International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:3, 2009

64 bit Computer Architectures for Space Applications – A study

Niveditha Domse, Kris Kumar, and K. N. Balasubramanya Murthy

***Abstract*—**The more recent satellite projects/programs makes extensive usage of real – time embedded systems. 16 bit processors which meet the Mil-Std-1750 standard architecture have been used in on-board systems. Most of the Space Applications have been written in ADA. From a futuristic point of view, 32 bit/ 64 bit processors are needed in the area of spacecraft computing and therefore an effort is desirable in the study and survey of 64 bit architectures for space applications. This will also result in significant technology development in terms of VLSI and software tools for ADA (as the legacy code is in ADA).

There are several basic requirements for a special processor for this purpose. They include Radiation Hardened (RadHard) devices, very low power dissipation, compatibility with existing operational systems, scalable architectures for higher computational needs, reliability, higher memory and I/O bandwidth, predictability, real-time operating system and manufacturability of such processors. Further on, these may include selection of FPGA devices, selection of EDA tool chains, design flow, partitioning of the design, pin count, performance evaluation, timing analysis etc.

This project deals with a brief study of 32 and 64 bit processors readily available in the market and designing/ fabricating a 64 bit RISC processor named *RISC MicroProcessor* with added functionalities of an extended double precision floating point unit and a 32 bit signal processing unit acting as co-processors. In this paper, we emphasize the ease and importance of using Open Core (OpenSparc T1 Verilog RTL) and Open "Source" EDA tools such as Icarus to develop FPGA based prototypes quickly. Commercial tools such as Xilinx ISE for Synthesis are also used when appropriate.

***Keywords*—**RISC MicroProcessor, RPC – RISC Processor Core, PBX – Processor to Block Interface part of the Interconnection Network, BPX – Block to Processor Interface part of the Interconnection Network, FPU – Floating Point Unit, SPU – Signal Processing Unit, WB – Wishbone Interface, CTU – Clock and Test Unit

.

Niveditha Domse, is a Member of Technical Staff, Research and Development at Cognitaa Pvt. Ltd., #1251, 32nd G cross, Jayanagar 4th 'T" Block, Bangalore-560041, India (e-mail: niveditha.work@gmail.com).

Kris Kumar is CEO of Cognitaa Pvt. Ltd. and Professor at Dept of Computer Science and Engineering, PES Institute of Technology, 100 – ft, Ring Road, BSK III Stage, Bangalore-560085, India (phone: +91(80)2672 1983 / 2108; e-mail: kris.kumar@pes.edu).

K. N. Balasubramanya Murthy is Principal, Director and Professor at Dept of Information Science and Engineering, PES Institute of Technology, 100 – ft, Ring Road, BSK III Stage, Bangalore-560085, India (e-mail: principal@pes.edu).

I. Introduction

COMPUTER Architectures, their evolution, performance and usage are well documented in [1, 2, 3, 4, 5, 6]. Space flight systems and associated designs and issues can be found in [7, 8].

Trends that affect Computing have had a major impact on Processor Design Evolution. Growing complexity of a processor design and worsening memory latency are two recent trends of importance which should be taken into account while designing new Processors.

Another most important paradigm in the evolution of Computer Architectures is the advent of the ***Reduced Instruction Set Computer (RISC)*** and ***Complex Instruction Set Computer (CISC).*** The focus of this paper is on scalable RISC Architectures.

The usage of Field programmable Gate Arrays and Reconfigurable systems constitutes yet another step in quick prototyping [9, 10, 11].

II. RISC MicroProcessor Overview

The development of a new and unique Architecture with a scalable Processor called RISC MicroProcessor is the thrust of this paper. RISC MicroProcessor has 3 main computational blocks: A RISC Processor Core (RPC), an extended double precision Floating Point Unit (FPU) and a Signal Processing Unit (SPU). Both the FPU and the SPU act as Coprocessors to the RPC. The Co-Processor approach to Floating Point Processing and Signal Processing allows for future extensions without compromising the meaning and importance of RISC processing. The proven Sun SPARC v8/v9 Architectures with ***suitable and significant*** modifications to its derivative ***OpenSparc T1*** is the ***baseline*** for the RISC MicroProcessor Core (RPC) [12, 13. 14, 15, 16, 17, 18, 19]. The FPU supplied in T1 will be modified/ enhanced in the RISC MICROPROCESSOR implementation.

