Review Paper on Implementation of DSP Unit in LEON3 Processor

Pushkar P. Joshi
Student of M.Tech, Electronics Engg.
KITS, Ramtek
Maharashtra, India
Email:pushkar_college@rediffmail.com

Prof. A.V. Mahatme
Dept. of Electronics & Communication Engg.
KITS, Ramtek
Maharashrta, India
Email:mahatme.archana@gmail.com

Abstract—LEON3 is a SPARC V8 processor developed by Gaisler Research. It is a synthesizable processor core for embedded applications.The LEON3 processor is part of the *GRLIB* package that is under continuous growth.To demonstrate the abilities of the processor it is a curiosity to introduce a DSP unit optimized for FIR filter in LEON3processor with additional cores from the GRLIB IP (Intellectual Property) library developed by Gaisler Research. The model is capable of performing filtering operations like low pass, high pass, band pass and band stop based on selection that is embedded into the design.The complete work is realized as a System on Chip (SOC) design. LEON3 has been programmed entirely in VHDL by Jiri Gaisler of the European Space Agency.

Keywords*- DSP Unit, LEON3 Processor, SOC Design.*

*****

I. Introduction

The present day form of embedded systems is an innovation that has taken more than a hundred years. The effect of the development of embedded systems is same as that of the discovery of fire or the invention of the wheel had on the growth of mankind. A digital system is a part of an embedded system embedded in a larger system and it is an application specific. Both in computing power and scope of embedded systems have experienced an enormous growth of their possible applications. Embedded systems are the collection of portable devices such asMP3 players and digital watches to large stationary installations like factory controllers or the systems controlling nuclear power plants, traffic lights. The advanced software IP cores and embedded operating system are required to create a working environment for the growth of' embedded system but they are license bounded and it also requires development kit.

GRLIB is a library of VHDL source codes of IP cores for designing a complete system on chip centered on the LEON3 processor. The library is structured into directories that group IP cores according to the contributor’s company. There are more subdirectories with complete IP cores organized in packages in the directories. These directories contain VHDL source codes of packages for simulation and synthesis and files with package configurations. A package configuration is propagated to VHDL entities via generics. The library is managed by an automated tool based on *GNU make* that manages configurations and compilation of packages for simulation and synthesis [1].

The propose research work is to present Filters designed with LEON3 processor which is superior over their analog counterparts for better specifications, stability, performance and reproducibility. This processor is tuned only for a particular application that can be used for low-power implementation and also word lengths can be adjusted to the current problem. By using this specific processor with LEON3 will improve the use of LEON3 processor. The platform is based on the AMBA SOC bus protocol and combines an innovative interfacing scheme which utilizes the bus hierarchy within AMBA in order to allow single and multiple high performance DSP Intellectual Property cores to be integrated to the SOC platform utilizing the LEON3 Processor.

II. Comparative study of LEON3, Micro Blaze and Open RISC 1200

Processor selection is one of the key design decisions in any SOC development [2]. Selection of processor depends on some technical aspects like speed, cost, size and some non-technical aspects such as development environment license etc [3]. Some of the well-known commercial cores are Altera's NIOS II. Xilinx's Micro Blaze and Pico Blaze, Tensilica's Xtensa. There are numerous soft cores freely available from open source communities across the internet like LEON3 by Gaisler research. These are called Open source cores. From [4] the comparative study of LEON3, Micro Blaze and Open RISC 1200 are surveyed. Micro Blaze performs nearly as well as LEON3 for the fastest configuration, where LEON3 operates at significantly lower clock frequency. Micro Blaze is significantly more efficient per area unit than the other two processors but it is highly optimized for Xilinx FPGAs. The LEON3 shows better efficiency per area unit than the Open RISC 1200 processor. For the area optimized configuration LEON3 and Open RISC l200 utilizes approximately the same area. LEON3 is less difficult than the others to manage. Micro Blaze has the best documentation and the best support for adding user defined IP-blocks. The Open RISC 1200 documentation is insufficient and the processor is in general more difficult to manage than both the Micro Blaze and LEON3 processors.

III. Architecture of LEON3 Processor

LEON3 is a synthesizable VHDL model of a 32-bit processor compliant with the SPARC V8 architecture. The model is highly configurable and particularly suitable for system-on-a-chip (SOC) designs. The full source code is available under the GNU LGPL license allowing free and unlimited use in both research and commercial applications [5],[6].

