Simultaneous multi-standard support in programmable baseband processors

Anders Nilsson[1], Eric Tell[2] and Dake Liu[1]
[1]Div. of Computer Engineering at Dept. of E.E.,
Linköping University, Linköping, Sweden
E-mail: {andni,dake}@isy.liu.se

[2]Coresonic AB, Linköping, Sweden
E-mail: eric.tell@coresonic.com

***Abstract*— Programmability will be increasingly important in future multi-standard radio systems. In this paper we present an enhanced baseband processor architecture capable of efficiently supporting simultaneous multi-standard operation. Our DSP processor is based on the SIMT (Single Instruction stream Multiple Tasks) architecture which allows concurrent tasks to be executed on the processor controlled by only a single instruction stream. We also discuss enhancements of the base processor architecture to increase the efficiency of simultaneous multi-standard execution. By profiling and mapping GSM and WLAN (IEEE 802.11g) to the architecture we show that simultaneous support for the above mentioned standards can be accomplished with 245 MHz clock frequency and 1359 words of complex data memory on the given architecture.**

I. Introduction

Programmable baseband processors are necessary to enable flexible multi-standard radio systems. Programmability can also be used to quickly adapt to new and updated standards since a pure ASIC solution will not be flexible enough. ASIC solutions for multi-standard baseband processors are also less area efficient than their programmable counterparts since processing resources cannot be efficiently shared between different operations. Traditionally the research focus has been on enabling multi-standard baseband processors and the issue of efficiently being able to execute several standards simultaneously has not been covered. By using a single programmable baseband processor to execute several baseband processing programs at the same time we can:

- Improve hardware reuse. (save silicon area)
- Share software kernel functions. (save program memory)
- Utilize shared information such as link state and channel parameters. (improve performance)

However, as baseband processing is a hard real time problem (with latency requirements on 1 μs scale) and is computationally heavy, special attention must be paid to achieve efficient multi-standard support with low scheduling overhead. One goal of this work is to identify what could be done in hardware to support efficient execution of several radio standards at once.

In this paper we first present a base architecture which is suitable for running the presented standards separately. The processor is based on a DSP core equipped with several SIMD execution units (clusters) which can operate in parallel controlled by a single instruction flow. The standards are later profiled on this architecture. Then scheduling principles and architectural enhancements are discussed. Finally mapping and resource usage are discussed. In this paper we only consider reception tasks in GSM and Wireless LAN, as reception is generally more demanding than transmission.

The paper is organized as follows: In Section II some of the unique properties of baseband processing are discussed. The base processor architecture is also presented in this section. Section III describes how to divide baseband processing algorithms into tasks. In section IV GSM and WLAN algorithms are profiled and mapped to the architecture. Scheduling is described in Section V and results are presented in section VII. Finally, conclusions are drawn in section VIII.

II. Background

In this section some of the unique properties of baseband processing are introduced. The base processor architecture is also described as well as some of the issues related to simultaneous multi-standard reception.

A. Baseband processing characteristics

Baseband processing differs from general computing in many ways. First baseband processing a is *hard real time* problem, secondly baseband processing is very computationally demanding. Latency requirements are on 1 μs scale. To manage these two points in a programmable processor, special computer architectures are necessary. Profiling (see Section IV) shows that a Wireless LAN receiver still requires hundreds of MIPS on a specialized processor to manage reception in software. However, analysis of baseband processing [1] reveals that the baseband processing jobs can be divided into task chains with no or little backward dependency between the tasks. This allows us to create software pipelines of tasks and increase processing parallelism.

B. SIMT base architecture

The processor architecture used in this paper is named Single Instruction stream, Multiple Task (SIMT). Unlike other parallel architectures [2] which issue a number of simple

operations in parallel, the SIMT architecture can issue larger (*vector*) operations such as a 128 sample complex dot-product with a single instruction. This allows the SIMT architecture to perform several large tasks in parallel using only a narrow instruction flow. This principle is illustrated in Figure 1. Vector

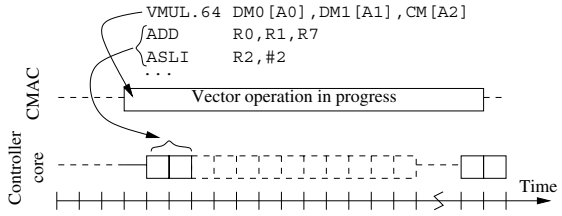

Fig. 1. The concept of single issue vector operations.

instructions are enabled by the underlying architecture which consists of:

1) Small separate memory blocks with private address generators
2) Execution units (Complex MACs and Complex ALUs)
3) An on-chip network
4) Accelerators

Since each execution unit is connected to its own memory blocks, concurrent operations can be supported in all execution units. By using a on-chip network, the number of memory accesses can be reduced since memories can be "handed over" to other execution units by a simple network reconnect. [4] The base architecture is presented in Figure 2.