A simple Interconnection Network contains the PBX (Processor to Block Interface part of the Interconnection Network) and BPX (Block to Processor Interface part of the Interconnection Network) interfaces that allow on-chip communication between the Main RISC Processor core and the other functional units including the FPU and the SPU.

This Interconnection Network also connects a Wishbone Interface that is useful to connect to a variety of External Memory devices through an external/internal Memory Controller. This is mainly for proof of concept and related

experiments. *All the Wishbone Interfaces and related devices may be removed prior to the fabrication of the chip, if so desired.*

A special RISC Processor DEBUG Port is also attached to the RISC MicroProcessor to facilitate tracing/monitoring of key signals. The Top Level Architecture of the RISC MicroProcessor and the associated blocks are as illustrated in Fig. 1.

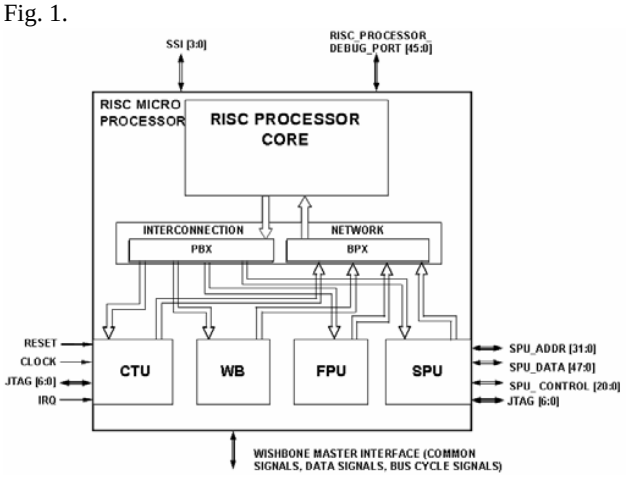

Fig. 1 Top Level RISC MICROPROCESSOR Blocks

III. RPC – RISC Processor Core

RPC – RISC Processor Core is a modified version of OpenSparc T1, which is a derivative of the SPARC v9 Architecture.

The main features of this core are:

1) A large windowed register file: At any one instant, a program sees 8 global integer registers plus a 24-register window of a larger register file. The windowed registers can be used as a cache for procedure arguments, local values, and return addresses. This implementation has 2 windows, though a maximum of 32 windows is permitted under the SPARC v9 architecture
2) Instruction size 32 bits wide
3) Integer registers – depends on the number of windows
4) Separate Floating point Registers
5) A linear address space with 64 bit addressing
6) Few and simple instruction formats: All instructions are 32 bits wide, and are aligned on 32 bit boundaries in memory. Only load and store instructions access memory and perform I/O
7) Few addressing modes: A memory address is given as either "register + register" or "register + immediate"
8) Hardware support for 4 threads in the Core
9) Optional 16 Kbytes of primary (Level 1) instruction cache
10) Optional 8 Kbytes of primary (Level 1) data cache

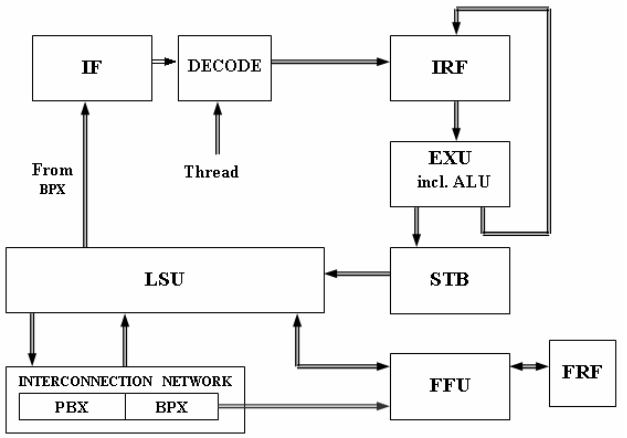

Fig. 2 RISC Processor Core (Parts of Internal Block Diagram)

IV. The Interconnection Network

The Interconnection Network manages the communication among the RISC Processor Core (RPC), the Clock and Test Unit (CTU), the Wishbone (WB), the Signal Processing Unit (SPU) and the Floating Point Unit (FPU). These functional units communicate with each other by sending packets, and the Interconnection Network arbitrates the packet delivery.