Features of LEON3 are-

•SPARC V8 integer unit with 7-stage pipeline
• Hardware multiply, divide and MAC units
• Separate instruction and data caches
• Support for 2 - 32 register windows
• Radix-2 divider (non-restoring)
•Single-vector trapping for reduced code size
• Advanced debug support unit
• Optional IEEE-STD-754 compliant FPU
• 20 DMIPS at 25 MHz system clock
• Fault-tolerant version available

Following figure shows the block diagram of LEON3 processor.

1) *Integer unit:* The LEON3 integer unit implements the full SPARC V8 standard, including all multiply and divide instructions. The number of register windows is configurable within the limit of the SPARC standard (2 - 32), with a default setting of 8.
2) *Floating-point unit and co-processor:* The LEON3 model does not include an FPU but provides a direct interface to the Meiko FPU core and a general interface to connect other floating-point units. A generic co-processor interface is provided to allow interfacing of custom co-processors.
3) *Cache Sub-System:* Separate instruction and data caches are provided each configurable to 1 - 64 Kbytes. With 8 - 32 bytes per line. Sub-blocking is implemented with one valid bit per 32-bit word.
4) *Memory Interface:* The memory interface provides a direct interface PROM, SRAM and memory mapped I/O devices. The memory areas can be programmed to either 8-, l6- or 32-bit data width.

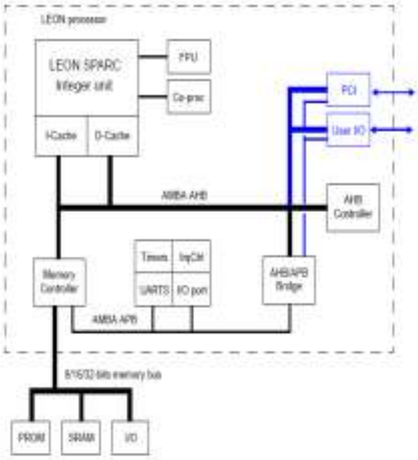

Fig.1- Block Diagram of LEON3 Processor

5) *Timers:* Two 24-bit timers are provided on-chip. The timers can work in periodic one-shot mode. Both timers are clocked by a common l0-bit pre-scalar.
6) *UARTs:* Two 8-bit UARTs are provided on-chip. The baud-rate is individually programmable and data is sent in 8-bits frames with one stop bit. Optionally one parity bit can be generated and checked.
7) *Interrupt Controller:* The interrupt controller manages a total of 15 interrupts originating from internal and external sources. Each interrupt can be programmed to one of two levels. A changed secondary controller for up to 32 extra interrupts is also available.
8) *Parallel I/O Port:*An l6-bit parallel I/O port is provided. Each bit can be programmed to be an input or an output. Some of the bits have alternate usage such as UART inputs / outputs and external interrupts inputs.
9) *AMBA On-Chip Buses:* The processor has a full implementation of AMBA AHB and APB on-chip buses. A flexible configuration scheme makes it simple to add new IP cores. Also all provided peripheral unit's implement the AMBA AHB/APB interface making it easy to add more of them or reuse them on other processors using AMBA.
10) *Boot Loader:* An on-chip boot loader can optionally be enabled allowing booting the processor and downloading applications without any external boot prom. This feature is mostly suitable for FPGA implementations.
11) *Instruction Pipeline:*The LEON3 integer unit uses a single instruction issue pipeline with 5 stages:
    a) *FE (Instruction Fetch):* If the instruction cache is enabled the instruction is fetched directly from the instruction cache. Otherwise the fetch is forwarded to the memory Controller. The instruction is valid at the end of this stage and is latched inside the IU.
    b) *DE (Decode):* The instruction is decoder and the operands are read. Operands may come from the register file or from internal data by pass. CALL and Branch target addresses are generated in this stage.
    c) *EX (Execute):* ALU, logical and shift operations are performed. From memory operations (e.g. LD) and for JMPL/RETT the address is generated. Store data read out in the E-stage is written to the data cache at this stage.
    d) *ME (Memory):* Data cache is accessed. For cache reads the data will be valid by the end of this stage at which point it is aligned as appropriate. Store data read out in the E-stage is written to the data cache at this time.
    e) *WR (Write):* The result of any ALU, logical, Shift cache read operations are written back to the register file. A chained secondary controller for up to 32 extra interrupts is also available.
12) *Parallel I/O Port:* A 16-bit parallel I/O port is provided. Each bit can be programmed to be an

input or an output. Some of the bits have alternate usage such as UART inputs/outputs and external interrupts inputs.

