Low-Power Reconfigurable Architectures for High-Performance Mobile Nodes

Matthias Hanke, Tim Kranich, Mladen Berekovic, Yannis Papaefstathiou

Technische Universität Braunschweig, Institut für Datentechnik,
Hans-Sommerstraße 66, 38100 Braunschweig, Germany
{hanke, kranich, berekovic}@ida.ing.tu-bs.de

Telecommunication Systems Institute, TU Chania, Greece
ygp@ece.tuc.gr

Abstract. Modern embedded systems have an emerging demand on high performance and low power circuits. Traditionally special functional units for each application are developed separately. These are plugged to a general purpose processors to extend its instruction set making it an application specific instruction set processor. As this strategy reaches its boundaries in area and complexity reconfigurable architectures propose to be more flexible. Thus combining both approaches to a reconfigurable application specific processor is going to be the upcoming solution for future embedded systems.

Keywords: reconfiguration, ASIP, RASIP, low power, high performance, video encoding, encryption, wireless sensor node, mobile device

1 Introduction

Embedded systems increasingly ease our everyday life. Mobile devices like cell phones, music and video players unrecognizably are driven by high-performance digital circuits. Simultaneously their advanced integration makes their handling become more intuitive for their users. Along with the emerging distribution of handy embedded system devices the demand on their performance and application stock increases. More functionality and higher capacity requirements go along with more complex, faster circuits and thus higher power dissipation. Furthermore bigger circuits let chip yields drop and result in higher costs.

Mobile devices in a remote environment have do rely on limited power sources. Accordingly their circuits are ought to be extremely power efficient. Especially the ubiquitous use of cell phones pushes the development of low-power techniques. But also other remote devices like wireless sensor nodes or space satellites require energy aware solutions. Furthermore pollution control and growing energy costs are upcoming issues resulting in green technology products which force to safe power even in stationary devices like home computers or servers in data centers.

Traditionally system designers have equipped general purpose processors (GPPs) with special functional units (SFUs) separately for each application (**Fig. 1**). The

Dagstuhl Seminar Proceedings 10281
Dynamically Reconfigurable Architectures
http://drops.dagstuhl.de/opus/volltexte/2010/2837

result is an application specific processor (ASIP) with extended instruction set. Deploying such an ASIP allows to overcome performance bottlenecks and improve power consumption and efficiency. However some disadvantages come with this approach.
Any additional functional unit installed for new design demands uses up area on the chip. A larger chip area leads to a decreasing chip yield and thus higher costs. But especially for the huge market of consumer electronics pricings are one of the major constraints. Bigger chips also have the consequence of longer wires which cause higher communication delays and need stronger more power consuming drivers.

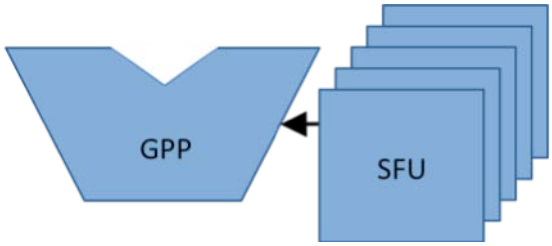

Fig. 1 General purpose processor with multiple special functional units

Typical system on chip designs consist of several different ASIPs with multiple SFUs each of them targeting and optimized for another functional block of an application. These have to exchange data amongst each other. Each processor added to support emerging demands brings higher complexity to the communication interconnect. Accordingly the design of such systems becomes very time consuming.

Highly specialized circuits are extremely inflexible as they are usually designed for one single application. This becomes obvious as disadvantage when porting a design to new application domains. Slight changes in algorithms or software might make a SFU not longer fit its domain. Accordingly complete SFUs have to be redesigned from the ground up.

One way to overcome these limitations is to deploy reconfigurable arrays instead of a SFU or even a whole set of SFUs hence allowing for a larger flexibility in mapping a vaster range of algorithms to the hardware. This approach allows the use of a standardized ASIP template that can be used to implement several function blocks such as audio and video. It refers to many concepts of coarse grained reconfigurable arrays (CGRAs) and is termed as reconfigurable ASIP (RASIP). Currently the concept is being demonstrated within the context of the European Artemis project SMART for use in low-power wireless sensor nodes supporting streaming video and encryption.