The RPC can send a packet to any one of the WB, the SPU, or the FPU or the CTU. Conversely, packets can also be sent in the reverse direction, where any of the WB, the SPU, the FPU or the CTU can send a packet to RPC.

The Interconnection Network consists of two main blocks – Processor to Block Interface (PBX) and the Block to Processor Interface (BPX). The PBX block manages the communication from RPC (source) to any of the WB, SPU, FPU or CTU (destination). The BPX manages communication from WB, SPU, FPU or CTU (source) to RPC (destination).

V. The Signal Processing Unit (SPU)

The Signal Processing Unit (SPU) in the RISC MicroProcessor is adapted from the ADSP-2106x SHARC (Super Harvard Architecture Computer) series [20]. It is a 32 bit Digital Signal Processor (DSP). The main features of the SPU are:

1) Three parallel Computation Units — Multiplier, ALU, and Shifter
2) Separate Data and Program memories
3) Optional internal instruction cache to execute every instruction in a single cycle
4) 32 Bit IEEE Floating Point Computation
5) Data Register File
6) Data Address Generators (DAG1, DAG2)
7) Program Sequencer with Instruction Cache
8) Interval Timer
9) External Port for Interfacing to Off-Chip Memory & Peripherals
10) JTAG Test Access Port

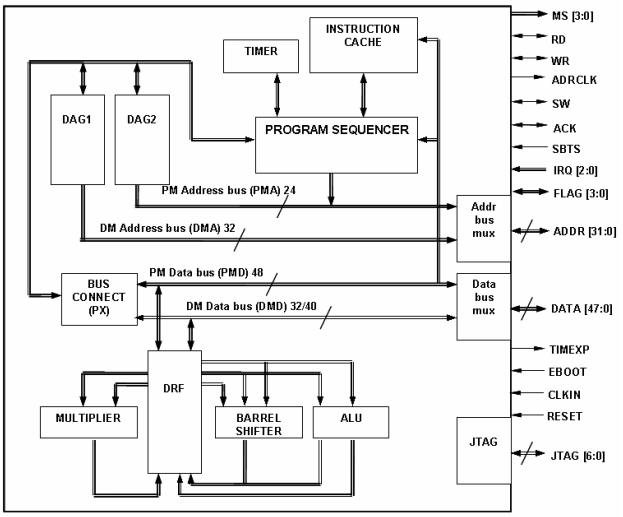

Fig. 3 The Signal Processing Unit (SPU)

VI. The Floating Point Unit (FPU)

The Floating Point Unit (FPU) is based on the on the IEEE standard for Binary floating point Arithmetic, IEEE standard 754-1985 [21, 22, 23, 24]. Its implementation will be as a Co-Processor of the RISC Processor Core (RPC). The main features of this block are:

1) The FPU implements the SPARC v9 floating point instruction set with the following exceptions:
   a. FSQRT(s,d), and all quad precision instructions
   b. Move-type instructions executed by the Floating point Frontend Unit (FFU)
   c. Loads and stores (the FFU executes these operations)
2) The Floating point Register File (FRF) and Floating point State Register (FSR) are not physically located within the FPU. The RISC Processor Core FFU owns the FRF and FSR. The FPU complies with the IEEE 754 standard.
3) One instruction per cycle may be issued from the FPU input FIFO queue to one of the three execution pipelines the FPU.
   a. One instruction per cycle may complete and exit the FPU.

Support for all IEEE 754 floating point data types (normalized, denormalized, NaN, zero, infinity). A denormalized operand or result will never generate an unfinished_FPop trap to the software. The hardware provides full support for denormalized operands and results.