13) *AMBA On-Chip Buses:* The processor has a full implementation of AMBA AHB and APB on-chip buses. A flexible configuration scheme makes it simple to add new IP cores. Also all provided peripheral units implement the AMBA AHB/APB interface making it easy to add more of them, or reuse them on other processors using AMBA.
14) *Boot Loader:* An on-chip boot loader can optionally be enabled allowing booting the processor and downloading applications without any external boot prom. This feature is mostly suitable for FPGA implementations.
15) *Watchpoint registers***:**To aid software debugging, up to four watchpoint registers can be configured. Each registercan cause a trap on an arbitrary instruction or data address range.

IV. Basic Principle of Digital FIR Filter

The term “digital filter” refers to the computational process or algorithm by which a digital signal or sequence of numbers (acting as inputs) is transformed into a second sequence of numbers termed the output digital signal. Digital filters involve signals in the digital domain (discrete time signals), whereas analog filter relate signals in analog domain (continuous time signals). Digital filters are used extensively in applications such as digital image processing, pattern recognition and spectrum analysis [7]. Some of the advantages of using digital filter over analog filter are:

- High reliability
- High accuracy
- No effect of component drift on system performance.
- Another important advantage of digital filter when implemented with programmable processor is the ease of changing filter parameters to modify the filter characteristics.

V. Integration of Dsp unit with LEON Processor

We propose a design and implementation of embedded systems based on real time operating system and the leon3 processor. Following figure shows the typical structure of an application developed in this environment.

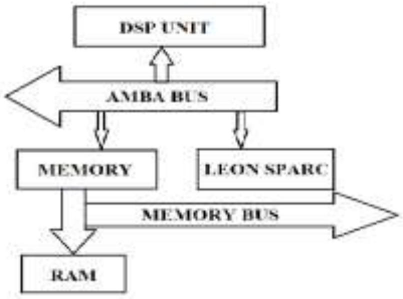

Fig 2 -Interfacing Of DSP Unit with LEON3 Processor

The proposed work uses the soft core for implementing the processor. Soft core will be simulated by using a simulation tool like Modelsim. For Synthesis it is proposed to use the synthesis tools like Leonardo Spectrum, Synopsys, Xilinx ISE etc. It will be emulated on VIRTEX FPGA development board.

A. DSP UNIT

The most important feature of a P is its ability to support repetitive and numerically intensive tasks. This ability is usedin its calculation of Fourier transforms, multi-filter systems and correlation calculations. The ability to perform a multiply-accumulate operation in a single clock cycle is key. The multiply accumulate is integrated into the data path. Specialized program control is another important aspect of a DSP. Control of loops for highly iterative algorithms is accomplished with a hardware counter and repeat buffer [8].

It is a 16-bit processor with only 144 words of data memory. The separate program memory can hold up to 4K words of instructions and coefficient data. The Instructions were broken into six categories: ALU functions, special operation functions, branch functions, multiplication functions, shift functions, and load functions. Instructions are further broken down depending on number of operands, source type, and special functionality and supports both DSP-specific (numeric-intensive) and general purpose operations [9].

VI. DEVELOPMENT BOARD

LEON3 processor fits in

- XLINX VIRTEX XCV300 (or upper)
- Altera FLEX10K200
- APEX 20K200 (or upper)
- VIRTEX FPGA development board will be used as it has sufficient memory for storage of programs.

References

[1] GRLIB IP Library User’s Manual Version 1.1.0 B4113 January, 2012, Copyright Aeroflex.

[2] Declan Staunton, “Successful use of an open source processor in a commercial ASIC",D&R Industry articles.

[3] Frank vahid, Tony Givargis, "Embedded System Design", John Wiley & Sons Inc., 2002.

[4] E Daniel Mattsson and Marcus Christensson Gothenburg, "Evaluation of synthesizable CPU cores ", 2lst of December 2004.

[5] Gaisler Research, “LEON3 processor user's manual", July 2005.

[6] Aeroflex Gaisler, SPARC V8 32 bit Processor LEON3/LEON3-FT Companion Core Datasheet, Copyright © March 2010 Aeroflex Gaisler AB.

[7] Miodrag Bolic , FIR filters

[8] Chang Choo, Jeff Chung, James Fong, Shinghin Eddy Cheung, Implementation of Texas Instruments TMS32010 DSP Processor on Altera FPGA, Department of Electrical Engineering, San Jose State University.

Al Lovrich and Ray Simar, Jr., Implementation of FIR/IIR Filters with the TMS32010/TMS32020, Digital Signal Processing – Semiconductor Group Digital Signal Processing Solutions 1989, TEXAS INSTRUMENTS.