IMPLEMENTATION OF A 2-D 8X8 IDCT ON THE RECONFIGURABLE MONTIUM CORE

L.T. Smit, G.K. Rauwerda

Recore Systems

P.O. Box 77, 7500 AB,
Enschede, The Netherlands
email: lodewijk.smit@recoresystems.com

A. Molderink, P.T. Wolkotte, G.J.M. Smit

Department of Computer Science,
University of Twente
P.O. Box 217, 7500 AE,
Enschede, The Netherlands
email: a.molderink@utwente.nl

ABSTRACT

This paper describes the mapping of a two-dimensional inverse discrete cosine transform (2-D IDCT) onto a word-level reconfigurable Montium® processor. This shows that the IDCT is mapped onto the Montium tile processor (TP) with reasonable effort and presents performance numbers in terms of energy consumption, speed and silicon costs. The Montium results are compared with the IDCT implementation on three other architectures: TI DSP, ASIC and ARM.

1. INTRODUCTION

Recore's Montium Tile Processor (TP) is a dynamically reconfigurable IP core for computational intensive DSP algorithms. The Montium TP can be used as an accelerator to offload DSP tasks from a processor or in a heterogeneous multiprocessor system. ASIC-like performance and energy-efficiency is obtained by configuring the Montium TP with the functionality required by the algorithm at hand. The Montium TP can be reconfigured almost instantly, as the size of the configuration binaries is very small. Yet, it has a low silicon cost, as the core is very small.

The same silicon area of the Montium TP can be time-multiplexed or reused for very different applications. Time-multiplexing results in smaller chip area (and thus lower costs). An interesting application area for time-multiplexing applications are multi-standard devices. Reuse reduces the design costs (by using off-the-shelf components), the non-recurring costs (due to larger volumes of the same chip) and substantially reduces the time to market.

In this paper, we will investigate the mapping of a two-dimensional inverse discrete cosine transform (2-D IDCT) to the Montium TP. The 2-D IDCT is a frequently used algorithm and numerous implementations are available. This enables comparing the 2-D IDCT implementation on the Montium architecture with implementations of the same algorithm on other architectures. The IDCT is a significant component in today's JPEG and MPEG decoders. Of all the stages in the decoding process of a JPEG file, the IDCT is the most computationally intensive [1]. For JPEG and MPEG-4 decoding a 2-D 8x8 IDCT is used.

Being able to map existing well-known algorithms to a new architecture is vital. This 2-D IDCT mapping exercise gives valuable information about:

- **ease of mapping** - how much effort is required to map algorithms to the Montium?
- **insight into the architecture** - how suitable is the architecture for the mapping of specific algorithms?
- **performance of the architecture** - how many cycles are required to execute the algorithm and what is the power consumption?

2. MONTIUM TP CORE

The Montium TP is a 16-bit word level reconfigurable architecture that obtains significant lower energy consumption than DSPs for fixed-point digital signal processing algorithms. The Montium TP targets computational intensive algorithm kernels that are dominant in both power consumption and execution time. In contrast to a conventional DSP, the Montium TP does not have a fixed instruction set, but is configured with the functionality required by the algorithm at hand. In particular, the Montium TP does not have to fetch instructions and, hence, does not suffer from the Von Neumann bottleneck. Once configured, the Montium TP resembles more an ASIC than a DSP. The Montium TP can be reconfigured almost instantly, as the size of the configuration binaries is very small. The size of a typical configuration is less than 1 KB and reconfiguration typically takes less than 5 μs. The Montium TP has a low silicon cost, as the core is very small. For instance, the silicon area of a single Montium TP with 10 KB of embedded SRAM is 2.4 mm^2 in 0.13 μm CMOS technology. The power consumption in this technology is approximately 500 μW/MHz (including all memory accesses). A Montium TP consists of 5 identical ALUs (see Figure 1) to exploit spatial concurrency in order to enhance performance. See [2] for a detailed architecture description of the Montium TP.

