# A PROGRAMMABLE SIMD-BASED MULTI-STANDARD RAKE RECEIVER ARCHITECTURE

*Anders Nilsson, Eric Tell, Dake Liu*

Department of Electrical Engineering
Linkoping University
Linkoping, Sweden
email: {andni,erite,dake}@isy.liu.se

## ABSTRACT

Programmability with its associated flexibility will be increasingly important in future multi-standard radio systems. We are presenting a fully programmable and flexible DSP platform capable of efficiently performing channel estimation and MRC-based channel equalization for several CDMA based wireless transmission systems in software. Our processor is based on a DSP core with SIMD-computing clusters. We have mapped Rake receiver kernel-functions supporting several 3G standards to this micro-architecture and benchmarking shows that with the proposed instruction set architecture, our architecture can support channel estimation, equalization and decoding of: WCDMA FDD/TDD-modes and HSDPA at clock rate not exceeding 76 MHz during soft handover conditions.

**KEY WORDS**
SDR, DSP, WCDMA, RAKE, HSDPA

## 1. INTRODUCTION

The convergence of mobile devices requires flexible multi-standard baseband processing devices. This in turn requires programmable solutions since a fixed-function ASIC solution is not flexible enough. In this paper we present a processor architecture capable of running a Rake receiver in software. The architecture combines several SIMD computing *clusters* into a processor core. Unlike other flexible Rake architectures [1], this processor will run both the multi-path search, Rake fingers and Maximum Ratio Combining (MRC) in software. The created flexibility can also be used to trade-off channel compensation capability versus multi-code capability in software. For example, if channel conditions are good, computing resources can be re-allocated from channel search to multi-code de-spread yielding a higher throughput for the end user. We have chosen to map channel estimation and correction of the following WCDMA-standards to our architecture: WCDMA-FDD/TDD, and HSDPA. Our architecture is proven by mapping the most demanding algorithms to the architecture. We have chosen to implement the High Speed Data Packet Access (HSDPA) [2] mode since it contains heavy processing tasks and requires a short computing latency. By proving the HSDPA mode, we can ensure architectural support for the less demanding modes.

## 2. BACKGROUND

### 2.1 Overview of communication standards

All the investigated standards are based on 3.84 Mcps WCDMA operating either in Time Division Duplex (TDD) or Frequency Division Duplex (FDD) mode. An oversampling rate of 4 was used resulting in a sampling frequency of 15.36 MHz. By using an OSR of 4, the multi-path searcher will provide quarter-chip resolution without the use of fractional delay filters.

### 2.2 Rake based channel equalization

Rake receivers are used in CDMA systems for two reasons; the first reason is inter-symbol interference cancellation and the second reason is to utilize the multi-path diversity [3].

The idea of a Rake receiver is to identify a number of different multi-path components and align them constructively, both in time and phase, thus utilizing the created multi-path diversity. The function of delaying a certain transmission path and correct its phase is often referred to as a “Rake finger”. A Rake finger is illustrated in Figure 1.

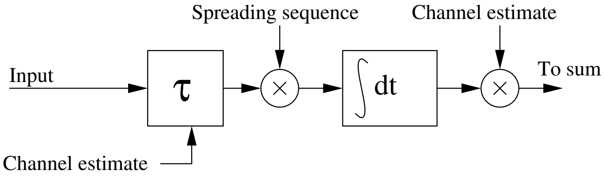


Figure 1: Rake finger.

The number of Rake fingers needed is determined by both the multi-path environment and the use of multi-codes. In a Rayleigh fading outdoor environment (ITU Pedestrian B) [4] up to 93% of the scattered energy could be utilized by four Rake fingers per code used.

## 3. PROCESSING CHALLENGES

There are several processing challenges in multi-standard WCDMA systems, namely channel estimation, the use of multi-codes and soft handover.

