Reconfigurable Mobile Multimedia Systems

Gerard J.M. Smit, Martinus Bos, Paul J.M. Havinga, Jaap Smit
University of Twente
departments of Computer Science and Electrical Engineering
+31 (0)53 4893734
smit@cs.utwente.nl

***Abstract* – This paper discusses reconfigurability issues in low-power hand-held multimedia systems, with particular emphasis on energy conservation. We claim that a radical new approach has to be taken in order to fulfill the requirements - in terms of processing power and energy consumption - of future mobile applications. A *reconfigurable systems-architecture* in combination with a QoS driven operating system is introduced that can deal with the inherent dynamics of a mobile system. We present the preliminary results of studies we have done on reconfiguration in hand-held mobile computers: by having reconfigurable media streams, by using reconfigurable processing modules and by migrating functions.**

***Keywords* – Handheld computers; energy efficiency; reconfigurable computing; multimedia.**

I. Introduction

In the next decade two trends will definitively play a significant role in driving technology: the development and deployment of personal mobile computing devices and the continuing advances in integrated circuit technology. The semiconductor technology will soon allow the integration of one billion transistors on a single chip [3]. This is an exciting opportunity for computer architects and designers; their challenge is to come up with system designs that use the huge transistor budget efficiently and meet the requirements of future applications. The development of personal mobile devices will give an extra dimension, because these devices have a very small energy budget, are small in size but require a performance which exceeds the levels of current desktop computers. It will be shown that state-of-the-art system-architectures cannot provide the wealth of services required by a fully operational mobile computer given the increasing levels of energy consumption. Without significant energy reduction techniques and energy saving architectures, battery life constraints will limit the capabilities of these devices.

A. Personal mobile devices

An exciting prospect for the coming years is the deployment of a new generation of hand-held computers. The technologies of PDA, wireless networking and smartcard, when combined and integrated well, have the potential of replacing all of the things people have to carry around by one small device, that we will call a *Mobile Digital Companion* (MDC). This device is a small portable computer with a smart card and communications device that can replace cash, cheque book, passport, keys, diary, phone/pager, walkman, radio, maps, etc. The MDCs can be used as multimedia terminals to watch a video fragment, to listen to your favourite music as a digital walkman or to take a picture with the on-board camera. In addition, the MDCs will be used as means to participate in an on-line information community. The combination of networking, security and mobility will engender many new applications and services. Not only do they provide the means for users to stay in touch while on the move and to receive notifications of important events, it also gives people a whole new way to interact with the infrastructure of large public institutions, such as interactive class-rooms, airports, supermarkets, or even whole cities. For example: standing in line for ticket or teller windows may become a thing of the past. Instead offices and public places will be equipped with access points, through which hand-held computer users will be able to communicate with the existing infrastructure.

The employment of the envisioned Mobile Digital Companion has several challenging implications:

- It must provide *multimedia functionality*

  It has been predicted that beyond the year 2000, 90 percent of the computer cycles will be spent on multimedia applications [4]. The MDC is an end user terminal so image processing, handwriting and speech recognition will be important and (soft) real-time properties will be evident. An extra challenge is that the system has to deal with limited resources (energy, communication bandwidth, processing power, memory, etc.).

- MDCs work in a very *dynamic* environment.

  The MDC should support *wireless* multimedia communication in a dynamically changing environment. For example, it will have to deal with unpredicted network outage or should be able to change to a different network, without changing the application. It should have the *flexibility* to handle a variety of multimedia services and standards (like different video decompression schemes and security mechanisms) and the *adaptability* to accommodate to the nomadic environment, required level of security, and available resources. Eventually even the user might notice these dynamics: he will have to live with Quality of Service changes, e.g. a lower audio quality or a change from full colour to black/white picture quality.

- MDCs are *personal* devices

  The MDC contains valuable private information such as electronic money, contracts, cryptographic keys, private addresses etc. Furthermore, because MDCs are used in an open and nomadic setting, the MDC communicates with potential hostile and untrusted service providers. For instance, when the user downloads software from an unknown service provider he may be prone to many forms of attack (viruses, Trojan horses).

- MDCs must be *small* and *light*.

  The weight and size should be adequate for its purpose: e.g. a hand-held device should fit into your shirt pocket. This implies that it should have an *ultra low energy consumption*, because only small batteries can be used.

