ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

An Efficient VLSI Architecture for Software Defined Radio by Using Montium Processing Tile

A.Saida
Assistant Professor
ECE Department
KG Reddy College of Engineering And Technology,
Moinabad,TS,India.
e-mail: angotusaida2@gmail.com

Abstract: ***Future wireless communication systems tend to become multimode, multifunctional devices. Adaptivity becomes more important now than ever. These systems have to adapt to changing environmental conditions (e.g., more or less users in a cell or varying noise figures due to reflections or user movements) as well as to changing user demands [bandwidth, traffic patterns, and quality-of-service (QoS)]. When the system can adapt (at run-time) to the environment, significant savings in computational costs can be obtained. Furthermore, the hardware architectures have to be extremely efficient as these are used in battery-operated terminals and have to be cost effective as they are used in consumer products. Heterogeneous reconfigurable hardware platforms offer the necessary flexibility for performing multiple wireless communication standards and can achieve the performance required by the wireless standards. Furthermore, the combination of mixed-grained reconfigurable solutions enables energy efficient implementations of the wireless standards. Much work has been done on software defined radio (SDR) in the SDR forum context. One of the main reasons for introducing reconfigurable hardware in a wireless terminal is to support multiple wireless communication standards. The support of multiple wireless communication standards introduces a first level of adaptivity in the wireless terminal because the terminal can switch between wirelesses communications standards.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.***

Keywords: ***Heterogeneous reconfigurable hardware, orthogonal frequency division multiplexing (OFDM), software defined radio (SDR), system-on-chip (SoC), wideband code division multiple access (WCDMA). Xilinx ISE Foundation 9.1i***

I. INTRODUCTION

A **software-defined radio** system, or **SDR**, is a radio communication system where components that have been typically implemented in hardware (e.g. mixers, filters, amplifiers, modulators/demodulators, detectors, etc.) are instead implemented by means of software on a personal computer or embedded computing devices. While the concept of SDR is not new, the rapidly evolving capabilities of digital electronics render practical many processes which used to be only theoretically possible.

A basic SDR system may consist of a personal computer equipped with a sound card, or other analog-to-digital converter, preceded by some form of RF front end. Significant amounts of signal processing are handed over to the general-purpose processor, rather than being done in special-purpose hardware. Such a design produces a radio which can receive and transmit widely different radio protocols (sometimes referred to as a waveforms) based solely on the software used. Software radios have significant utility for the military and cell phone services, both of which must serve a wide variety of changing radio protocols in real time. In

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

the long term, software-defined radios are expected by proponents like the SDR Forum (now The Wireless Innovation Forum) to become the dominant technology in radio communications. SDRs, along with software defined antennas are the enablers of the cognitive radio.

1.1 Existing Systems:

Homogeneous flexible architecture like DSP processor or GPPS.A Multicore Software-Defined Radio(SDR) architecture for Global Navigation Satellite System (GNSS) receiver implementation. A GNSS receiver picks up very low power signals from multiple satellites and then uses dedicated processing to demodulate and measure the exact timing of these signals from which the user's position, velocity, and time (PVT) can be estimated. In the GPPS are flexible but inefficient and have relatively poor performance. Application Specific architecture are efficient, it shows good performance but are inflexible.

1.2 Proposed System:

Heterogeneous reconfigurable hardware: It consisting of processing elements of different granularities is design with these constraints flexibility performance and energy efficiency in mind. The idea of heterogeneous processing elements is that one can match the granularity of the algorithm with the granularity of the hardware. Coarse Grained: these devices are flexible at word level; multipliers, adders, etc are hardware in these devices. Because only coarse functional block have to be configured, the configuration our head is small. Example is Montuim tile processor.

The energy-efficiency of the system can be improved significantly by executing computational kernels2 on algorithm domain specific hardware. To satisfy above requirements we can design and develop Montium tail-processor in hardware implementation for SDR.

1.3 Problem Definition

The existing system working at 100 MHz's at this frequency level the processing speed is low. In homogeneous flexible architecture: It's relative energy inefficiency. It does not have a balance between the energy efficiency, flexibility and performance.

2. DIFFERENT DSP PROCESSOR

2.1 Introduction

A **digital signal processor** (**DSP**) is a specialized microprocessor with an optimized architecture for the fast operational needs of digital signal processing.

