Oolong: A Baseband processor extension to the RISC-V ISA

Cecil Accetti R. A. Melo, Edna Barros
CIn - Informatics Center
Federal University of Pernambuco - UFPE
Recife, Brazil
caram@cin.ufpe.br

***Abstract*— RISC-V is an open-source instruction set-architecture, designed to support customized extensions and architectures. This paper presents an instruction-set extension to the RISC-V ISA, idealized for software-defined radio applications. The custom instructions perform complex-number arithmetic, tailored for complex or quadrature modulation and baseband processing, and can perform one complex multiply-accumulate per cycle. The proposed system architecture includes the processor core, a WISHBONE bus interconnection, IO and peripherals, and was targeted to an Altera Cyclone III FPGA, achieving 0.9 DMIPS/MHz without the use of any compiler optimizations.**

Keywords—SDR, ASIP, DSP, RISC-V, Software-defined Radio, Baseband processor

I. INTRODUCTION

In the past two decades, software-defined radio (SDR) has evolved from novelty to a reality. What was once done purely in the analog domain, has now become sophisticated multistandard mixed-signal systems [1]. SDR implementations can be defined in two trends: using high-performance general-purpose processors (GPP) for the base-station, application-specific instruction-set processors (ASIP), for the embedded or mobile realm. Digital signal processors (DSP) are included in the latter category, and have been used in cellphones together with ARM processors since the beginning of GSM and other digital standards [2]. Outside the SDR world, digital baseband processing is usually performed with application-specific integrated circuits (ASIC), customized to a specific protocol. ASICs can be power-efficient solutions but development costs are high, and due to the lack of programmability, they do not support evolving protocols and standards.

Partial customization of standard instruction sets (ISA) not only lowers the effort to design the necessary software tools, but allows the use of a known, well tested architecture as a starting point to the new ASIP [3]. In this paper, we propose a baseband processing extension to the RISC-V ISA [4], customized for small packet modem applications, and present the processor core that implements the proposed extension, targeted to a Cyclone III FPGA

II. ARCHITECTURE

Figure 1 shows the block diagram of the processor and its supporting platform. The processor core consists of a 32 bit RISC-V processor [9], with integer base instructions (I), the standard multiplication extension (M) (R-V 32IM) and the Oolong baseband processing accelerator unit.

This work was sponsored by the Brazilian National Research Council, CNPq.

A. Main Processor Core

The main processor core is a Harvard architecture 3-stage in-order pipeline, which implements the RISC-V integer base instructions (RV32I) and the integer multiplication and division standard extension (RV32M). The processor core supports static branch prediction, with 2-cycle penalty for prediction misses.

To compare the performance of our RISC-V implementation we've compared the results of the Dhrystone 2.1 benchmark with the published results for 3-stage pipeline ARM cores. The Cortex-M1 core [5] was targeted to an Altera Cyclone III FPGA, the same device family of our test platform, getting a slightly lower performance (0.8 DMIPS/MHz vs. 0.9 DMIPS/MHz).

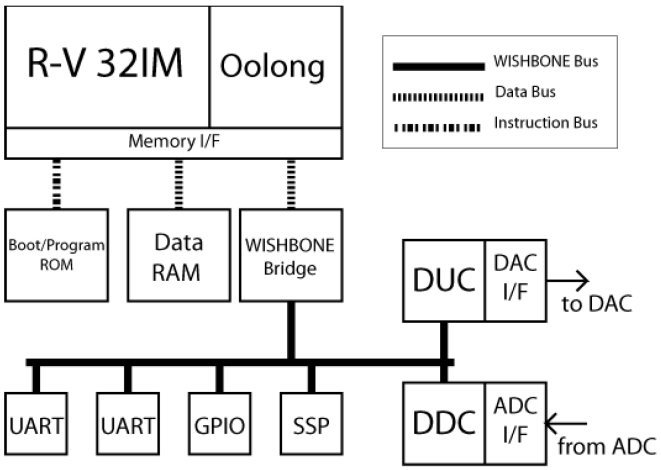

Fig. 1. Block diagram of the processor core and its peripherals.

B. Ooolong baseband accelerator

Oolong is a hardware accelerator designed for execution of baseband processing (complex arithmetic) and packed 4 x 8bit (SIMD) fixed-point instructions.

Most of baseband processing tasks, especially during modulation and demodulation deal with complex numbers or in-phase (I) and quadrature (Q) samples. To represent complex numbers, with real and imaginary parts, without changing the addressing modes and data width, and to preserve the use of the standard register bank of the RISC-V architecture, 32-bit complex-data words are considered vectors of two 16-bit words. The I part is stored in bits 31 through 16, and the Q part is stored in the lower 16 bits

The instructions listed on Table I can all be implemented using standard RISC-V instructions. However, the number of cycles needed to perform those operations with the standard ISA

would affect the performance of the core in baseband processing applications. The challenge is to guarantee real-time operation to all tasks on low clock frequencies, as a soft-core processor in an FPGA. Low clock frequency also reduces power demands, which are critical to small embedded systems. As an example, the simplest instruction, CADD executes two 16-bit additions on a single clock cycle. The CMAC instruction executes seven 16 bit additions and three 16-bit multiplications on a single cycle, and the VMAC instruction performs eight 8-bit additions and multiplications in a single cycle. To perform a complex multiply-accumulate operation using standard RISC-V instructions, 10 clock cycles would be needed. Data parallelism is explored with a four-lane 8-bit vector ALU, that also executes the Galois Field polynomial multiply instructions, PXMUL and VPXMUL. These instructions were added to the Oolong extension, to accelerate the execution of the Advanced Encryption Standard (AES) cryptography algorithm. Since there are no brute-force attacks known that could decrypt AES-encrypted data, adding an accelerating instruction to Oolong would contribute to the security of the data processed by our software-defined radio baseband processor.