1-4244-1060-6/07/$25.00 ©2007 IEEE.

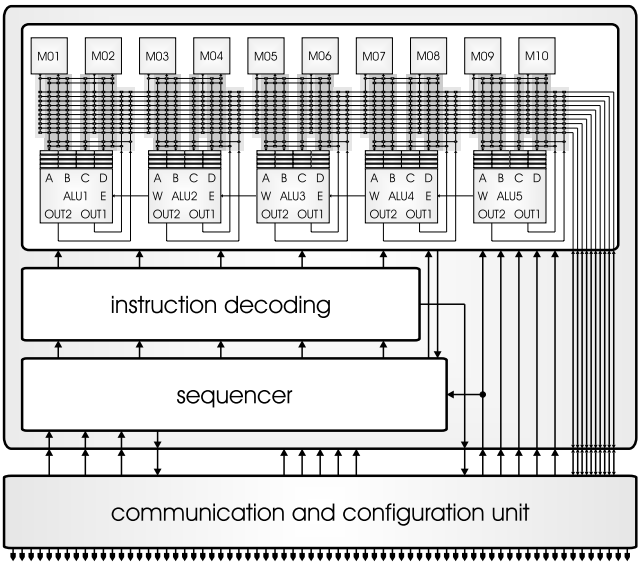

Fig. 1. Montium Processing Tile

3. MONTIUM 2-D IDCT IMPLEMENTATION

This section describes the implementation of a 2-D 8x8 IDCT on the Montium. A 2-D IDCT can be obtained by applying first a 1-D IDCT over the rows, followed by a 1-D IDCT over the columns of the input data matrix [3]. The definition of a 8-point 1-D IDCT is:

$$f_x = \sum_{u=0}^{7} C_u \times F_u \times \cos\left(\frac{(2x+1)u\pi}{16}\right), \qquad (1)$$

where F_u are input samples, f_x output samples, C_u is a constant and $x \in [0..7]$ is the index in output vector f_x.

We use Chen's method [4] for a 1-D (i)DCT implementation, as there is a good balance between regularity and number of operations required. Chen derived a way of implementing a 1-D IDCT with 16 multiplications and 26 additions, without complicated memory operations or divisions. The whole 2-D 8x8 IDCT implemented with the Chen algorithm requires 256 multiplications ($16 \cdot 16$) and 416 additions ($16 \cdot 26$). The 5 ALUs of a Montium TP are able to perform multiple operations per clock cycle and can generate up to two outputs per ALU. Furthermore, it is possible to use results from one ALU in another (neighboring) ALU in the same clock cycle (which is actually used in our implementation). Table 1 shows the mapping of the IDCT operations to the Montium. Each column in this table represents a different ALU; each row represents a different clock cycle. The mapping of the 1-D IDCT takes 4 clock cycles. So, 8 times 1-D IDCT takes $8 \cdot 4 = 32$ clock cycles. In order to perform the 1-D IDCT on the columns, the rows need to be transposed into columns. This operation requires 16 Montium clock cycles. After execution of the second 1-D IDCT we need another transpose operation (rows to columns) requiring another 16 clock cycles. Therefore, the whole 2-D 8x8 IDCT requires 96 clock cycles on the Montium (= 1.5 cycles per input sample).

Mapping Effort

Studying the IDCT and selecting the most suitable algorithm for implementation required the most effort and took about 2.5 weeks. The mapping of the 2-D 8x8 IDCT to the Montium processor took about 1.5 weeks. The mapping was performed by an MSc student, who had no substantial prior knowledge of IDCTs or the Montium architecture. Afterwards we spend another few days for optimization. We expect that further optimization is possible by pipelining the transform operations.