2 Related Work

High effort has been spent for research of CGRAs and there are multiple projects targeting different applications applying manifold approaches. Granularity for

reconfigurable arrays may vary in bit length as well as in the complexity of functional units and interconnect.

An approach that uses very small functional units with a simple interconnect is ADAPTO [1]. It targets digital signal processor (DSP) operations on fixed word length and is able to perform basic logic operations on bit level with a unidirectional direct connection.

A more complex design is SmartCell [2]. It addresses streaming operations. The functional units are able to perform different multi bit operations. A hierarchical structured complex interconnect allows high data rates.

PACT-XPP's [3] hierarchical communication structure is to similar to SmartCell's. Its functional blocks have a comparable complexity. But it offers another new feature - self-reconfigurability. The device is able to reconfigure it self due to application requests.

Along with KRESS arrays [4] came the possibility to map algorithms described in high level programming language to reconfigurable architectures. Its compiler architecture is able to analyze high level program code to optimize it and map it to any hardware template described in its own hardware description language.

All of them share a problem that they have in common with most other CGRA concepts. They are only designed for a limited set of algorithms and not for whole applications which need general purpose instructions for example to run an operating system.

ADRES [5] goes one step ahead. It combines a very large instruction word (VLIW) processor with a reconfigurable matrix. It is a template to create a VLIW processor with reconfigurable matrix suitable for any chosen application.

Concerning low-power issues multiple works have explored that the amount of registers and register accesses directly influences power consumption. As follows in order to save energy the amount of register accesses has to be kept low and the register file size has to be as small as possible [6, 7].

3 RASIP Architecture

The SMART sensor node [8] for wireless networks is designed to have low-power capabilities for image processing and cryptographic sensor applications. As especially video codices but also cryptographic standards vary its hardware has to suit future demands. It has to be sufficient flexible to support new algorithms in these application domains. Additionally its components shall operate on low power to guarantee enduring remote applicability. Therefore it is planned as reconfigurable architecture.

The central processing element will be realized in two versions. One version will have a commercial, partly reconfigurable field programmable gate array (FPGA) as computing element. Version two is planned to be equipped by a RASIP. Compared to the FPGA solution our RASIP will be more power efficient and enables faster adoption to upcoming video and cryptographic algorithms.

Increased power efficiency compared to the FPGA is reached by less complex, smaller computational elements on system level. Furthermore coarser grained

function blocks consume less energy than fine grained FPGA structures on component level.

Fast adoption comes from the programmability in high-level ANSI-C programming language where algorithms for the FPGA have to be completely ported to hardware description language (HDL) for bitstream creation. In contrast to the FPGA with its fine grained configurability the RASIP is coarse grain configurable. Reconfiguration is depends on the complexity level of functional blocks. The RASIP remains efficient if the protocol changes are moderate compared to the original design target.

From a system perspective the RASIP is a composition of two cores (**Fig. 2**). The host core is responsible for system administration and control tasks. It provides access to a set of low bandwidth interface for the radio link, generic sensor connection and the system memory. The second core, called hive core, is connected via an on-chip bus. Because of the 16bit nature of the selected host core (MSP430) and the 32bit nature of the hive core a bus width adoption is required. The hive core shall handle all computationally intensive tasks like video encoding and data encryption. Raw video data is fed directly to the hive core through a high bandwidth streaming port.