B. Semiconductor technology

The semiconductor technology is realising chips with substantially smaller features each year. This leads to a magnitude shrink (1/10) of all mask-features in ten years. The industry decreased the energy consumption per operation with a factor of 1/1000 in the past decade. Greatly enhanced performance levels has been achieved e.g. due to a 100-fold increase in the clock speed. Functionality has moved from 16-bit integer arithmetic to 64 bit floating point arithmetic. A 100-fold increase in performance can be expected for the decade ahead. Computer architects are already discussing the architecture of future one billion transistor processor designs. In our view, personal mobile computing will play a significant role as a driving technology in processor design. Other researchers [[11] share this view. The two main reasons are the above-mentioned increasing use of multimedia applications and the growing popularity of portable devices. One major obstacle to designing one billion transistor systems is the physical design complexity, which includes the effort devoted to the design, verification and testing of an integrated circuit. A possible solution is to work with a highly regular structure such as the FPFA (Field Programmable Function Array) structure described in section II. These structures only require the design and replication of a single processor tile and an interconnection structure. Design and verification of a regular structure circuit is much easier. Although the precise formulation of such architectures is complex, as the architecture should be optimal for many applications; the great reward is that the verification of its physical design is much more straightforward, due to the restricted use of automatic routing tools. Furthermore, production level testing is less complicated too due to the repetition of well-defined structures.

C. Energy efficiency

In the area of mobile computing it will be an enormous challenge to work with a minimal power budget. Yet, the architecture must provide the performance for functions like speech recognition, audio/video compression/decompression and data encryption. Power budgets close to current high-performance microprocessors, are unacceptable for portable, battery operated devices. MDCs should be able to execute functions at the minimum possible energy cost. On the other hand they must be flexible and adaptable to environment changes.

Today, a lot of research is mainly focused on performance and (low power) circuit design of individual components. We believe it is more effective to save energy by a carefully designed hardware- and software architecture of the mobile. There is a vital relationship between hardware architecture, operating-system architecture, applications' architecture and human-interface architecture. For example: the applications can adapt to the power situation if they have an appropriate operating-system API for doing so; the operating system can optimize the battery consumption by adapting reconfigurable components to the required Quality of Service; the hardware architecture can handle the data in such a way that, for critical functions, only a minimum number of components need to be active. We think progress has to be made in two areas in particular:

- *Reconfigurable system architectures*

  These architectures use the chip area effectively, are relatively easy to design and are flexible and adaptive to handle the dynamics of the mobile environment.

- *Energy aware operating systems*

  MDC's should be flexible and adaptive to the inherent unpredictability of the mobile environment, should be able to control the multimedia streams through the reconfigurable architecture. We think the operating system has to be Quality of Service driven, it has to use a QoS framework to handle the flexibility in a uniform way. Here QoS not only incorporates network performance parameters, but also energy cost and infrastructure cost. Some of these parameters such as energy are 'vertical' controls, they have impact on all layers of the protocol stack, from applications down to the physical layer. Our approach is based on an extensive use of power reduction techniques at all levels of system design.

The remaining part of this paper will address these two main issues in more detail.

II. Reconfigurable Systems Architecture

We believe the previous section gives more than enough evidence for the thesis that a radical new approach in the systems architecture has to be taken in order to fulfill the requirements of the MDC, in terms of processing power and energy consumption. We propose a reconfigurable systems-architecture that in combination with a QoS driven operating system can cope with the inherent dynamics of a mobile environment. The system architecture should be flexible and/or reconfigurable in many ways. The main research question is how this reconfiguration can be structured. This is a rather new research field and to give an impression what kind of reconfigurability we are considering we describe three ways how we think reconfiguration could be done. We do not have the space nor the intention to give an overview of all possible forms of reconfiguration here. In the next sections we will elaborate on the following three reconfiguration methods:

- Reconfigurable media streams,
- Reconfigurable processing modules,
- System decomposition.

A. *Reconfigurable media streams*

In a previous phase of our project Moby Dick [7] we found that in low power systems much energy profit can be gained by improving the component interaction. We experimented with a systems-architecture that accommodated the required functionality, within the energy limitation constrains of a small battery-powered device. This systems-architecture has some similarities with the Desk Area Network in Cambridge [5] and the Pleiades project in Berkeley [1] [2].

