983, The MODHEL Microcomputer for RISCS Study, Microprocessing and Microprogramming, Vol. 12, No. 3/4, pp. 199-206.
18. MacNicol, G., March 1985, A Risky New Architecture for the Future?, Digital Design, pp. 92-98.
19. Markoff, J., Nov. 1984, RISC Chips, BYTE, pp. 191-206.
20. Kartashev, S.P., and Kartashev, S.I., 1982, Designing and Programming Modern Compters and Systems, Vol. I LSI Modular Computer Systems, Prentice Hall, Englewood Cliffs., NJ.
21. Walker, P., May 1985, The Transputer, BYTE, pp. 219-235.
22. Lipovski, G.J., 1976, The Architecture of a Simple, Effective, Control Processor, in Sam i, M., Wilmink, J.,and Saks, R., Editors, Microprocessing and Microprogramming, EUROMICRO 76, North Holland, Amsterdam, pp. 7-18.
23. Lipovski, G.J., 1978, On Conditional Moves in Control Processors, Proc. 2nd Rocky Mountain Symp. on Microcomputuers, pp. 63-94, Pingree Park, CO.
24. Tabak, D., and Lipovski, G.J., 1980, MOVE Architecture in Digital Controllers, IEEE Trans. on Compuers, Vol. C-29, No. 2, pp. 180-190.
25. Azaria, H. , and Tabak, D., 1980, Bit-Sliced Realization of a CMOVE Architecture Microcompter, EUROMICRO J., Vol. 6, No. 6, pp. 373-379.
26. Azaria, H., and Tabak, D., 1981, A CMOVE Distributed Processing System, in Richter, L., LeBeux, P., Chroust, G., and Noguez, G., Editors, Implementing Functions, EUROMICRO 81, North Holland, Amsterdam, pp. 17-26.
27. Azaria, H., and Tabak, D., 1983, Design Considerations of a Single-Instruction Microcompuer - A Case Study, Microprocessing and Microprogramming, Vol. 11, No. 3/4, pp. 187-194.

28. Greenberg, P., Trabels i, A., and Tabak, D., 1983, Distributed Microcomputer-Based Control of Multiple Signalized Traffic Intersections, Proc. 4th IFAC Conf. on Control in Transdportation Systems, Baden-Baden, F.R. Germany, April 1983.

29. Etkin, J., and Tabak, D., 1985, Microcomputer-Embedded Distributed Control of a Switching and Communication System, in Sinha, N.K., Ed., Microprocessor-based Control Systems, D. Reidel Publ. Co., Dordrecht, The Netherlands.

30. Etkin, J., and Tabak, D., 1984, Method for Implementing Distributed Control of a Communication System, Microprocessing and Microprogramming, Vol. 14, No. 3/4, pp. 181-186.