Processors, FPGAs, SOCs, trends and questions

J. Vásárhelyi József*, D. Drótos*, J. Turán**, J. Végh***
* University of Miskolc/Department of Automation and Communication Technology, Miskolc, Hungary, {vajo, drdani}@mazsola.iit.uni-miskolc.hu
** Technical University of Kosice/ Department of Electronics and Multimedia Communications, Kosice, Slovakia, Jan.Turan@tuke.sk
*** University of Debrecen/Department of Informatics Systems and Networks, Debrecen, Hungary, jvegh@unideb.hu

***Abstract*— The paper analyses embedded systems, processors, and reconfigurable architectures, from the point of view of technological possibilities and possible future directions. There is analysed the integrated circuit technologies from the point of view of Gordon Moor's law. Which factors can influence the embedded systems and reconfigurable architectures in their development from the viewpoint of the user, and are there crises if any?**

I. INTRODUCTION

Integrated circuit makers always care about Gordon Moore's law. In our days when the dimension of the integrated transistors on silicon is approaching to atomic dimensions (See one atom transistor [10]), one can put the question when semiconductor technology will approach its limits.

Gordon Moore published his law in the Electronics Magazine (19th April, 1965). He stated that "the number of transistors on integrated circuits doubles approximately every 18 month" [1]. At the end of 1970th the law was known as the highest limit of the number of integrated transistors in an integrated circuit [1]. The law gives a prediction to the integration density, from which does not result the speed of microprocessor generations, which has more and more register, cache resources. The cited paper [1] gives the wrong conclusion about the relationship of the increasing number of transistors and increasing working frequency of transistors. The density of integrated transistors is the result of the technology, while the increasing speed is the result of circuit optimisation. Gordon Moore's law consequence to the integrated circuit (IC) technology is presented in Figure 1. One cannot conclude from the number of transistors to the microprocessor working frequency. That is true that circuit optimisation is possible as the result of IC technology, but not only.

Microprocessors working frequency is growing faster than as one can conclude from Moore's law. There are different reasons of the working frequency, but one should mention the main difference between the processing speed and working frequency, which is often mixed up.

Often the processing speed is defined as the working frequency of the microprocessor, which definition is completely wrong. Another misunderstanding is when the processing speed is defined as the frequency of the instruction cycle. Furthermore the instruction cycle can vary by the "WAIT" clock cycles introduced by the microprocessor.

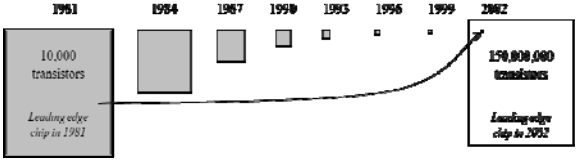

Figure 1. Moore law's prediction to IC density [9]

In our point of view the processing speed (period) is the time needed for the execution of a specified task. This period can be speeded up by operating systems performance, cache memories, pipeline instruction execution and other techniques.

II. EMBEDDED SYSTEM TECHNOLOGIES

Before the analyses of microprocessor working efficiency one have to consider the embedded system design technologies. These design technologies have influence on the embedded system performances and design parameters [9]

Vahid and Giravis [9] defined the technology as the way which makes possible the completion of the design target considering the integrated circuit technologies, processes, techniques, and knowledge.

Under this consideration the embedded system design consider three technologies: processor, IC and design technologies [9].

Processor technology under this definition means the processing element, which makes possible the data processing. In [9] are considered three technologies: general processors, application specific and single purpose processors.

In the paper there are analysed the general purpose processors. General purpose processors are the key element of several embedded systems but they can be also the main central processing unit of several computers (PC).

III. COMPUTER ARCHITECTURES, ADVANTAGES AND DISADVANTAGES

In computer sciences one speaks about two types of architectures. These architectures are the Princeton (von Neumann) and the Harvard architecture.