4. RESULTS

To benchmark the Montium performance, we decided to make a comparison between the Montium and three other architectures for the following reasons:

a) In literature, MPEG-4 implementations based on an ARM processor are often combined with hardware accelerators. Therefore, we decided to benchmark the Montium versus the ARM. In other words, what is the gain obtained by offloading computationally intensive processes from the ARM (Ahmdahl's law)?

b) An ASIC implementation has extreme characteristics. It is the best choice of all architectures in terms of performance and energy consumption. However, it is the worst choice of all architectures in terms of flexibility, non-recurring costs and time-to-market. A benchmark of the Montium versus an ASIC gives an idea of how close the performance is to the best lower bound. In other words, what is the price of the flexibility?

c) Finally, the Montium is benchmarked against a Texas Instruments DSP. This represents one of today's most likely design choices. This shows how the reconfigurable approach compares to a conventional DSP solution.

The benchmarks only considers computational power and does not consider communication. Much of the communication latency can be hidden by overlapping of communication time and computing time (also referred to as "streaming" communication).

This section explains the results that are presented in Table 2. We benchmark on two criteria: energy and performance. The latter is normalized by chip area to express the silicon costs. These benchmarks are depicted in the last two rows of Table 2 and also in the Figures 2 and 3.

The approach is as follows.

1) We determine the number of clock cycles required to execute a 2-D 8x8 IDCT. Next, we determine the energy consumption per clock cycle for the architecture. To make a

ALU 1	ALU 2	ALU 3	ALU 4	ALU 5
$X_2 = F_1 \times C_7 - X_1$	$X_1 = F_7 \times C_1$	$X_8 = F_7 \times C_7 + X_7$	$X_7 = F_1 \times C_1$	$X_9 = F_0 \times C_4$
$X_{4a} = X_2 + (F_5 \times C_3 - X_3)$ $X_{4b} = X_2 - (F_5 \times C_3 - X_3)$	$X_3 = F_3 \times C_5$	$X_{6a} = X_8 + (F_3 \times C_3 + X_5)$ $X_{6b} = X_8 - (F_3 \times C_3 + X_5)$	$X_5 = F_5 \times C_5$	$X_{10a} = X_9 + F_4 \times C_4$ $X_{10b} = X_9 - F_4 \times C_4$
$X_{16a} = X_{10b} + (F_2 \times C_6 - X_{11})$ $X_{16b} = X_{10b} - (F_2 \times C_6 - X_{11})$	$X_{11} = F_6 \times C_2$	$X_{15a} = X_{10a} + (F_6 \times C_6 + X_{13})$ $X_{15b} = X_{10a} - (F_6 \times C_6 + X_{13})$	$X_{13} = F_2 \times C_2$	$X_{17} = X_{4b} \times C_4$
$f_1 = X_{16a} + (X_{6b} \times C_4 + X_{17})$ $f_6 = X_{16a} - (X_{6b} \times C_4 + X_{17})$	$f_0 = X_{15a} + X_{6a}$ $f_7 = X_{15a} - X_{6a}$	$f_2 = X_{16b} + (X_{6b} \times C_4 - X_{17})$ $f_5 = X_{16b} - (X_{6b} \times C_4 - X_{17})$	$f_3 = X_{15b} + X_{4a}$ $f_4 = X_{15b} - X_{4a}$	

Table 1. Each row of this table contains IDCT operations that are performed in 1 clock cycle of the Montium, where X_i is a temporary intermediate result and f_u and F_u refer to the variables in Eq. 1 and C_u is a constant for the cosine expression in Eq. 1.

fair comparison, the power figures for each architecture are normalized to 0.13μm technology, using a nominal voltage of 1.2V. Finally, the energy consumption per IDCT is computed by multiplying the number of clock cycles required for the execution of a 2-D 8x8 IDCT with the energy consumption per clock cycle.