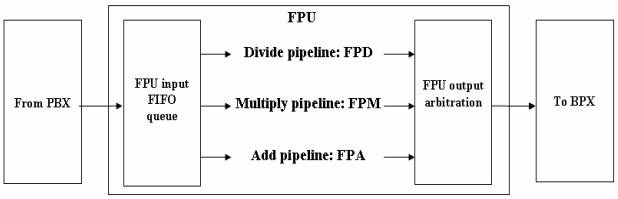

Fig. 4 Functional Block diagram of Floating Point Unit (FPU)

VII. Wishbone interface for External Memory (WB)

Wishbone interface for External Memory (WB) allows designers to combine several designs written in Hardware Description languages in a standard way [25]. Wishbone adapts well to common topologies such as point-to-point, crossbar switches, etc. Wishbone is defined to have 8, 16, 32, and 64-bit buses. All signals are synchronous to a single clock but some slave (the term "slave" is described below) responses must be generated combinatorially for maximum performance.

The WISHBONE System-on-Chip interconnect defines two types of interfaces which are called MASTER interface and SLAVE interface. The MASTER interfaces are those that are capable of generating bus cycles and the SLAVE interfaces are those that are capable of receiving bus cycles. Connections between the MASTER and SLAVE interfaces can be established in a number of ways such as a simple point-to-point connection.

WISHBONE signals can be combined between the MASTER and SLAVE interfaces. The WISHBONE signals can be divided into three groups called as common signals, data signals and bus cycle signals. The RPC is connected to the Wishbone **(revision B.3)** interface through the Interconnection Network.

VIII. Clock and Test Unit (CTU)

A. Clock Architecture

The three synchronous clock domains on the T1 processor are expected to be converted into a suitable set of clock domains – one for the RPC, one for the SPU, one for the memory interfaces, etc. All of these are sourced from the same phase-locked loop (PLL). Synchronization pulses are generated to control transmission of signals and data across clock domain boundaries.

The clock and test unit (CTU) is responsible for resetting the PLL and counting the lock period. Once sufficient time has passed to enable the PLL to have locked, the clock control logic is responsible for enabling distribution.

Further information on the T1 clock architecture, including sequences for changing input clock frequency or clock ratios, clock control registers, PLL bypass, stop and scan support, and clock stretch, can be found in the *UltraSPARC T1 Processor Supplement*.

B. Reset Architecture

The processor has two flavors of reset – power-on reset (POR) and warm reset. POR is defined as the reset event arising from turning power on to the chip and is triggered by assertion of power-on reset. Warm reset encompasses all resets when the chip has been in operation prior to the event triggering the reset.

C. JTAG and Boundary Scan

The CTU contains the JTAG block, which enables access to the shadow scan chains. The unit also has a control register (CREG) interface that enables the JTAG to issue reads of any I/O-addressable register, some ASI locations, and any memory location while the processor is in operation.

The T1 processor contains an IEEE 1149.1-compliant test access port (TAP) controller with the standard five-pin JTAG interface. In addition to the standard IEEE 1149.1 instructions, roughly 20 private instructions provide access to the processor's design for testability (DFT) features. On further investigation, similar features will be incorporated in RISC MicroProcessor.

More general details can be found in [26, 27, 28, 29, 30].

D. Serial system interface (SSI)

The T1 processor has a 50 Mbyte/sec serial system interface (SSI) that connects to an external application specific integrated circuit (ASIC), which in turn interfaces to the boot read-only memory (ROM). In addition, the SSI supports PIO accesses across the SSI, thus supporting optional control status registers (CSR) or other interfaces within the ASIC.

IX. Signal Description

The RISC MICROPROCESSOR main interface signals and the corresponding pin counts are illustrated in Fig. 5.