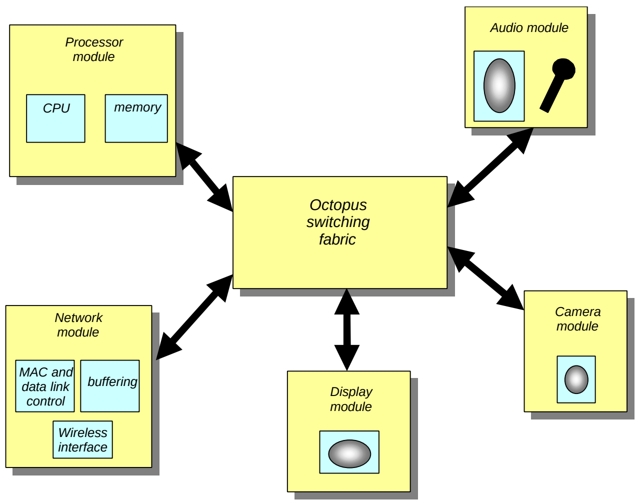

Fig. 1. ***System architecture***

In the architecture, we have an organization of a programmable *communication switch* surrounded by several autonomous modules [5]. Fig. 1 gives a schematic overview of the MDC's architecture. The functional tasks are allocated to dedicated (reconfigurable) modules (e.g. display, audio, network interface, security, etc.). The switch activates only those data paths actually carrying data.

As in switching networks, the use of a multi-path topology will enable parallel data flows between different pairs of modules and thus will increase the performance. In our architecture modules are autonomous and can communicate without involvement of the main processor. For example, if a video/audio stream enters the terminal via the network interface, this data is sent directly to the video/audio module, without main processor intervention. The main processor is used only initially to setup the connection. The architecture has a number of premises:

- An energy efficient communication mechanism for multimedia tasks as well as non-media tasks is provided by a structure of a general-purpose processor accompanied by a set of heterogeneous reconfigurable modules. The modules are capable of performing device or application specific tasks efficiently. They can for example decompress a video stream, just before it is displayed on the screen. Dedicated modules can be optimized to execute specific tasks, with minimal energy overhead. Instead of executing all computations in a general-purpose processor, as is commonly done in conventional PDA architectures, the energy- and computation-intensive tasks are executed in optimized reconfigurable modules.
- A reconfigurable internal communication network exploits locality of reference and eliminates wasteful data copies. Memory accesses consume quite a bit of energy and this energy is wasted if the data only occupies memory in transit between two devices (e.g., network and screen or network and audio).
- The main CPU is relieved of having to service device interrupts and to perform context switches, or to copy buffers to or from a device every time new data arrives.
- The system avoids wasteful activity: e.g. by using of autonomous modules that can be powered down individually and are data driven. The modules can easily *adapt* their behavior to changes in the environment, either imposed by the user (when it starts a new or different application) or by resource changes (for example when the network module notices a change in the wireless channel conditions).
- The modules are *autonomous*. For instance: the wireless communication is designed for low energy consumption by using *intelligent* network interfaces that deal efficiently with a mobile environment, by using a power aware network protocol stack, and in particular by using a energy aware MAC protocol. The network protocol stack can be handled by the network interface such that the CPU can be turned off for frequent media streams.

B. *Reconfigurable processing modules*

Multimedia applications have a high computational complexity, they have a regular and spatially local computation, and the communication between modules is significant. The quest for processors with increased processing power has lead to multi-issue CPU's and speculative instruction pre-fetch strategies, which have driven the general purpose CPU's far away from the energy lower-bound for the processing tasks at hand.

Fig. 2 shows the energy consumption for a single instruction of many microprocessors over the last 10 years. Note that all processors lie in a range, which spans a factor of ten, with a few exceptions, which are actually low-power prototypes. The lower bound for the calculation of a multiply-add operation is shown in the left bottom by the line named 16x16 MAdd. The actual application gap is at least 40 for the 33MHz 5V Intel 486, 240 for the Motorola 68040 and even 700 for the first Intel Pentium processor. The trend is that even with better technology, the energy consumption to perform a single instruction increases.

The factor 1000 increase of performance for the decade to come cannot be realized through an increase of the clock-speed with a factor 100, due to physical limitations. Hence it will be necessary to extend the parallelism of the devices. This can be done through the use of multiple ALUs on one hand and a cache memory on the other hand.