The Neumann architecture is characterized with common data and program memory. This means that program code and data is loaded in the same bus into the microprocessor. The Harvard architecture has separate

code and data memory. So the main difference between the two computer models is the data and code management. However the Harvard architecture speed up the data processing in PC-s the used processors are von Neumann like.

The advantage of Neumann architecture result in simpler computer architecture, but the processor during the instruction period mainly processes the instruction code. So the Princeton architecture is instruction flow centric.

Looking in the literature one can find hundreds of papers about how to improve the von Neumann structure.

While the Harvard architecture advantage is the parallel instruction and data processing, which in this case is the disadvantage, which result in much complex system architecture with data and code memory. But in some cases the data processing is much faster because of the different buses.

Certainly there exist another machine model called the Kress-Kung machine (anti machine) is data stream based, but not instruction stream based. The Kress-Kung machine (anti machine) has no CPU (Central Processing Unit). If it is hardwired, it has a DPU (Data Processing Unit) instead, or even a DPA (DPU array) See for details [11]. Against its clear advantages and data flow centric architecture, this model was completely disregarded by the computer sciences, only was considered by the reconfigurable architectures community (see Field Programmable Gate Arrays - FPGA).

A. *Microprocessor "tuning" does really help?*

The Neumann architecture still dominates the computer hardware. Let us analyse some improvements on the x86 like architectures, since the first processor was introduced. The bottleneck of the Neumann architecture i.e. the common data and code memory is a serious obstacle in front of the data processing improvements of this model.

The first technique we should mention is *the pipeline instruction execution*. Certainly was an instruction execution improvement, but the relationship between the processor and memory was the same.

The continuously increasing on chip memory solution tries to decrease the frequency of memory read and write cycles. But this does not solve the unbalanced Neumann model problem; only delay a real solution for a new computing model.

The technology advances resulted in higher external memory capacity, which has the result that the software application designers and operating system designers did not care anymore about writing efficient executable code, since the increased memory capacity allowed the wasteful memory usage. This is known as "Bill Gates law", which successfully compensated the results of the Moore law. In this way the memory demand for program execution doubled every two year.

Multicore processors are a solution to the parallel computing still using the Neumann model. The common code and data memory for all the processors in our point of view only narrowed the bus between the memory and processors. Perhaps a solution for this would be as many code and data memories as many cores in the chip, this is at least as complex and costly like increasing cache. Unfortunately the IC technology improvements did not improve the programing efficiency.

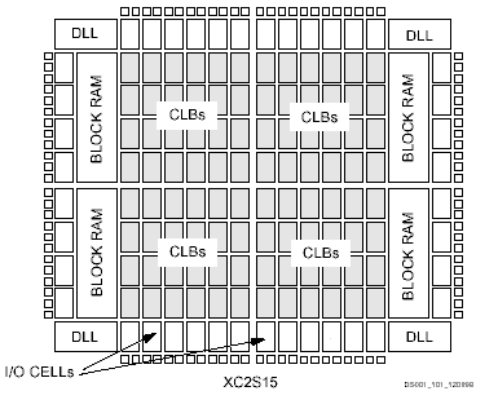

Figure 2. FPGA architecture view

IV. PROGRAMMABLE LOGIC ARCHITECTURE

Field Programmable Gate Arrays (FPGA) was introduced in 1985, and since then they totally changed the digital design and embedded system market. FPGA are usually symmetric high logic resources. The digital design after synthesis, map, translation and place and route the design can be downloaded directly on the chip. The on chip reconfigurable and extensible platform is the result of the development and a consequence of Moore's law.

The user to design its product use the logic configurable logic cells, routing resources, block memory and configurable input output blocks. The advantage of FPGAs is that their logic resources can be reconfigured at any time...

The first dynamically and partially reconfigurable FPGA architecture was the CAL FPGA introduced in 1995, which later becomes the Xilinx 6000 FPGA. This possibility resulted in a completely new research area the configurable computing one.