2) To benchmark the silicon costs, we first determine the number of 2-D 8x8 IDCTs that can be executed per second. However, it is evident that doubling the chip area will increase the performance. Therefore, we normalize this number to mm^2 chip area. Thus, the measure is the number of 2-D 8x8 IDCTs that can be executed per second per mm^2 chip area. There is a strong correlation between the chip area and the chip cost price. Therefore, this is a good measure for the production cost effectiveness of an architecture for the IDCT. Note that non-recurring design costs are not included in this measure (which can be substantial).

4.1. Montium

Section 3 shows that the Montium needs 96 cycles for a 2-D 8x8 IDCT. The energy consumption per clock cycle for the Montium is obtained by power simulations on a gatelevel netlist. This estimation has an accuracy of less than 10% difference compared to a realization. The energy consumption per Montium clock cycle is 0.5nJ in 0.13μm 1.2V CMOS technology [5]. The energy consumption per 2-D 8x8 IDCT is $96 \cdot 0.5 = 48$ nJ. The Montium can run on 100 MHz in 0.13μm technology. At this frequency, it can execute $\frac{100 \cdot 10^6}{96} = 1.042$M 2-D IDCTs per second. Normalization gives $\frac{1042 \cdot 10^3}{2.4} = 434$k IDCTs per second per mm^2. Note that we used a Montium TP with 10KB memory. This is a lot more memory than required for the 2-D 8x8 IDCT (current memory utilization was below 3%). This means that if the Montium TP memory capacity is tailored to the 2-D 8x8 IDCT, the area will be much smaller.

4.2. Dedicated 2-D 8x8 IDCT ASIC

To compare the performance of the Montium with a dedicated IDCT application specific integrated circuit (ASIC) we looked for a state-of-the-art reference implementation in the literature. In [1] an ASIC implementation of an IDCT is presented, which is identical to our IDCT implementation. This ASIC is implemented in TSMC 0.18μm technology and has a power consumption of 634.5 mW at the maximum frequency of 154 MHz. The Montium power estimates are made for 0.13μm technology. According to [6] it is possible to estimate the energy consumption for a smaller technology. The common dependency of the dynamic power consumption is that it is linearly related to the total capacitance and frequency and quadratically related to the voltage. With reduction from 0.18 μm to 0.13 μm the capacitance goes down with a factor $\frac{0.18}{0.13}$. The ASIC requires a nominal voltage of 1.8V, while the estimations of the Montium are based on 0.13μm technology with a nominal voltage of 1.2V. This makes it reasonable to assume that the power consumption decreases with a factor $(\frac{1.8}{1.2})^2 \cdot \frac{0.18}{0.13} = 3.12$. An estimation of the energy consumption per clock cycle in 0.13μm is $\frac{634.5 \cdot 10^{-3}}{154 \cdot 10^6 \cdot 3.12} = 1.32$ nJ. This ASIC implementation needs 30 clock cycles per 2-D 8x8 IDCT [1]. The area of the ASIC is 12.17 mm^2 in 0.18μm technology [1]. In TSMC technology, the gates density is 100 and 200 kgates per mm^2 for 0.18μm and 0.13μm technology respectively [7, 8]. After normalization to 0.13μm technology, the area of the ASIC becomes about $12.17 \cdot \frac{100}{200} = 6.09$mm^2. According the ITRS roadmap [9] the max. clock frequency scales with a factor 1.4 per technology generation. Therefore, the max. frequency of this ASIC will be around $154 \cdot 1.4 = 216$ MHz in 0.13μm . This results in $\frac{216 \cdot 10^6}{30 \cdot 6.09} = 1182$k 2-D IDCTs per mm^2 per second.