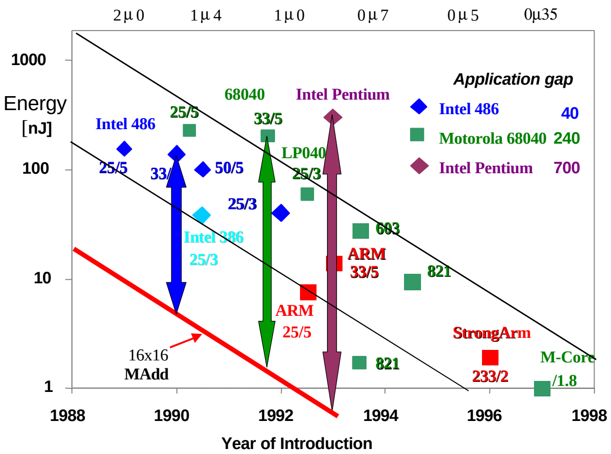

Fig. 2. Energy consumption and application gap

The most common alternative is to use a full custom design style. Application-specific coprocessors perform multimedia tasks more efficient - in terms of performance and/or energy consumption - than general-purpose processors. Even when the application-specific coprocessor consumes more power than the processor, it may accomplish the same task in far less time, resulting in net energy savings. The processor can for example be offloaded with tasks like JPEG and MP3 decoding, encryption, and some network protocol handling. An MPEG chip can handle video much more efficient than a general-purpose processor. However, this option is getting less and less attractive. The main reasons are: the fixed schedule in the high-level synthesis, the related effect that the design is not scalable, and the costly design process which does not support any form of real-time prototyping. In our opinion this will lead to a rapid acceptance of a totally new design styles based on reconfigurable devices.

The difference in area and power dissipation between a general-purpose approach and application specific architectures can be significant. Full custom chips can be designed and manufactured at relatively low cost. However, this comes at the price of less flexibility, and consequently a new chip design is needed for even the smallest change in functionality.

A hybrid solution with *application domain specific modules* can offer the flexibility that allows the implementation of a predefined set of (usually) similar applications, while keeping the costs in terms of area, energy consumption and design time to an acceptable low level [9]. The modules are optimized for one specific application domain. Fig. 3 shows three different approaches in the spectrum of hardware organizations.

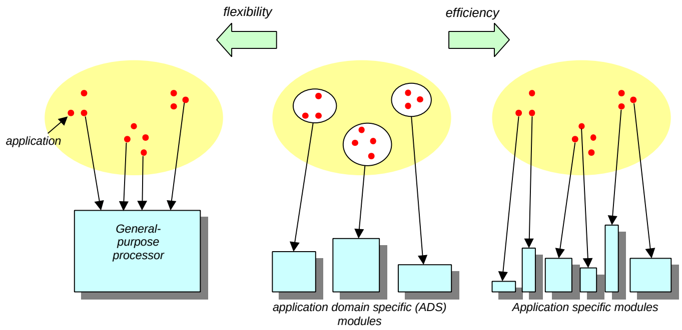

*Fig. 3: **The spectrum of hardware organisations [9].***

We believe that the functional requirements of future mobile devices including the adaptability and flexibility of various system functions (both in terms of performance and energy) can be implemented using *energy-efficient reconfigurable modules.* Today there are commercially available *Field Programmable Gate Arrays* (FPGA).

They operate as a field-programmable graph of 1-bit-wide lookup tables (LUTs) or CLBs [8]. It can be shown that the construction of an ALU from multiple 1-bit-wide lookup tables is energy inefficient. For a wide range of multimedia functions that use digital filtering algorithms on parallel data: video (de)compression, data encryption and digital signatures these devices do not posses the required processing power. For these functions 16/32 bit calculations (multiply, add) are required. We have experimented with a structure called FPFAs (*Field-Programmable **Function** Array).* These devices are reminiscent to FPGAs, but with a matrix of ALUs and lookup tables [8] instead of CLBs (Configurable Logic Blocks).

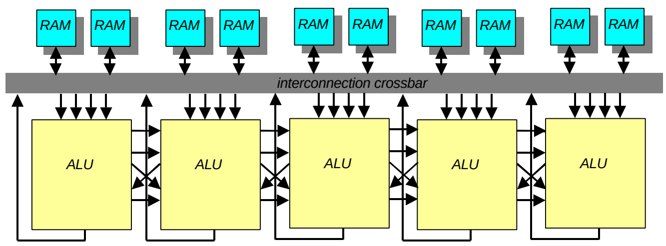

*Fig. 4: **FPFA architecture***

The instruction set of an FPFA-ALU can be thought of as the set of ordinary ALU instructions, with the exception that there are no load and store operations which operate on memories. Instead, they operate on the programmable interconnect; that is, the ALU loads its operands from neighboring ALU outputs, or from (input) values stored in lookup tables or local registers. Hence, these devices use the locality of reference principle extensively.