Digital signal processing algorithms typically require a large number of mathematical operations to be performed quickly and repetitively on a set of data. Signals (perhaps from audio or video sensors) are constantly converted from analog to digital, manipulated digitally, and then converted again to analog form, as diagrammed below. Many DSP applications have constraints on latency; that is, for the system to work, the DSP operation must be completed within some fixed time, and deferred (or batch) processing is not viable.

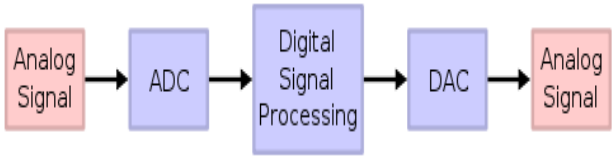

Fig 2.1 Simple DSP processor

Most general-purpose microprocessors and operating systems can execute DSP algorithms successfully, but are not suitable for use in portable devices such as mobile phones and PDAs because of power supply and space constraints. A specialized digital signal processor, however, will tend to provide a lower-cost solution, with better performance, lower latency, and no requirements for specialized cooling or large batteries.

The architecture of a digital signal processor is optimized specifically for digital signal processing. Most also support some of the features as an applications processor or micro controller, since signal processing is rarely the only task of a system. Some

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

useful features for optimizing DSP algorithms are outlined below.

2.2 Architecture:

By the standards of general purpose processors, DSP instruction sets are often highly irregular. One implication for software architecture is that hand optimized assembly is commonly packaged into libraries for re-use, instead of relying on unusually advanced compiler technologies to handle essential algorithms.

Hardware features visible through DSP instruction sets commonly include:

- Hardware modulo addressing, allowing circular buffers to be implemented without having to constantly test for wrapping.
- Memory architecture designed for streaming data, using DMA extensively and expecting code to be written to know about cache hierarchies and the associated delays.
- Driving multiple arithmetic units may require memory architectures to support several accesses per instruction cycle.

3. IMPLEMENTATION

3.1 Montium Processing Tile:

The MONTIUM processing tile targets the 8-bit digital signal processing (DSP) algorithm domain. A single MONTIUM processing tile is depicted in Figure 3.1. The lower part of Figure 4.1 shows the Communication and Configuration Unit (CCU) and the upper part shows the reconfigurable Tile Processor (TP). The CCU implements the interface for off-tile communication. The definition of the off-tile interface depends on the interconnect technology that is used in the FPGA. The figure 3.1 reveals that the hardware organization of the tile processor is very regular. The five identical ALUs (ALU1…ALU5) in a tile can exploit spatial concurrency to enhance performance. This parallelism demands a very high memory bandwidth, which is obtained by having 10 local memories (M01…M10) in parallel. The small local memories are also motivated by the locality of reference principle. The data path has a width of 8-bits and the ALUs support both signed integer and signed integer arithmetic.

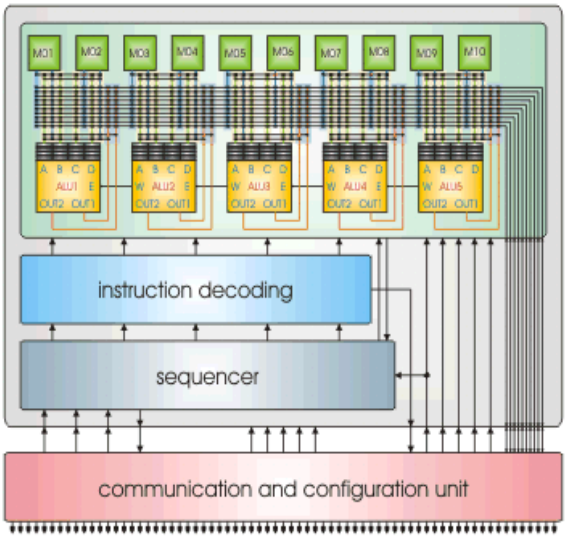

Figure 3.1: The MONTIUM processing tile

The ALU input registers provide an even more local level of storage. Locality of reference is one of the guiding principles applied to obtain energy-efficiency in the Montium. A vertical segment that contains one ALU together with its associated input register files, a part of the inter connect and two local memories is called a Processing Part (PP). The five Processing Parts together are called the Processing Part Array (PPA). A relatively simple sequencer controls the entire PPA. The sequencer selects configurable PPA instructions that are stored in the decoders of Figure 3.1.