4.3. Texas Instruments TMS320C6454 DSP

In order to compare the Montium implementation with a state-of-the-art digital signal processor (DSP), we chose the Texas Instrument (TI) TMS320C64xTM DSP platform. This DSP platform was launched in December 2006. We selected the high performance TMS320C6454-720 for comparison. According to [10] the power consumption is 1.18W @ 1.2V with 60% utilization. Note that we only consider the power consumption used for the internal logic and not the total power consumption, which is including the I/O (memory access). Assuming a linear function [5] between power consumption and utilization, we expect 1.97W @ 1.2V for

100% utilization. The TMS320C6454-720 is produced in 0.09μm technology [11]. We use the same method as in the previous subsection to normalize to 0.13μm technology for a fair comparison. This results in an increase of power consumption with a factor of $(\frac{1.2}{1.2})^2 \cdot \frac{0.13}{0.09} = \frac{13}{9}$. An estimation of the energy consumption per clock cycle in 0.13μm is $\frac{1.97}{720 \cdot 10^6} \cdot \frac{13}{9} = 3.95$ nJ.

TI provides a library with imaging functions. We expect that these functions are highly optimized. According to [12], the number of clock cycles for a 2-D 8x8 IDCT is $72 \cdot n + 63$, where n is the number of IDCTs. We assume that it is realistic to execute 6 IDCTs sequentially, as MPEG-4 uses 6 IDCTs per macroblock. So, on average $\frac{72 \cdot 6 + 63}{6} = 82.5$ clock cycles per 2-D 8x8 IDCT are used. The estimation of the energy consumption for a 2-D 8x8 IDCT in 0.13μm is $3.95 \cdot 82.5 = 325$ nJ.

TI does not disclose the area of the TMS320C6454-720. Therefore, we have to estimate the area. At the International Electron Devices Meeting in fall 1999, TI presented a roadmap [13], which reveals that a TMS320 DSP core will contain ca. 100M transistors in 2005. As the number of transistors follows Moore's law, this prediction should be quite accurate despite the long forecast period. We assume that the TMS320C6454-720 has about 100M transistors. This is a conservative estimation, as this number was estimated for 2005, while this DSP has been launched in 2006.

To know the number of transistors (T) per mm^2, we investigated 0.13μm TSMC technology parameters [14]. We distinguish between memory and logic gates, because the density is quite different. For SRAM memory the density is 2.43 - 2.14 μm (6 transistors). This means a density of $6/2.43\mu = 2.46$MT/mm^2. TSMC gate density is 219k gates/mm^2. With 4 transistors per gate, the transistor density is 219k $\cdot$ 4 = 0.88 MT/mm^2. We compared the densities to UMC technology parameters and the numbers are quite similar.

We now make an area estimation of the TMS320C6454-720 using 0.13μm TSMC technology parameters. This DSP contains 8 Mbit SRAM cache requiring 48M transistors, which is equivalent to an area of 20mm^2. The remaining 52M transistors require $\frac{52M}{0.88} = 59$mm^2. Total estimation of area is therefore 79mm^2 in TSMC 0.13μm technology.

Normalization of max. frequency to 0.13μm gives a speed of $\frac{720}{1.4} = 514$MHz. Therefore, $\frac{514 \cdot 10^6}{82.5} = 6.23$M 2D 8x8 IDCT/s can be executed. This results in $\frac{6.23 \cdot 10^6}{79} = 79$k 2D 8x8 IDCT per mm^2 per second.

4.4. ARM 946E-S

We also implemented the Chen implementation of the 2-D 8x8 IDCT algorithm on the Advanced Risc Machines (ARM) 946E-S processor. The implementation is coded in C and compiled to binary code using the GNU GCC 3.4.3 compiler. This compiler was cross-compiled to generate code for exactly the right type of ARM instruction set (ARMv5TE) to ensure best optimizations. To make accurate measurements, we used a dedicated hardware clock to measure the number of clock cycles for the execution of 396 2-D IDCTs. Use of a dedicated hardware clock ensures a high accuracy. The ARM 946E-S needs 2796 clock cycles to execute one 2-D 8x8 IDCT with the cache enabled. Because we expected a higher performance, we decided to make a new implementation using hand optimized assembly parts that exploits the DSP extensions (e.g. 1 cycle MAC) of the ARM. This did not improve the performance. Careful examination of the assembly code revealed that the real bottleneck was due to the limited number of available general purpose registers combined with a slow (at least 2 cycle delay) cache access.