### 3.1 Multi-code transmission

To further improve the data-rate to a user in a WCDMA network, the user can be assigned several spreading codes in parallel. This will allow the user to receive and decode each of the codes in parallel, thus increasing the data-rate.

In most 3G standards de-spreading is accomplished by first de-scrambling the data by a scrambling sequence (Gold code), then by separating each data channel by multiplication with an Orthogonal Variable Spreading Factor (OVSF) code.

Since the de-spread operation in a classical Rake finger uses a compound code, one Rake finger is needed for each code used. This obstructs the use of multi-code transmission. If the number of Rake fingers is limited, the system must trade-off channel compensation capability and data-rate. Multi-code transmission is necessary to achieve the higher data-rates used in the WCDMA and HSDPA systems.

### 3.2 Soft handover

Since CDMA networks can be built as single-frequency networks, handover between base-stations can be done by *soft handover*. Soft handover can be explained as: when a user is on the border between two base-stations, the handover is performed gradually by transmitting data from both base-stations instead of switching base-stations instantly. This is possible due to multi-code transmission; a further development is *softer handover* where the mobile terminal is handed over between sectors in the same base-station. Then the same code-set is used so that the user cannot distinguish the new transmitter from a strong echo. The 3GPP WCDMA standards require mobile stations to handle up to 6 simultaneous base-stations during soft handover. To handle a soft handover scenario with 3 base-stations and three multi-codes, at least nine Rake fingers are necessary in the mobile terminal in order to de-spread the data.

## 4. ARCHITECTURE OVERVIEW

The presented processor architecture consists of a DSP processor with multiple SIMD-execution units. The data paths are grouped together into SIMD clusters. Each cluster has it's own load/store unit and vector controller. The clusters can execute different tasks while every data path within a cluster performs a single instruction on multiple data. The processor architecture is illustrated in Figure 2.

The processor core consists of five main blocks:

- A RISC-type controller.
- A 4-way complex short MAC (complex ALU).
- A 2-way CMAC.
- A memory sub-system with address generators.
- A memory crossbar switch.

In addition to the previously mentioned components, the core also consists of a number of accelerators used by all supported standards. The architecture contains the following accelerators:

- A digital front-end used for filtering and decimation.
- A configurable code generator for scrambling codes.

The SIMD units are the 4-way complex ALU, and the 2-way complex MAC. The main difference between this processor and a VLIW processor is the instruction issue process. By only allowing one instruction issue per clock cycle, excessive hardware could be saved.

### 4.1 Instruction set

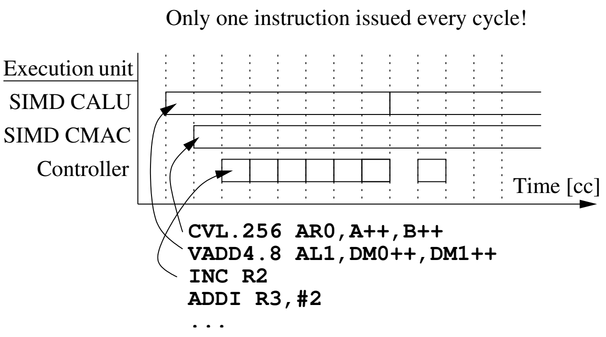
The instruction set architecture (ISA) of the processor consists of three classes of compound instructions.

1. **RISC instructions**, operating on 16 bit integers.
2. **DSP instructions**, operating on complex numbers.
3. **Vector instructions**, running vector operations on a particular SIMD-cluster.

The RISC-instruction class contains most control oriented instructions and this instruction class is executed on the controller unit of the processor. (Control ALU and MAC-unit). The DSP-instructions operate on complex-valued data and are executed on one of the SIMD-clusters. Vector instructions are extensions of the DSP instructions since they operate on large data-sets and utilize advanced addressing modes and vector loop support.