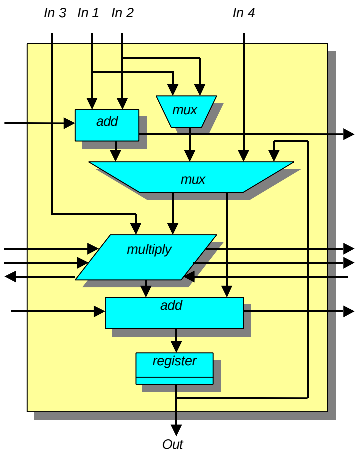

Fig. 5. ***FPFA ALU***

The graph-based execution of the FPFA is used to execute the inner loop of an application. The regular, general-purpose structure of the device makes a rapid context switch from one inner loop to another possible, hence on-the-fly reconfiguration. This is how a broad class of compute intensive algorithms can be implemented on an FPFA. Several non-trivial algorithms have been mapped successfully to the FPFA families introduced. Examples are a Super Resolution Volume Rendering application, shading, texture mapping and an FFT, to name just a few. The FPFA concept has a number of advantages:

- The FPFA has a highly regular, it requires the design and replication of a single processor tile, hence the design and verification is rather straightforward. The verification of the software might be less trivial. Therefore, for less demanding applications we use a general-purpose processor core.
- Its scalability stands in contrast to the dedicated chips designed nowadays, where it takes considerable effort to implement circuitry for tasks such as Digital Audio Broadcast and Digital TV. In FPFAs, there is no need for a redesign of a scalable chip in order to exploit all the benefits of a next generation CMOS process or the next generation of a standard.
- The FPFA can do media processing tasks such as compression/decompression efficiently. Multimedia applications can benefit from compression by saving (energy-wasting) network bandwidth. This requires however an energy-efficient platform to perform the compression.

C. *System decomposition*

The design of hand-held multimedia computers cannot be done in isolation. With high-speed wireless networks, many different architectural choices become possible, each with different partitioning of functions between the hand-held and the servers resident in the network. Partitioning is an important architectural decision, which dictates where applications can run, where data can be stored, the complexity of the mobile and the cost of communication services [10].

For example: in traditional systems most communication protocol functions are implemented on the main processor of the mobile. A consequence is that the network interface and the main processor must always be ‘on’ for the network to be active. Therefore mobile devices consume a lot of their energy in the ‘idle’ mode, waiting for packets to come in. Decomposition of the network protocol stack and a careful analysis the data flow in the system can reduce the energy consumption considerably. A (programmable) dedicated processor of the network module can handle most of the lower levels of the protocol stack much better, thereby allowing the main processor to sleep for extended periods of time without affecting system performance or functionality.

III. QoS Driven Operating System Architecture

The operating system for the Mobile Companion has to deal with the peculiarities of the MDCs, their flexibility and adaptability and their energy restrictions. Applications for the MDC will be used in a variety of computing environments. Many applications are now designed for particular computing environments like personal computers or set-top boxes or a specific handheld, all with static performance. But in the MDC applications will have to run in environments that differ dramatically in processor performance, communication performance and communication cost. Such applications will have to adapt their behavior to the environment in which they run. The operating system will have to provide assistance for this adaptation, now called Quality of Service (QoS). This term stems from the notion that the quality of service an application can deliver depends on the resources that can be made available to it.

Traditionally, QoS is used in the context of network communication resources and systems resources needed for multimedia applications. In mobile-computing environments this notion of QoS has to be extended to all applications. An important issue is that all applications must deal with energy efficiency of a handheld multimedia device. Applications can deliver better QoS when the hardware they run on is in a higher energy state. So there is a QoS tradeoff between performance and battery life. Adaptability, flexibility and interoperability will be crucial for the entire system: from hardware components up to application programs.

A power model is needed to predict the power consumption of MDC designs in order to allow a fast and flexible design of the low-power central processing unit(s) and the related multimedia/protocol coprocessor(s). A careful power analysis of the architecture of all the system-level components is needed for the successful design of the next generation of hand-held devices. It will be necessary to judge the design of the CPU, multimedia-processing units, and related peripherals in terms of their ability to conserve energy, as hardware components on one hand and as programmable components - controlled by the core functions in the operating system - on the other hand. The net energy consumption should be as low as possible for a given semiconductor technology.