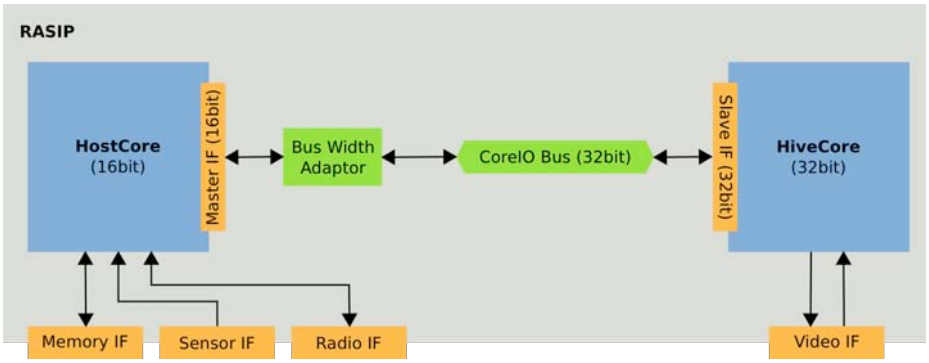

Fig. 2. Block diagram of the SMART RASIP architecture

The data path of the hive core is similar to a VLIW processor. Its parallel processing power arises from the multiple instruction multiple data (MIMD) concept. Unlike a VLIW architecture the data is not directly written back but can sustain in intermediate register files (**Fig. 3**). Additionally the function units are not limited typically standard operations. A detailed investigation of the target applications, i.e. video encoding and cryptography, shall propose some common or often required operation like special shifts, comparisons or permutations.

Reconfigurability lies within the hive core's data path. A dedicated compiler maps ANSI-C code to HDL described functional block templates. Resulting C libraries can be included in program code executed on the host core. Predefined functions reconfigure the hive core data path according to active precompiled algorithms.

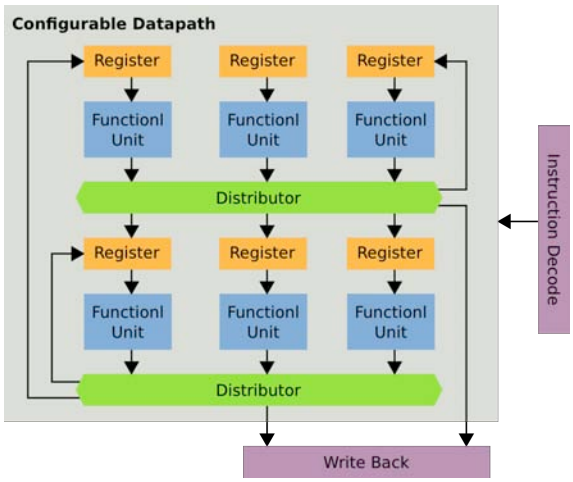

Fig. 3. Data path structure of the reconfigurable core

Cryptographic Algorithms

In difference to cable connections which can be directly protected by structures like fences and walls the radio connections between wireless devices are in a media which is relatively easy to access from everywhere within its range. Thus for wireless devices it is essential to be equipped with techniques ensuring secure communication channels. These have to be persistent against any kind of attack for preventing unauthorized network access. Otherwise sensitive data might be corrupted, modified or stolen causing immense financial damage.

The basic protection is encrypting data with a secret key to prevent anyone eavesdropping the network traffic from accessing the clear data. State of the art and widely utilized is the advanced encryption standard (AES) algorithm [9]. Its very regular highly parallelizable loops contain intensive shift, multiplication and XOR operations.

For the encrypted communication between two partners both need information about each others key. Usually each partner holds his own private key that is never made visible. It is possible to derive a public key from the private one. It is practically nearly impossible to recover the secret private key from the public one. Nevertheless by exchanging their public keys both partners are able to create a secret shared key which can be used to encrypt and decrypt the exchanged data. This procedure is known as public key exchange. Although an attacker might capture the public keys he can not conclude the secret shared key from them hence this requires at least one of the secret private keys.

Elliptic curve cryptography (ECC) is the upcoming technique for such key exchanges. Compared to traditional approaches like RSA it needs shorter keys while offering the same security [10]. Thus key operations become less complex effecting

less power consumption. Especially remote and mobile devices will profit from this fact. During key operations very big integer values of at least 160Bit are performed.

When exchanging keys the partners have to rely on that they really communicate with each other. Otherwise an intruder who is able to manipulate the network could pretend being a communication partner and offer his own keys to the partners. Consequently he would be able to control the complete traffic between both of them. To protect the key exchange against intruders the communication partners can authenticate to each other. Cryptographic hash functions are included in authentication methods. SHA-1 [11] for example is mainly based on shift and parallelizable XOR operations.