### 4.2 Instruction issue

Analysis and benchmarking of Rake receiver tasks shows that most operations are performed on long *vectors* of complex numbers. To utilize this property of baseband processing, the new processor architecture has been designed to perform vector operations efficiently. The processor has a special class of instructions, namely vector instructions that operate on a vector of data using memories as direct operand buffers. In this architecture vectors can be of any length between 2 and 128 elements long. By using vector instructions both processing efficiency and code-density can be significantly increased. This type of vector instructions was introduced in [5]. Instead of using a VLIW-scheme, this vector property of baseband processing is used in the design of the instruction set architecture. To reduce the complexity and improve the efficency of the control path, only one instruction can be issued every clock cycle. However since vector (SIMD) instructions run on long vectors, many RISC instructions can be executed during the vector operation. This is illustrated in Figure 3.



```
CVL.256 AR0,A++,B++
VADD4.8 AL1,DM0++,DM1++
INC R2
ADDI R3,#2
...
```

Figure 3: Instruction issue.

Unlike VLIW-machines, our architecture will allow consecutive vector operations and control flow instructions without the drawbacks and memory usage of a pure VLIW machine.

## 5. SIMD PROCESSING CLUSTERS

This processor architecture contains two SIMD execution clusters, accelerators and a controller unit. Common to the two SIMD computing clusters are the vector controller and vector load/store unit. (VLU/VSU). The VLU is the interface towards the memory blocks and the crossbar switch. The purpose of the VLU is to relax the memory access rate and reduce the number of memory data fetches.

The VLU can load data in two different ways. In the first mode, multiple data items are loaded from a bank of memories. In the other mode, data are loaded one data item at a

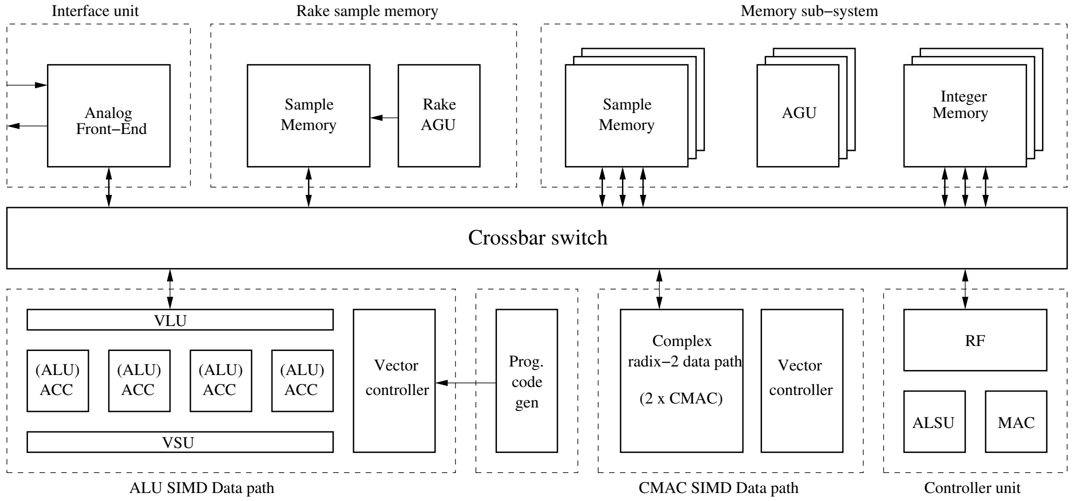


Figure 2: Processor micro-architecture

time and then distributed to the SIMD-data paths in the cluster. This later mode is used to reduce the number of memory accesses when consecutive data are processed by the SIMD cluster. If consecutive data are processed, the load unit can reduce the number of memory fetches by $3/4$ in a 4-way execution unit. The VSU unit is used to create local feedback between data-paths within the execution unit. This is used to post process data in the accumulator registers without having to move data to a memory for intermediate storage.