A QoS driven operating system integrates QoS management into every software module, and all modules are responsible for the collection of the QoS management information they require. In the design of a module, it is important to express both the resources it needs from other modules and the adaptation that is required based on what resources the module actually gets. The design of software modules for the MDC therefore focuses on co-operation and adaptation issues rather than just performance.

A hierarchical QoS model of the whole system (covering the architecture, communication, distributed processing, and applications) can be used to adapt to the changing operating conditions dynamically in the most (energy) efficient way. Besides the functional modules and their ability to adapt (e.g. the effects on its energy consumption and QoS when the image compressor changes its frame rate, its resolution, or even its compression mechanism) this model also includes the interaction between these modules. Such a model is required to predict the overall consequences for the system when an application or functional module adapts its QoS. Using this model the inherent trade-offs between e.g. performance and energy consumption can be evaluated and a proper adaptation of the whole system can be made. Together with the fact that the new architecture will include reconfigurable hardware in all modules, the aforementioned rises some challenging research questions.

An operating system must be created that can handle distributed computation and process migration. As the architecture includes programmable hardware, migration includes moving from software to hardware computation and vice-versa. Migration must also be possible to and from remote servers when this is more efficient. Extensive real-time capabilities are necessary for handling continuous-media data (e.g. phone calls or video presentations) and are also useful in providing the operating system with information on current and future workload, which is needed in decision-making for QoS changes. The needed integrated QoS management which effects all layers of the system further complicates the operating system tasks. Also challenging is that all this management must happen online as well due to the possibly rapidly changing environment.

IV. CONCLUSION

Personal mobile computing will play a significant role as a driving technology in processor design. Neither contemporary architectures nor state-of-the-art technology can provide the wealth of services required by a fully functional mobile multimedia computer. The increasing levels of performance and integration that is required will be accompanied by increasing levels of energy consumption. Without significant energy reduction techniques and energy saving architectures, battery life constraints will limit the capabilities of a Mobile Digital Companion. Furthermore it is known that mobile systems work in a very dynamic environment. We claim that a *flexible* and *reconfigurable* systems-architecture in combination with a QoS driven operating system is needed to deal with the inherent dynamics of a mobile system. This reconfigurability can be found in the interaction of multimedia devices, in the media processing and in migration of functionality.

REFERENCES

[1] Abnous A., Rabaey J., "Ultra-Low-Power Domain-Specific Multimedia Processors," *Proceedings of the IEEE VLSI Signal Processing Workshop*, San Francisco, October 1996.

[2] Abnous A., Seno K., Ichikwaw Y., Wan M., Rabaey J.: "Evaluation of a Low-Power Reconfigurable DSP architecture", *Proc. 5th Reconfigurable Architectures workshop (RAW 98),* March 1998.

[3] Burger D., Goodman J., "Billion-Transistor Architectures", *IEEE Computer*, September 1997.

[4] Dally W., "Tomorrow's Computing Engines", keynote speech, 4th Int. Symposium on *High-Performance Computer Architecture*, Feb. 1998.

[5] Havinga P.J.M., Smit G.J.M.: "Octopus: embracing the energy efficiency of handheld multimedia computers" , *proceedings fifth annual ACM/IEEE international conference on mobile computing and networking (Mobicom'99)*, pp.77-87, August 1999.

[6] Hayter M.D., McAuley D.R.: "The desk area network", *ACM Operating systems review*, Vol. 25 No 4, pp. 14-21, October 1991.

[7] Smit G.J.M., et al.: "An overview of the Moby Dick project", *1st Euromicro summer school on mobile computing*, pp. 159-168, Oulu, August 1998.

[8] Smit J., et al, "Low Cost & Fast Turnaround: Reconfigrable Graph-Based Execution Units", *Proceedings Belsign Workshop*, 1998.

[9] Leijten J.A.J.: "Real-time constrained reconfigurable communication between embedded processors", *Ph.D. thesis, Eindhoven University of Technology*, November 1998.

[10] Lettieri P., Srivastava M.B.: "Advances in Wireless Terminals", *IEEE Personal Communications*, pp. 6-19, Feb. 1999.

[11] Patterson D.A., Kozyrakis C.E., "A new direction for Computer Architecture research", *IEEE Computer*, November 1998.