Video Algorithms

SMART's main sensor is a video camera. The RASIP shall encode and compress the video before it is transmitted to a base station. The project targets high video quality and low power consumption at the same time. We chose H.264 [12, 13] as video standard. It's efficient compression for high quality images makes it a suitable candidate. The dedicated hardware of the hive core shall be optimized for its computationally intensive operations. H.264 in its full extent is fairly complex. By reducing the encoder parameter set we plan to establish an embedded version, which still offers sufficient compression capabilities for our target application like surveillance and observance of borders, habitats, building, etc. Typical restrictions are reducing the number of prediction modes, choosing one dedicated entropy-encoding scheme or limiting the number of prohibited encoding iterations. Image processing consists mainly of matrix calculations, transforms and filtering. Each algorithm core and its implementation needs to be investigated for its common operations. If we think about the ¼ pixel accuracy for motion estimation for an example, we know that a 6-tap FIR is proposed in H.264. This type of filter relies on several multiply-accumulate (MAC) operations. A common encoder error calculation is the mean square error which includes MAC operations, too.

Conclusion

Power-efficient hardware architectures for mobile nodes are ought to imply dedicated properties. They have to offer general purpose instructions for running operating systems. Furthermore they have to contain power efficient SFUs for CPU intensive tasks.

Reconfigurable arrays allow to remap new applications and modified algorithms on existing hardware which does not have to be redesigned as follows. The array's function blocks have to be as coarse grained as possible to reduce register accesses and thus power. Furthermore they have to be has fine grained as necessary to offer flexibility.

Basic structures for mobile devices according to cryptographic and video algorithms should be arranged in parallel data paths to offer high performance

processing for regular loops. Furthermore they are supposed to offer optimized structures for shift, multiplication for normal and extended size integer values, XOR and MAC operations.

References

1. Cardarilli, GC and Di Nunzio, L. and Re, M.: A full-adder based reconfigurable architecture for fine grain applications: ADAPTO. In: 15th IEEE International Conference on Electronics, Circuits and Systems, pp. 1304--1307. ICECS (2008)
2. Cao, L. and Xinming, H.: SmartCell: an energy efficient coarse-grained reconfigurable architecture for stream-based applications. In: EURASIP Journal on Embedded Systems. Hindawi Publishing Corporation. (2009)
3. Baumgarte, V. and Ehlers, G. and May, F. and Nueckel, A. and Vorbach, M. and Weinhardt, M.: PACT XPP—a self-reconfigurable data processing architecture, Journal of Supercomputing, vol 26, no 2, pp. 167--184, Springer (2003)
4. Hartenstein, R.W. and Kress, R.: A datapath synthesis system for the reconfigurable datapath architecture, Proceedings of the 1995 conference on Asia Pacific design automation, pp. 77 (1995)
5. Mei, B. and Vernalde, S. and Verkest, D. and De Man, H. and Lauwereins, R.: ADRES: An architecture with tightly coupled VLIW processor and coarse-grained reconfigurable matrix. In: Field-Programmable Logic and Applications, pp. 61--70. Springer (2003)
6. Hu, Z. and Martonosi, M.: Reducing register file power consumption by exploiting value lifetime characteristics, Workshop on Complexity-Effective Design (WCED) (2000)
7. Wehmeyer, L. and Jain, MK and Steinke, S. and Marwedel, P. and Balakrishnan, M.: Analysis of the influence of register file size on energy consumption, code size, and execution time, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol 20, no 11, pp. 1329--1337 (2001)
8. Ladis, E. and Papaefstathiou, I. and Marchesani, R. and Tuinenbreijer, K. and Langendorfer, P. and Zahariadis, T. and Leligou, HC and Redondo, L. and Riesgo, T. and Kannegiesser, P. and others: Secure, Mobile Visual Sensor Networks Architecture, 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops, 2009. SECON Workshops' 09, pp. 1--3 (2009)
9. FIPS, N.: Announcing the advanced encryption standard (AES). Information Technology Laboratory, National Institute of Standards and Technology (2003)
10. Hankerson, D.R. and Vanstone, S.A. and Menezes, A.J.: Guide to elliptic curve cryptography, Springer-Verlag New York Inc (2004)
11. NIST, F.P.U.B.: 180-1: Secure Hash Standard, National Institute of Standards and Technology (1995)
12. International Telecommunication Union: Advanced video coding for generic audiovisual services, Series H: Audiovisual and Multimedia Systems (2010)
13. Wiegand, T. and Sullivan, G.J. and Bjontegaard, G. and Luthra, A.: Overview of the H. 264/AVC video coding standard, IEEE Transactions on circuits and systems for video technology, vol 13, no 7, pp. 560--576 (2003)