3.2 The MONTIUM TP ALU:

Each local memory is a single-ported SRAM with a capacity of 8-bit. A reconfigurable Address Generation Unit (AGU) accompanies each memory. It is also possible to use the memory as a lookup table for complicated functions that cannot be calculated using an ALU, such as sine or division (with one constant). A memory can be used for both integer and fixed-point lookups.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

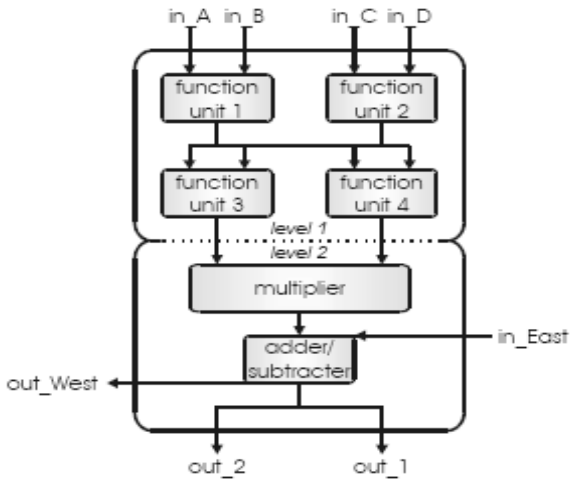

Fig 3.2: The Montium tile processor arithmetic logic unit

Figure 3.2 depicts a single MONTIUM TP ALU. It has four 8-bit inputs and each input has a private input register file that can store up to four operands. The input register file cannot be bypassed, i.e., an operand is always read from an input register. Input registers can be written by various sources via a flexible interconnect. An ALU has two 16-bit outputs, which are connected to the inter connect. The ALU is entirely combinational and consequentially there are no pipeline registers within the ALU. Neighboring ALUs can also communicate directly on level 2. The west-output of an ALU connects to the east-input of the ALU neighboring on the left. The east west connection does not introduce a delay or pipeline, as it is not registered.

The Montium Tile Processor (TP) is a programmable 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.

- Primary requirements of wireless multimedia handheld computers are high-performance, flexibility, energy-efficiency and low costs.
- A compromise for these contradicting requirements can be found in a heterogeneous SOC.
- The MONTIUM is a prototype of a novel coarse grained reconfigurable processor. The SOC template offers a balance between flexibility, efficiency and performance.

The hardware organization of the tile processor is very regular. The five identical ALUs (ALU1…ALU5) in a tile can exploit spatial concurrency to enhance performance. This parallelism demands a very high memory bandwidth, which is obtained by having,10 local memories (M01…M10) in parallel. The ALU input registers provide an even more local level of storage. Locality of reference is one of the guiding principles applied to obtain energy-efficiency in the Montium.

3.3 MEMORY DESIGN STYLE:

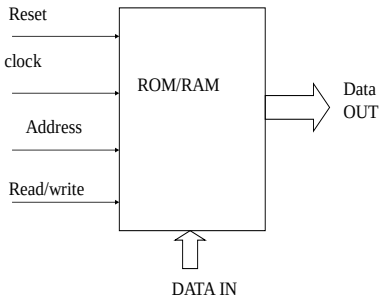

Figure no 3.3 Memory design styles

The above figure no 3.3: shown the total operation dependent on the address and read/write operations. If memory is empty there is need first fill the memory (write data) and then access the data from memory. The total operation takes 2n clock cycles: n-clock cycles for write and n-clock cycles to read. It is a time consuming process.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

3.5 Linear Feedback Sift Register:

A linear feedback shift register (LFSR) is a shift register whose input bit is a linear function of its previous state. The only linear function of single bits is XOR, thus it is a shift register whose input bit is driven by the exclusive-or (XOR) of some bits of the overall shift register value.