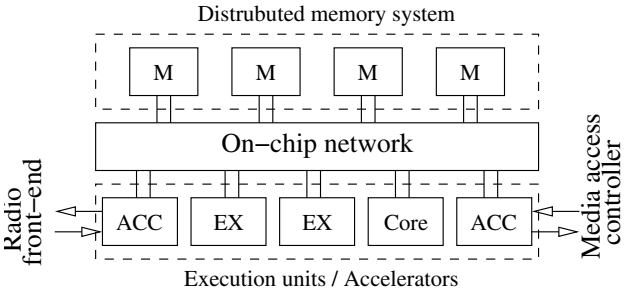

Fig. 2. Base architecture.

III. Task analysis

To aid profiling of GSM and WLAN reception baseband programs and to further facilitate scheduling in the processor, we introduce the *task* concept. Besides being used in scheduling, division of a program into a chain of tasks helps identifying data dependencies and maximum tolerable latency of different operations during the profiling stage. In this paper we define a task to be an atomic operation that is executed on one execution unit. Hence a program considered to be a sequence of atomic tasks.

A. Task classification

To assist profiling and scheduling, we define the concept of task-groups, e.g. a set of tasks with a common dead-line.

Some task-groups typically associated with the radio configuration will have very short (2-5 μs) dead-lines. Common to those tasks, they need to be performed before any data can be stored in memory. Automatic Gain Control (AGC) and similar tasks often fall in this group. These tasks often start a chain of further processing tasks. The processor must have enough resources to manage all possible such tasks within the maximum latency time for each of them. A task-group is further divided into atomic tasks.

B. Task latency

Since baseband processing is a hard real-time problem, hard deadlines exist where data must be completely processed. These limiting factors are imposed by the radio front-end and higher data layers:

- **AGC/AFC:** The processor must perform Automatic Gain Control (AGC) and Automatic Frequency Control (AFC) tasks before the data is stored in memory for further processing.
- **Higher protocol layers:** In most packet systems such as WLAN, the standard stipulates the maximum allowed time from the end of a received packet to the beginning of the transmitted response packet.

IV. Profiling

Profiling of WLAN and GSM programs visualizes resource usage and highlights task dependencies. During profiling, the programs are divided into chains of tasks that execute on the initially assumed hardware. Profiling results are then used to enhance and dimension the initially used hardware architecture accordingly. In Table I basic properties of the selected standards are discussed.

TABLE I
STANDARD OVERVIEW

Parameter	GSM	IEEE 802.11g (OFDM)
Symbol period:	$3.69\mu s$	$4\mu s$
Symbols/sub-carriers per frame/slot:	156.25	64
OSR:	4	2
Sample Rate:	1.083 Msps	40 Msps
Maximum latency: for packet/frame reception	$576\mu s$	$10\mu s$

A. Initial hardware assumption

To support the task analysis, the following hardware resources have been assumed:

1) Four memory banks with 1024 words of complex data each.
2) A 4-way Complex MAC unit capable of performing a radix-4 butterfly each cycle.
3) A controller core with a real-valued ALU and multiplier as well as private memory for software stacks etc.
4) Accelerators for cycle intensive bit-manipulation tasks such as Viterbi decoder, Scrambling, Interleaving et.c.

The assumptions are based on the processor presented in [4]. The processor core is also assumed to sustain one instruction

per clock cycle. Due to the SIMT architecture this is true for most operations executing on the architecture.

B. Task analysis

Algorithm selection and analysis of WLAN and GSM reception yield the task division presented in Table II. The cycle count cited is the number of cycles the corresponding kernel operation will consume on the hardware.