### 5.1 Vector ALU unit

The complex vector ALU along with address and code generators are the main components used for Rake finger processing. By implementing a 4-way complex ALU unit with accumulator, we can perform either four parallel correlations or de-spread of four different codes at the same time. These operations are enabled by adding a “short” complex multiplier capable of only multiply by $\{0, \pm 1; 0, \pm i\}$. The short complex multiplier can be controlled from either the instruction word, a de-scrambling code generator or from a OVSF code generator. All subunits are controlled from a vector controller which manages load and store order and hardware loop counting. A part of the micro-architecture of the Vector ALU is shown in Figure 4.

### 5.2 Vector CMAC unit

The control and load/store structures for the vector CMAC unit are identical to the control structures of the Vector ALU. The vector CMAC contains two full complex data paths which can be run separately or together as a Radix-2 FFT butterfly [5].

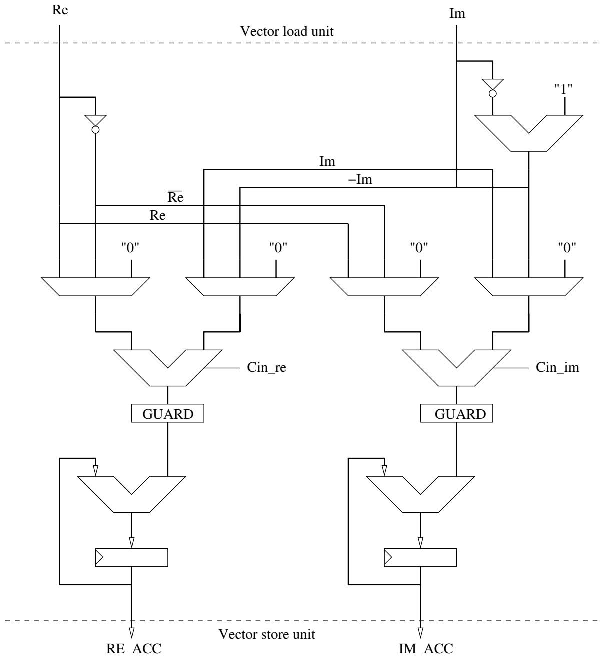


Figure 4: Part of vector ALU data path micro-architecture

## 6. MEMORY SUB-SYSTEM

Efficient memory management and allocation are essential in order to efficiently use SIMD architectures. To achieve a high enough memory bandwidth (4 complex samples per memory bank), the memory banks contain several small memory blocks operating in parallel. Each memory bank contains its associated address generator unit (AGU). These memories are in turn connected to a crossbar switch where they can be connected to several different computing engine ports. Since the memory access patterns are irregular, a special AGU is used to control the delay equalizer buffer used in the “Rake finger” processing.

To further optimize the data flow within the processor, several memory banks are used to effectively transfer data between computing engines [6]. Memory banks combined with the crossbar switch allow memory banks to be instantly (within 2 clock cycles) reconnected to another execution unit. This strategy provides an efficient and predictable data flow without memory move operations. The memory crossbar switch and memory resources are statically scheduled in order to reduce the complexity of the memory architecture while maintaining a predictable worst case timing.

## 7. FUNCTIONAL MAPPING

The four main kernel functions performed by a Rake receiver are:

1. Delay equalization.
2. De-scramble and de-spread.
3. Maximum Ratio Combining.
4. Multi-Path search and channel estimation.

By separating the de-scrambling operation from the despread operation, the same hardware can be re-used between operations in WCDMA systems. The mapping of the Rake finger functionality to hardware blocks is shown in Figure 5. First the 4-way complex ALU unit is used to de-scramble the received data. By later feeding each accumulator the corresponding OVSF-code the architecture can run de-spread on four simultaneous OVSF codes.

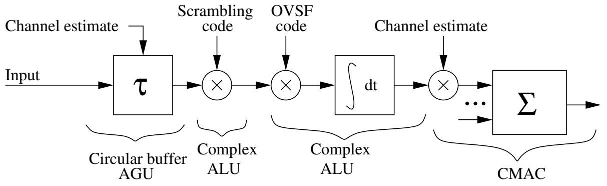


Figure 5: WCDMA Rake mapping.