	ASIC	Montium	TI	Arm
max freq. [MHz]	154	100	720	200
power@max fr.[mW]	634.5	50	1967	92
technology [μm]	0.18	0.13	0.09	0.13
Voltage [V]	1.8	1.2	1.2	1.2
area [mm^2]	12.17	2.4	n/a	1.86
cycles / 2-D 8x8 IDCT	30	96	495/6	2796
energy/clock cycle				
@ 0.13μm [nJ], 1.2V	1.34	0.5	3.95	0.46
area in 0.13μm [mm^2]	6.1	2.4	$\approx$79	1.86
max. fr. 0.13μm [MHz]	216	100	514	200
# 2-D IDCT/s in 0.13μm	7.2M	1.0M	6.2M	0.072M
energy/2-D IDCT [nJ]	**40**	**48**	**325**	**1286**
# 2-D IDCT/mm^2/s	**1182k**	**434k**	**$\approx$79k**	**38k**

Table 2. Characteristics and benchmarking of 2-D 8x8 IDCT in terms of energy and area on four different architectures

The ARM946-S (optimized for area) has a maximum frequency of 200 MHz and an area of 1.86mm^2 (including cache) in 0.13μm. The power consumption is 0.46mW/MHz [15]. The energy consumption per 2-D IDCT is $2796 \cdot \frac{0.46 \cdot 10^{-3}}{1 \cdot 10^6} = 1286$ nJ. The number of 2-D IDCTs that can be executed per second is $\frac{200 \cdot 10^6}{5173} = 71530$. Normalized, this are $\frac{71530}{1.86} = 38$k 2-D IDCTs per second per mm^2.

5. CONCLUSION

The benchmarks in this paper provide important information to make a fair trade-off between different architectures.

The Montium Tile Processor (TP) offers much more flexibility than an ASIC, while being much more energy-efficient than a conventional DSP. The Montium performs near energy efficient as an ASIC but uses more silicon area. However, the Montium area can be reused for different functions by means of time-multiplexing due to the offered flexibility while an ASIC is restricted to one dedicated algorithm. Therefore, the Montium is an attractive alternative to

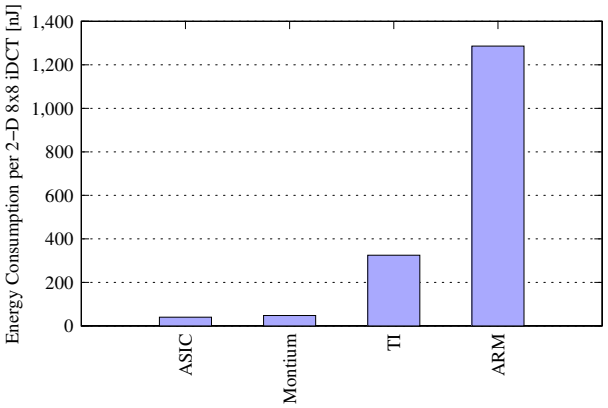

Fig. 2. Energy Consumption for a 2-D 8x8 IDCT on different Architectures after Normalization to 0.13μm [nJ]

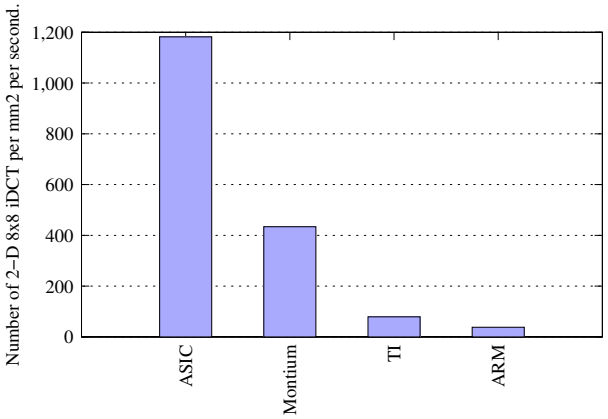

Fig. 3. Area efficiency (silicon costs) in number of 2-D 8x8 IDCTs per mm^2/s on different Architectures