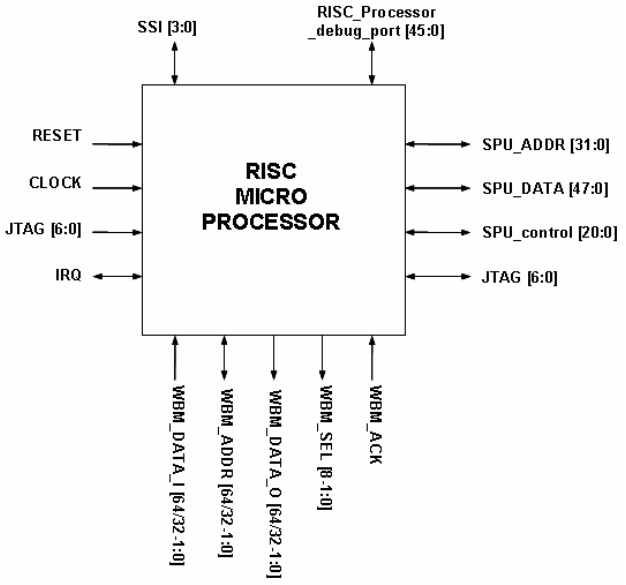

Fig. 5 Signal description of RISC MICROPROCESSOR

X. Conclusion

The growth of OpenCore Technology such as OpenSparcT1 is a favorable trend. This work demonstrates that Processor Architects and Designers can quickly prototype and demonstrate new variants of 64 bit Processors [31] using FPGA implementations. Special Coprocessors are added with relative ease to customize the System on Chip. Space borne Electronics' unique needs such as Deterministic behavior (removal of Cache, Polled I/O as opposed to Interrupt driven I/O) and pin count limitations have been addressed to show feasibility. Future work involves 128 by 128 bit Floating point unit and addressing DSP requirements in other ways.

TABLE I
Detailed Description of Signals (RISC MICROPROCESSOR)

PIN	TYPE	NAME
CLOCK	I	System Clock
RESET	I	System reset
IRQ	I	System Interrupt
WBM_ACK	I	Wishbone Acknowledge
WBM_DATA_I[64/32-1:0]	I	Wishbone Input Data Bus
JTAG[6:0]	I/O	Standard IEEE
WBM_ADDR[64/32-1:0]	I/O	Wishbone Address Bus
WBM_DATA_O[64/32-1:0]	O	Wishbone Data Bus
WBM_SEL[8-1:0]	O	Wishbone data select bits
JTAG[6:0]	I/O	IEEE Standard
SPU_CONTROL[20:0]	I/O	Signal Processing unit Control signals
SPU_DATA[47:0]	I/O	Signal Processing unit data bus
SPU_ADDR[31:0]	I/O	Signal Processing unit Address bus
RISC_PROCESSOR_DEBUG_PORT[45:0]	I/O	RISC MicroProcessor Debug signals
SSI[3:0]	I/O	Bootstrap serial Interface

Total estimated signal pins for 64 bit RISC Architecture = 368 pins
Total estimated signal pins for 32 bit RISC Architecture = 272 pins
The total Lookup Table / Gate count estimates are currently unavailable

Acknowledgment

We thank Dr. V. K. Agrawal, Group Director, Control Systems Group, ISRO, ISAC; Mr. K. Parameshwaran, Division Head, Control Electronics Division, ISRO, ISAC; Dr. J.K. Kishore, Focal Point and Section Head, OBCS/ CSG / ISAC, ISRO and PPEG Group ISRO, Sun Microsystems Group, MEDF and Lab support at ISRO, ISAC, Bangalore for their valuable support and reviews.

References

[1] Cerruzi, P. A History of Modern Computing, 2nd Edition, MIT Press, May 2003.
[2] Nisan, N and Schocken, S. The Elements of Computing Systems, Building a Modern Computer from First Principles, MIT Press, June 2005.
[3] Hennessey, J.L. and Patterson, D.A. Computer Architecture – A Quantitative Approach, 4th Edition, Morgan Kaufman Publishers, 2007
[4] Hayes, J. P. Computer Architecture and Organization, 3rd Edition, McGraw-Hill, 1997.
[5] D. A. Patterson and J. L. Hennessey, "Computer Organization and Design – The Hardware/Software Interface", Elsevier Publications, South East Asia, 3rd Edition, 2005.
[6] C. Hamacher, Z. Vranesic and S. Zaky, "Computer Organization", 5th Edition, McGraw-Hill, 2002.
[7] A scientific study of the problems of digital engineering for space flight systems, NASA office of Logic Design, Dec 2004.
[8] J. Gaisler, "The LEON Processor User's Manual", Version 2.3.7, Gaisler Research, August 2001.
[9] W. Wolf, "FPGA – Based System Design", Pearson Education Limited, First Indian Reprint, India, 2005.
[10] Dhir, A. The Digital Consumer Technology Handbook: A Comprehensive Guide to Devices, Standards, Future Directions, and Programmable Logic Solutions, Xilinx, Inc. 2005.
[11] **BRASS –** Berkeley Reconfigurable Architectures, Systems and Software : http://brass.cs.berkeley.edu/