Multi-path search is performed by using matched filters implemented as software running on the same SIMD units as the Rake finger processing. Correlation with the spreading/pilot sequence is performed by the Complex ALU, whereas peak detection and absolute maximum search is performed by the CMAC unit.

To achieve optimal resource usage, the number of computing elements is balanced between SIMD clusters so that each subtask consumes approximately the same number of clock cycles. Also, when an execution unit is not used, it's inputs are masked to reduce unnecessary activity on internal logic.

## 8. SOFTWARE DEVELOPMENT

Effective software development tools are essential to speed up implementation of various algorithms on the architecture. The software tool-set for the processor architecture consists of:

- A C-compiler.
- A cycle-true simulator for de-bugging and program development.

These tools are currently under development. For this research other system simulation tools were developed and used to select algorithms, evaluate memory access costs and to create a coarse scheduling of the processor. Here bit-true and cycle-true C++-libraries in conjunction with a standard C++ compiler were used. To fully understand all aspects of Rake receivers in practice, a test system was built and real WCDMA data were recorded in the baseband domain. The data were used as input stimuli to the system models and the simulator to further verify the accuracy of the used models.

## 9. RESULTS

In this paper the main focus has been on a flexible and efficient SIMD based processor architecture capable of performing Rake-based channel equalization for common and future WCDMA based communication standards in software. Care has been taken to ensure architectural support for soft and softer handover and multi-code transmission. We have presented a system architecture and a memory sub-system and we have mapped Rake kernel functions to this architecture.

Furthermore, the processor architecture is designed with low power techniques in mind. The control paths are minimized due to the inherit control locality of the CSIMD architecture. Furthermore, single port memories are used to reduce the power consumption due to memory accesses.

Benchmark results show that the architecture is capable of performing all kernel functions associated with a Rake receiver for the standards discussed at a clock rate not exceeding 76 MHz de-spreading three multi-codes while performing soft handover to six base stations. This limit is given by the memory access rate for WCDMA. Furthermore, assembly simulations of a WCDMA Rake receiver show that the cycle cost of control code execution is masked by vector operations running in parallel.

## 10. FUTURE WORK

In order to fully take advantage of the single issue concept, automatic scheduling tools and algorithms for dynamic scheduling must be further investigated. Research efforts must also be spent on development of a C-compiler. Also, VLSI implementation of the processor is necessary to measure power and silicon area.

## 11. CONCLUSION

Programmability is essential for multi-standard baseband processors. In order to be able to efficiently support multiple standards within a processing device, new architectures are necessary. As a response to this, we have presented a SIMD based architecture targeted at WCDMA systems. We have also presented a mapping of Rake kernel functions to our programmable baseband processor. Our architecture is versatile and area efficient since we ensure maximum utilization of each execution unit.

## REFERENCES

[1] L. Harju, M. Kuulusa, J. Nurmi; Flexible implementation of WCDMA Rake receiver; *Signal Processing Systems, 2002. (SIPS '02).* IEEE Workshop on , 16-18 Oct. 2002 Pages:177 - 182

[2] 3rd Generation Partnership Project; Physical channels and mapping of transport channels onto physical channels (FDD) 3GPP TS 25.211 V6.0.0

[3] H. Holma and A. Toskala; *WCDMA for UMTS Radio Access For Third Generation Mobile Communications*, John Wiley and Sons, Inc., 2001.

[4] ITU ITU-R M.1225, Guidelines for evaluations of radio transmission technologies for IMT-2000, 1997.

[5] Eric Tell, Anders Nilsson and Dake Liu; "A Low Area and Low Power Programmable Baseband Processor Architecture" in *Proceedings of IWSOC*, Banff, Canada, July 2005

[6] A. Nilsson, E. Tell; *An accelerator structure for programmable multi-standard baseband processors.* Proc. of WNET2004, Banff, AB, Canada, July 2004.