an ASIC due to the offered flexibility, the fast time to market and the lower costs. The Montium outperforms a conventional DSP solution both in terms of energy-efficiency and in silicon costs expressed in number of 2-D 8x8 IDCTs per second per mm^2 area. As expected, an ARM is not an efficient solution for the implementation of an IDCT.

The mapping of the 2-D 8x8 IDCT did not reveal any shortcomings of the Montium architecture. These kind of mappings give valuable insight into architectural improvements. The reconfigurable Montium architecture is mature and provides a good balance between efficiency and flexibility.

Considering the scenario where the person who did the mapping had no prior knowledge (about algorithm nor architecture), we can conclude that mapping kernels to the Montium TP can be done with reasonable effort (in this example the coding took about eight days). Use of the Montium TP provides a much faster time to market compared to the use of a dedicated ASIC, while being cheaper because the Montium IP can be reused for different applications.

Acknowledgement

This research is conducted within the FP6 Smart ChipS for Smart Surroundings (4S) project (IST-001908) supported by the European Commission. Thanks to Arjan Boeijink for improving our initial Montium IDCT mapping.

6. REFERENCES

[1] R. Swamy, M. Khorasani, Y. Liu, D. Elliott, and S. Bates, "A fast, pipelined implementation of a two-dimensional inverse discrete cosine transform," in *Proc. of Canadian Conference on Electrical and Computer Engineering*, May 2005, pp. 665–668.

[2] P. M. Heysters, G. J. M. Smit, and E. Molenkamp, "Montium - balancing between energy-efficiency, flexibility and performance," in *Proceedings of ERSA'03, Las Vegas, USA*, June 2003, pp. 235–241.

[3] C. Loeffler, A. Ligtenberg, and G. Moschytz, "Practical fast 1-d dct algorithms with 11 multiplications," in *Proceeding of International Conference on Acoustics, Speech, and Signal Processing (vol 2)*, May 1989, pp. 988–991.

[4] W. Chen, C. Smith, and S. Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Transactions on comm.*, vol. 25, pp. 1004–1009, Sept. 1977.

[5] P. Heysters, "Coarse-grained reconfigurable processors: Flexibility meets efficiency," Ph.D. dissertation, University of Twente, Sept. 2004.

[6] Committee on Networked Systems of Embedded Computers, *Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers*. National Academy Press, 2001, iSBN: 0-3090-7568-8.

[7] "Tsmc 0.18-micron technology." [Online]. Available: http://www.tsmc.com/download/enliterature/018_bro_2003.pdf

[8] "Tsmc 0.13-micron technology." [Online]. Available: http://www.tsmc.com/download/enliterature/013_bro_2003.pdf

[9] "International technology roadmap for semiconductors," 2003.

[10] G. Martinez, *Application Report: TMS320C6455/C6454 Power Consumption Summary*. Texas Instruments, Sept. 2006, document: SPRAAE8A.

[11] *Data sheet: TMS320C6454 Fixed-Point Digital Signal Processor (Rev. A)*. Texas Instruments, Dec. 2006, document: SPRS311A.

[12] *TMS320C64x+ DSP Image/Video Processing Library Programmers's Reference*. Texas Instruments, 2006, literature Number: sprueb9.

[13] Http://www.elecdesign.com/Articles/Index.cfm?AD=1&ArticleID=1004.

[14] Http://www.tsmc.com/download/english/a05_literature/Advanced_Technology_Overview_Brochure_2006.pdf.

[15] Http://www.arm.com/products/CPUs/ARM946E-S.html.