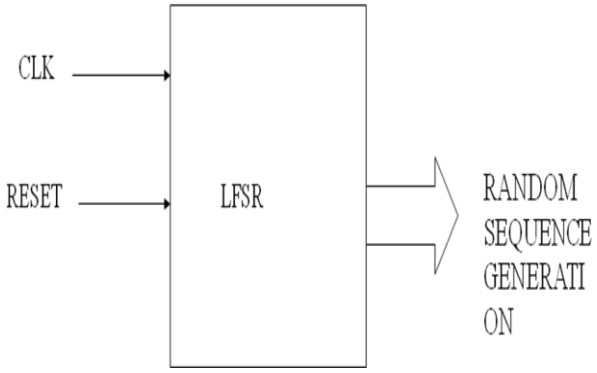

Figure no 3.5a: linear feedback shift register black diagram

The initial value of the LFSR is called the seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current (or previous) state. Likewise, because the register has a finite number of possible states, it must eventually enter a repeating cycle.

However, an LFSR with a well-chosen feedback function can produce a sequence of bits which appears random and which has a very long cycle.

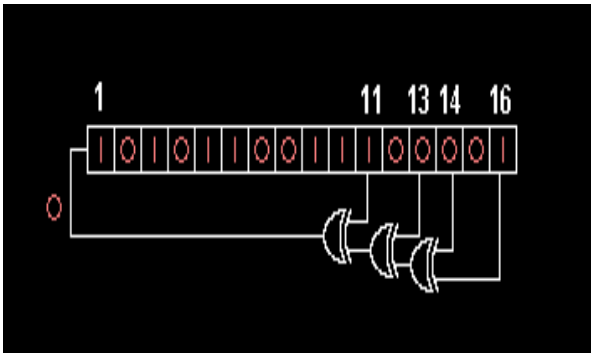

Figure no 3.5b: 16 bit LFSR black diagram

.

A 16-bit Fibonacci LFSR. The feedback tap numbers in white correspond to a primitive polynomial in the table so the register cycles through the maximum number of 65535 states excluding the all-zeroes state. The state ACE1 hex shown will be followed by 5670 hex.

4. RESULT ANALYSIS

1. Each architecture is coded in Verilog and simulated.
2. The memory locations are implemented by verified using coding.
3. The simulation results and the memory location simulation are verified for each architecture.

The chapter deals with the implementation of memories, the significance of various locations, reports obtained and simulation waveforms of architectures developed to implement memory decoder and LFSR methods.

4.1 Memory Decoder Architecture For Implementation Of Four Level Memory Locations Architecture:

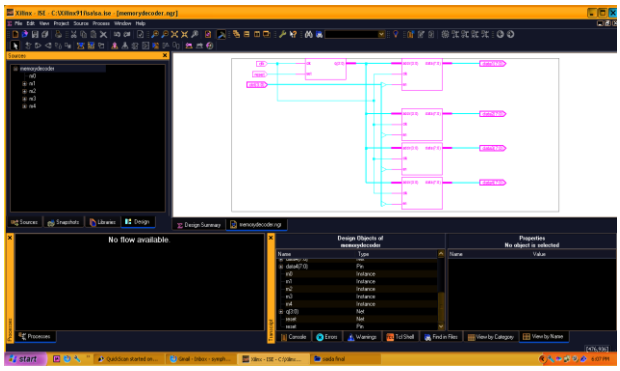

Fig no 4.1a: Memory decoder RTL systematic

A priority encoder is a circuit or algorithm that compresses multiple binary inputs into a smaller number of outputs. They are often used to control interrupt requests by acting on the highest priority request. If two or more inputs are given at the same time, the input having the highest priority will take precedence. An example of a single bit 4 to 2 encoder is

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

shown, where highest-priority inputs are to the left and "x" indicates either a 1 or 0.

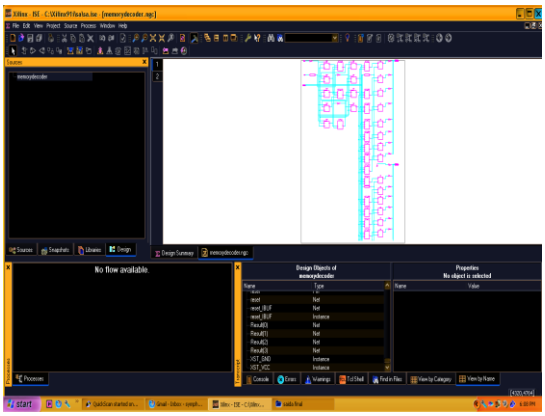

Fig no 4.1b: Memory decoder technology systematic

The figure 4.1b shows the memory decoder technological systematic. This is suitable for serial applications. By using counter it will generate sequence to decoder, the decoder out contains at maximum one only. The counter is a asynchronous counter. It increments its count value at negative clock edge. If the communication medium is very slow at that situation is suitable.