The FPGA unlike the microprocessor load from the external memory its configuration bit stream in order to complete the data processing. While the microprocessor in order to execute the task, have to load every instruction sequentially from the memory.

The reconfigurable computing still awaiting for find real applications at least we do not know about the existence of to many. But the data flow computer solution with the Kress architecture is a possibility. In this case the power dissipation will result in dynamically reconfiguration. Reconfiguring to often is the danger of this solution.

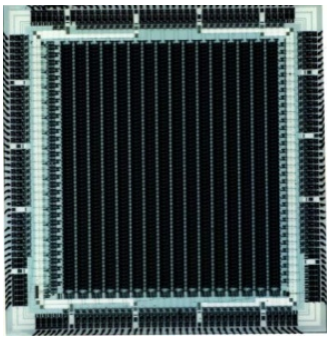

Figure 3. The CAL Architecture [Algotronix]

V. SOFTWARE STUCK UP

The new versions of operating systems (OS) and some programs need more and higher memory capacities, while their functional development is not evolving. These OS and programs packets completely disregard the hardware evolution. This process is known as "software stuck up".

While the hardware in general, processors, memories price is decreasing, and the operating speed/cent ratio is better and better, the software developers by-pass this fact. This resulted in omitting the so called "code optimisation", this is not at all in the focus of development phase.

One of the reasons why is happen this is the fact that software development in not only the privilege of mathematicians, engineers, but is "a right" of the ones who are not versed in software technologies.

Software development environments stand for speed-up of development cycle, and productivity. In this way software development became a copy-paste "programming style" from several modules. In many cases this result in quickie software, which contain only the optimisation included in the basic setup of the development environment. In this way the software remains many times un-optimised.

The software stuck up result in a multi-dimensional space as stated in [2]. The multiple reasons are as follows:

- disregarding the minimal memory needed for the task execution; result in increasing memory needs
- the increasing dimension of software installer program; result in increasing installation time;
- installation trash increased;
- software start-up/stop time increased;
- processor time increased for task execution;
- the need for newer hardware (for example video card);

In this way the benefits, this should result from the new software version completely disappear. [2]

VI. RECONFIGURABLE/EXTENSIBLE PROCESSOR MODEL

The research target is to create completely new processor architecture then one should give up the Princeton or Harvard thinking way. The anti-machine concept is a good starting point, but an all-time reconfigurable machine is the other extremity of the problem.

The FPGA technology allows the run time reconfigurable architecture, also probably even allows the run-time parameterisation of the processing element. This problem needs more research. In this way the processor can adopt to the dataflow needs resulting in an optimal architecture for effective data processing. The procedure seems to be relatively simple, but in the research of this field there is need for a closer analyses. In the actual phase of the research one has to consider the most important research results of the reconfigurable computing. Based on Makimoto's law all the chips will converge in to sing le one wh ich it m ay b e called Multiprocessing Programmable System on Chip MPSoC [7].

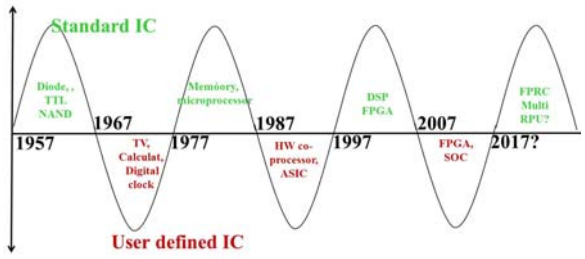

Figure 4. Makimoto's wave

Certainly this architecture is one possible way. The multi core chips need corresponding software optimisation tools. Probably these architectures will generate a total reconsideration of software technologies, which allows run-time dynamically, partially reconfiguration, but not all time reconfiguration.

The question is that should this future solution decide in run time the hardware resources or the hardware resources will be decided in compile time?

The following questions arise when an operating system based new architecture is created:

How can one measure the load of the processor components or multiprocessor cores when a task performs? This question is extremely important when this task is performed by the processor itself (the processor hardware is compute this), without using external software components.