TABLE II
TASK DIVISION AND CYCLE USAGE

#	Task	GSM normal burst	WLAN Preamble	WLAN Data
1	**AGC*:**	32 cc	48 cc	-
2	**CFO estimation*:**	-[a]	88 cc	-
3	**Synch.:**	62 cc	220 cc	-
	FFT:	-	70 cc	-
	Multiplication:	-	16 cc	-
	IFFT:	-	70 cc	-
	Max Search:	-	64 cc	-
4	**Channel est.:**	36 cc	16 cc	-
5	**Viterbi ch. dec.:**	25984 cc	-	-
	116 x Mod.:	224 cc	-	-
6	**FFT:**	-	-	70 cc
7	**Ch. tracking:**	-	-	120 cc
8	**Ch. comp:**	-	-	16 cc
9	**Demap:**	-	-	64 cc (Acc)
10	**Interleaving:**	18 cc (Acc)	-	29 cc (Acc)
11	**Conv. decoder:**	464 cc (Acc)	-	228 cc (Acc)
12	**Scrambling:**	116 cc (Acc)	-	228 cc (Acc)

[a] GSM use a special frequency correction burst.

Tasks marked with * in Table II are classified as immediate tasks since they need to be performed before any data can be stored in memory. Dependency analysis give the following processing latency requirements presented in Table III. The table also indicates the peak memory usage by a task-group. (Local memory in accelerators is however not included.)

TABLE III
TASK LATENCY REQUIREMENTS AND MEMORY USAGE

Task group	Maximum latency	Maximum memory usage
GSM:		
1:	3.6 μs	32 words
3-5,10-12:	570 μs	671 words
WLAN:		
1-2:	6.4 μs	120 words
3-4:	4 μs	144 words
6-8:	1.2 μs	208 words
9-12:	2.8 μs	64 words
Coefficients:		480 words

V. SCHEDULING

Scheduling can be performed in many ways, most traditional scheduling principles are discussed in [5]. However, in this paper we have used a lightweight scheduler due to the low scheduling complexity and extreme performance requirements. The scheduler described below is only intended to perform basic scheduling to illustrate hardware performance.

As the baseband processing tasks consume about 40-100 cycles, the scheduling and and task switch operations must be performed with a minimal cycle count not to dominate the processor load.

To efficiently support scheduling, special context switch instructions are inserted between tasks in the software at compile-time. This task partition could be performed automatically or by the programmer. Upon execution of such an instruction the processor performs a context switch to the supervisory code which then performs another context switch to issue the next task.

A. Scheduling principle

The scheduling principle used in this investigation is based on knowledge of the period in which GSM and WLAN tasks are initiated. From the GSM standard we know that we might receive a data burst every 576 μs. WLAN bursts can however be received more often. WLAN activity can in worst case be back-to-back on the radio channel. With this in mind and the latencies from Table III, we conclude that all WLAN tasks will have precedence over GSM tasks (Shortest-period-first scheduling). This will require the processor to being able to serve the largest GSM task within the respective latency time.

The processor will execute the scheduler between each task in the GSM task flow. If a WLAN packet is detected, the task scheduler will start to execute the WLAN flow. This is illustrated in Figure 3.

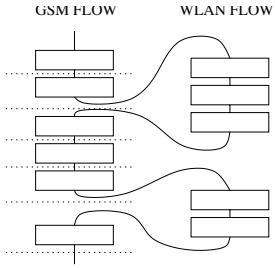

Fig. 3. Scheduling example

VI. ARCHITECTURE ENHANCEMENTS

Taking the step from execution of a single standard at a time to simultaneous execution of several standards requires additional features of the architecture to maintain a high computing efficiency. By profiling we have found the following architecture enhancements necessary:

1) **Additional memory banks.** In order to sustain concurrent tasks in accelerators and vector units, the number of memory blocks must be increased from the initially assumed number of blocks. Profiling yields that another four memory banks are needed to support worst case load. The memory blocks will be used as ping-pong buffers towards accelerators.
2) **DFE buffer** Automatic buffering of the first received samples will relax the requirements of the AGC task as buffering will start as soon as a packet is detected.

3) **Fast context switches.** Since baseband processing tasks are fairly short, the number of cycles spent on context switches must be kept to a minimum. Fast context switches require the ability to swap in/out parts of the register file for each task. Each context also needs its own stack-pointer as well as a private program counter.
4) **Stack support.** If additional registers are necessary to be saved between context switches, the values must be stored in memory.