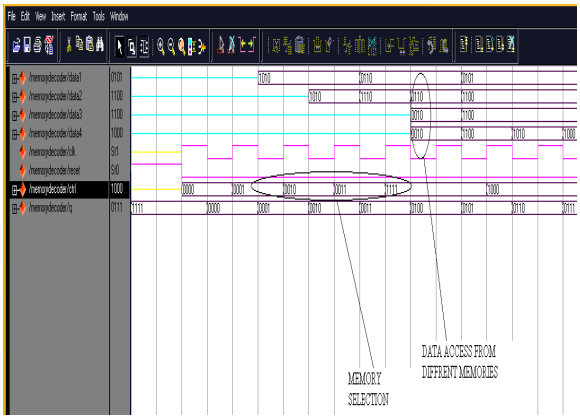

Figure no 4.1c: simulation result for memory decoder

The above figure4.1c simulation result for memory decoder which shows the four memory locations of on a memory decoder. In this case 0010 or 1000 etc are input bit to the memory decoder, gives the memory location one, 1100 or 0011 etc are input bit to the memory decoder, gives the memory location two, 1110 or 0111are input bit to the memory decoder, gives the memory location three and finally all ones give the memory location four.

4.2 Linear Feedback Shift-Register Architecture For Implementation Of Four Level Memory Locations:

The figure no 4.2 LFSR modules RTL systematic gives the register have a finite number of possible states; it must eventually enter a repeating cycle.

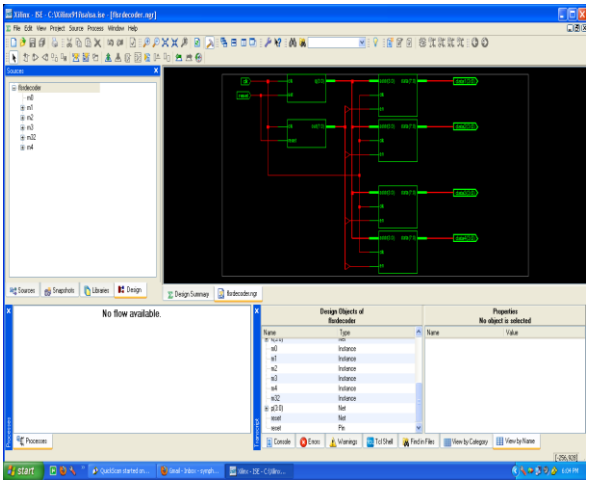

Fig no 4.2: LFSR module RTL systematic

However, an LFSR with a well-chosen feedback function can produce a sequence of bits which appears random and which has a very long cycle. A 16-bit Fibonacci LFSR. The feedback tap numbers in white correspond to a primitive polynomial in the table so the register cycles through the maximum number of 65535 states excluding the all-zeroes state. The state ACE1 hex shown will be followed by 5670 hex.

The bit positions that affect the next state are called the taps. The rightmost bit of the LFSR is called the output bit. The taps are XOR'd sequentially with the output bit and then fed back into the leftmost bit. The sequence of bits in the rightmost position is called the output stream.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

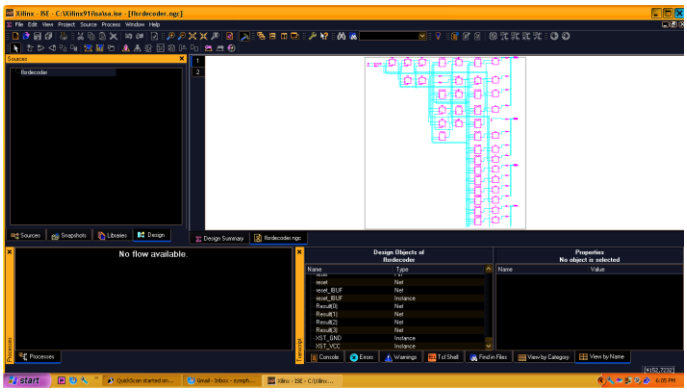

Fig no 4.2a: LFSR technology systematic

The above figure 4.2a shows linear feedback sift register technological systematic. The bits in the LFSR state which influence the input are called taps (white in the diagram).A maximum-length LFSR produces an m-sequence (i.e. it cycles through all possible $2^n - 1$ states within the shift register except the state where all bits are zero), unless it contains all zeros, in which case it will never change. As an alternative to the XOR based feedback in an LFSR, one can also use XNOR.