TABLE I. OOLONG INSTRUCTIONS

Instruction	Description
CADD	Signed Complex Add, 16+16bit
CSUB	Signed Complex Subtract, 16+16bit
CCONJ	Complex Conjugate, 16+16 bit
CMUL	Complex Multiply, 16+16 bit
CMAC	Complex multiply-accumulate 16+16 bit
CRADIX2	Complex radix-2 butterfly, 16+16bit
CSMUL	Complex-scalar multiply,16+16bit
CABS	Complex absolute value, 16+16bit
PXMUL	Galois Field (GF) polynomial multiply-by-x (8bit data)
VPXMUL	Packed GF multiply-by-x (4 bytes)
VMUL	Packed Multiply (4 bytes)
VMAC	Packed Multiply-accumulate (4 bytes)
VADD	Packed Add (4 bytes)

Instead of designing higher-level instructions that implement parts or rounds of the AES algorithm, as is done in x86 and ARMv8 instruction sets, we opted to implement the most critical paths of AES, especially the GF polynomial multiplication. The same approach was followed to the design of the remaining instructions, maintaining the RISC paradigm of simple, low complexity instructions. The PXMUL instruction, Galois field (GF-2^8) multiplication by the polynomial x, is a basic operation that is used in GF-2^8 polynomial multiplication, Multiply() in AES terms. Using the standard RISC-V instructions, the Multiply() function takes 74 instructions, and is called 64 times for each round of the AES algorithm. For AES-128 a total of 10 rounds are needed. With the PXMUL instruction, the Multiply() function takes 26 instructions, thus allowing a 65% code reduction, in one of the most time consuming tasks of AES.

III. SYNTHESIS AND TESTS RESULTS

The processor RTL and all of its peripherals were described in 3098 lines of code of the VHDL hardware description language, following design guides from both the RISC-V 2.0 [9] and the WISHBONE B4 [10] specifications.

Synthesis was targeted to the Altera Cyclone III FPGA device family, obtaining a maximum clock frequency of 75MHz at a speed class 6 device. Typical operating clock frequency for a speed class 8 is 50MHz. Evaluating applications were AM/AM-DSB, BPSK, 4QAM, 16QAM modems, using real-time scheduling.

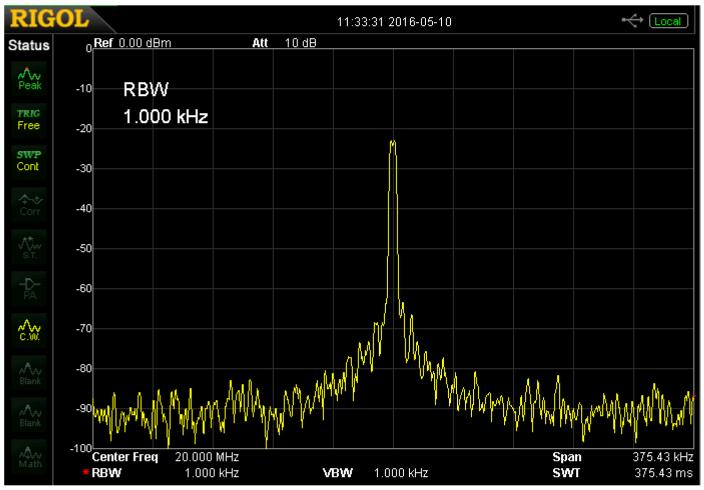

Fig. 2. Measured spectrum of a AM-DSB signal generated with the proposed platform.

REFERENCES

[1] Nilsson, Anders; Tell, Eric; Wiklund, Daniel; Liu, Dake. *Design methodology for memory-efficient multi-standard baseband processors*. Asia-Pacific Conference on Communications, Oct. 2005.

[2] Texas Instruments Calypso chipset.
http://www.ti.com/corp/docs/webemail/ctiamar03/GSM_GPRS_021903.pdf

[3] Kong, Ji; Peilin, Liu; et al.; *Next-Generation Consumer Audio Application Specific Embedded Processor*. 2010 IEEE 8th Symposium on Application Specific Processors (SASP)

[4] Andrew Waterman, Yunsup Lee, David Patterson, Krste Asanovic; *The RISC-V Instruction Set Manual Volume I: User-Level ISA Version 2.0*. http://riscv.org/specifications/

[5]"Cortex-M1 processor" http://www.arm.com/products/processors/cortex-m/cortex-m1.php