How can one determine that what kind of computation task is processed? The subcomponents load parameters can help this analysis.

How can one define the processor subcomponents needed for execution of a specific task, and which components are useless? This problem can be evaluated by several processor speed tests.

The operating system is needed in order "to simplify" the user and the developer "life". What are the basic tasks like scheduling, interrupt and other functions solved by the hardware? This would speed-up the execution of time critical tasks.

How can be solved the process or run-time reconfiguration in order to optimise it for the fastest system adaptivety? This is a purely technical question, but unsolved yet. Reconfiguration process compared to the processing speed is a slow process.

How can one determine the effectivity of the self-configuring processor? At this point of the research one has to consider the results or other research groups.

In our point of view the new reconfigurable–extensible system should contain a system supervising processor, which can decide and handle the needed resources and allocate them for each task. This allocation is double and is made in hardware and software. Probably there will be standard peripherals. In this way some resources can be estimated in advance.

VII. CONCLUSIONS

In the paper were analysed some advantages and disadvantages of the actual microprocessors. There was considered the microprocessor technology and their influence on the embedded system design.

One can state that the base of future systems will be multi core reconfigurable platforms. The new platform model should be built from the basics.

ACKNOWLEDGMENT

This Research was carried out as part of the TAMOP-4.2.1.B-10/2/KONV-2010-0001 project with support by the European Union, co-financed by the European Social Fund.

REFERENCES

[1] *** Wikipedia, „Moore Törvénye”, http://hu.wikipedia.org/wiki/Moore-t%C3%B6rv%C3%A9ny

[2] *** Wikipedia, „Szoftverek felfuvódása”, http://hu.wikipedia.org/wiki/Szoftverek_felf%C3%BAv%C3%B3d%C3%A1sa

[3] *** Grid Cafe: „A Moore törvény félreértelmezése”, http://gridcafe.eu-egee.hu/Breaking-Moore-law.html

[4] H. Amano “A survey on Dynamically Reconfigurable Processors”, IEICE Transactions on Communication, Vol E-89B No 12, December 2006, pp.3179-3189.

[5] F. Muller, F. Muhammad, „Virtual Platform for Hw RTOS – Multiprocessor Hardware RTOS”, Proceedings of DATE 09 Design, Automation and Test in Europe, 20-24 April, 2009, Nice, France, http://www.date-conference.com/date09/files/file/09-ubooth/Session7/S77.pdf

[6] T. Makimoto, „The hot decade of programmable technologies”, https://www.doc.ic.ac.uk/~wl/teachlocal/cuscomp/k01_makimoto.pdf

[7] J. Becker, “Cyber-physical MPSoC Systems - Adaptive Multi-Core Architectures for future Mobility & Technologies” http://www.cs.tsukuba.ac.jp/~yoshiki/HEART/HEART2012/keynote.html

[8] http://www.dagstuhl.de/Materials/index.en.phtml?10281

[9] F. Vahid, T. Givargis, *Embedded System Design: A Unified Hardware/Software Introduction,* John Wiley & Sons; ISBN: 0471386782, 2002.G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” *Phil. Trans. Roy. Soc. London*, vol. A247, pp. 529–551, April 1955.

[10] M. Fuechsle, J. A. Miwa, S. Mahapatra, H.R yu, S. Lee, O. Warschkow, L.C.L Hollenberg, G. Klimeck, M. Y. Simmons, „A singel atom transistor”, Nature Nanotechnology, 7, doi:10.1038/nnano.2012.21, 2012, pp. 242–246

[11] *** “Anti machine” http://anti-machine.org/

[12] S. Oniga, A. Tisan, C. Lung, A. Buchman, I. Orha, Adaptive hardware-software co- design platform for fast prototyping of embedded systems, 12th International Conference on Optimization of Electrical and Electronic Equipment, OPTIM 2010, May 20-22, 2010, Brasov, Romania, pp.1004-1009