This function is not linear, but it results in an equivalent polynomial counter whose state of this counter is the complement of the state of an LFSR. A state with all ones is illegal when using an XNOR feedback, in the same way as a state with all zeroes is illegal when using XOR. This state is considered illegal because the counter would remain "locked-up" in this state. The sequence of numbers generated by an LFSR or its XNOR counterpart can be considered a binary numeral system just as valid as Gray code or the natural binary code.

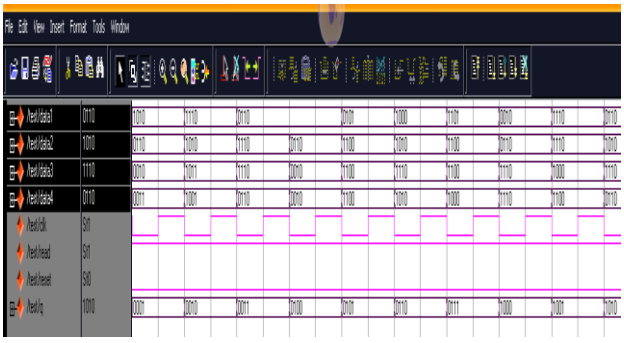

Figure no 4.2b: simulation result for LFSR

The above figure 4.2b: shows the simulation result for LFSR. It gives randomly accesses memory location. Here all memory locations are to be activated randomly hence it's required frequency is to be increased, so the time will be decreases, and hence it increases the speed of the memory locations.

5. CONCLUSION

This project explains the methods used in high speed parallel memory implementation. By using the memory decoder and LFSR technique increases the frequency of memory location, that means decreases the time. So it enhances the speed of the memory locations.

Hence heterogeneous reconfigurable hardware platforms offer the necessary flexibility for performing multiple wireless communication standards and achieve the performance required by the wireless standards. Furthermore, the combination of mixed-grained reconfigurable solutions enables energy efficient implementations of the wireless standards. Much work has been done on software defined radio (SDR) in the SDR forum context.

6. FUTURE SCOPE

The future scope of this project involves analyzing the effects of different depth of the memory hardware increase and the timing optimizations we get. This can be done by applying different levels decoders for the standard Montium-tile processor and concluding the results respect to hardware and timing issues. Also the design can be analyzed for different levels of unfolding factors to discuss the hardware overhead involving in different parallelism levels.

7. REFERENCES

1. Th. Claasen: "Is High-Speed the Only Solution to Exploit the Intrinsic Computational Power of Silicon?", keynote at the ISSCC 99, February, 1999.

2. P.M. Heysters, G.J.M. Smit & E. Molenkamp: "Montium – Balancing between Energy-Efficiency,

Flexibility and Performance", Proceedings of Engineering of Reconfigurable Systems and Algorithms (ERSA) 2003, pp 235-242, Las Vegas, Nevada, 2003.

3. P.M. Heysters & G.J.M. Smit: "Mapping of DSP Algorithms on the Montium Architecture", Proceedings of the 17th International Parallel & Distributed Processing
Symposium Reconfigurable Architectures Workshop (RAW) 2003, Nice, France, April 2003, ISBN 0-7695-1926-1.

4. G.J.M. Smit, P.J.M. Havinga, L.T. Smit, P.M. Heysters &M.A.J. Rosien, "Dynamic Reconfiguration in Mobile Systems", Proceedings FPL 2002, Montpellier France, pp
171-181, September 2002.

5. H. Veendrick: "Deep-Submicron CMOS ICs", 2nd edition, Kluwer academic publishers, 2000, ISBN 90-440-01116.

6. K. Yarlagadda, "ARM Refocuses DSP Effort", Microdesign resources, Microprocessor report, June 1999.