[12] The SPARC Architecture Manual, *Version 8*, SAV080SI9308 SPARC International Inc., Menlo Park, CA 94025.
[13] The SPARC Architecture Manual, *Version 9*, SAV09R1459912 Prentice Hall, Englewood Cliffs, NJ 07632.
[14] Boney, Joel [1992], "SPARC Version 9 Points the Way to the Next Generation RISC," *SunWorld*, October 1992, pp. 100-105
[15] Catanzaro, Ben, ed. *The SPARC Technical Papers*, Springer-Verlag, 1991.
[16] Cmelik, R. F., S. I. Kong, D. R. Ditzel, and E. J. Kelly, "An Analysis of MIPS and SPARC Instruction Set Utilization on the SPEC Benchmarks," *Proceedings of the Fourth International Symposium on Architectural Support for Programming Languages and Operating Systems*, April 8-11, 1991.
[17] Ditzel, David R. [1993]. "SPARC Version 9: Adding 64 Bit Addressing and Robustness to an Existing RISC Architecture." Videotape available from University Video Communications, P. O. Box 5129, Stanford, CA, 94309
[18] Garner, R. B. [1988]. "SPARC: The Scalable Processor Architecture," *SunTechnology*, vol. 1, no. 3, Summer, 1988; also appeared in M. Hall and J. Barry (eds.), *The SunTechnology Papers*, Springer-Verlag, 1990, pp. 75-99.
[19] OpenSPARC™ T1 Microarchitecture Specification, Sun Microsystems, Part No. 819-6650-10, August 2006, Revision A.
[20] ADSP-2106x SHARC® Processor User's Manual, Revision 2.1, Part Number 82-000795-03, March 2004, Analog Devices, Inc., Norwood, Mass. 02062-9106.
[21] IEEE Standard for Binary Floating Point Arithmetic, IEEE Std 754-1985, IEEE, New York, NY, 1985.
[22] IEEE Computer Society (1985), *IEEE Standard for Binary Floating-Point Arithmetic*, IEEE Standard 754-1985.
[23] Hollasch S**,** "IEEE Standard 754 Floating Point Numbers", Feb 2005
[24] http://en.wikipedia.org/wiki/Floating_point_unit
[25] "Wishbone Datasheet" revision B.3 – www.opencores.org
[26] IEEE Standard 1149.1-1990: Standard Test Access Port and Boundary-Scan Architecture, 1990.
[27] Maunder, C.M. & R. Tulloss, Test Access Ports and Boundary Scan Architectures, IEEE Computer Society Press, 1991.
[28] Parker, Kenneth. *The Boundary Scan Handbook*, Kluwer Academic Press, 1992.
[29] Bleeker, Harry, P. van den Eijnden, & F. de Jong, *Boundary-Scan Test—A Practical Approach*, Kluwer Academic Press, 1993.
[30] Hewlett-Packard Co, "*HP Boundary-Scan Tutorial and BSDL Reference Guide"*, Hewlett-Packard Part No. E1017-90001, 1992.
[31] Dr. Kris Kumar, Dr. K.N.B. Murthy, Niveditha Domse, , *"64 bit Computer Architectures for Space Applications – A study"*, Research Project report submitted to ISRO as a part of RESPOND PROGRAM under Grant # 9/2/77/2006 jointly undertaken with ISRO and PES Institute of Technology, Bangalore, Nov 2007.