VII. Results

From Table II and Table III we can conclude that the maximum memory usage case will be when a GSM processing burst is interrupted by a WLAN packet. This will require a total of 1359 words of complex memory (671 words from GSM, 208 words from WLAN and 480 constant words. Stacks etc. in the controller not included).

A. Resource usage

In Table IV the resource usage for different task groups is presented.

TABLE IV
Overview of required MIPS and memory

Task group	Required MIPS	Required Memory
GSM:		
1:	9 MIPS	32 words
3-5,10-12:	47 MIPS	671 words
WLAN:		
1-2:	22 MIPS	120 words
3-4:	59 MIPS	144 words
6-8:	171 MIPS	208 words
9-12:	196 MIPS	64 words
Constants:		480 words

In Figure 4 the required computing capacity v.s. time is illustrated.

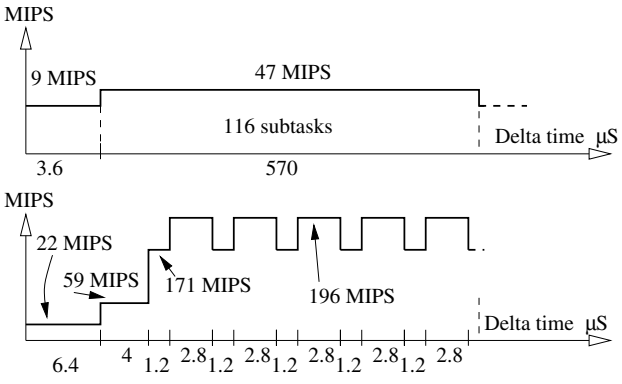

Fig. 4. Required computing capacity v.s. time.

B. Peak load

The peak load of the processor will occur if a long WLAN packet is received just after the start of the synchronization task in GSM. Then the processor needs to simultaneously support 47 MIPS (GSM) along 196 MIPS (WLAN) during peak conditions as illustrated in Figure 4. The worst case load originates from the overall computing requirements over a GSM slot with full WLAN traffic. The case where WLAN tasks 1-2 (See Table II and III) occur precisely after a subtask in GSM task 5 is issued will only require 56.25 MIPS. (224 + 48 + 88 cycles on 6.4 μs).

To reduce the power consumption, clock gating support is essential. As shown in Figure 4, not all computing capacity is needed all time. Turning off the clock of individual execution units, accelerators and memories can thus save power.

C. Scheduling overhead

We have deliberately chosen a lightweight scheduling scheme in this paper to keep scheduling overhead low, at the cost of a slightly over-designed hardware in order to maintain a guaranteed performance. By enhancing the core with single cycle context switch instructions and mechanisms for resource allocation, scheduling overhead can me kept to a minimum. Under full load in both GSM and WLAN there will be maximally 126 task switches per GSM slot (which is the largest scheduling period).

According to the peak load calculated in Section VII-B, the peak load of the processor is 243 MIPS. Each task switch will consume maximally 12 cycles (context switching, network setup, et.c). This will add another load of maximally 1512 cycles per 576 μs which corresponds to 2.6 additional MIPS.

VIII. Conclusion

It is possible to manage both GSM and WLAN reception on the presented architecture at only 245 MHz using 1359 words of complex data memory. The scheduler and task switch mechanism only adds 1.1% cycle overhead. This extra overhead is compensated by the hardware reuse. This makes enhanced SIMT based baseband processors well suited for simultaneous multi-standard processing in mobile terminals.

References

[1] A. Nilsson, *Design of multi-standard baseband processors* Linköping Studies in Science and Technology, Thesis No. 1173, Linköping, Sweden, June 2005

[2] J. Glossner et al, *A Software-Defined Communications Baseband Chip*, IEEE Communications Magazine, January 2003.

[3] G. Fettweis, *Embedded SIMD Vector processor design*, SAMOS Workshop, July 2003

[4] E. Tell, A. Nilsson, and D. Liu *A Low Area and Low Power Programmable Baseband Processor Architecture*, Proc of the International workshop on SoC for real-time applications, Banff, Canada, July 2005

[5] Bacon and Harris: *Operating Systems - Concurrent and Distributed Software Design*, Addison-Wesley, 2003

[6] GSM standard; ETSI TS 145 001 V5.8.0 (2004-11) Digital cellular telecommunications system (Phase 2+); Physical layer; General description

[7] IEEE 802.11g, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band, 1999.