Hyper Pipelined RISC Processor Implementation- A Review

Simran Rana (1) Rajesh Mehra (2)

HIET shahpur, kangra, H.P(1) NITTTR CHD (2)

Abstract

Reduced Instruction Set Computing or RISC pronounced as Risk is a CPU design strategy based on its insight construction that simplifies the instructions which provides higher performance. Pipelining is the concept of overlapping of multiple instructions during execution time. Pipeline splits one task into multiple subtasks. In the embedded application the concept of Core Multiplier multiplies the functionality of cores, bus-systems or complete sub designs (hyper pipelining). It implements registers (called pipes) in the design to create CMF independent designs. Since only registers are inserted, the resulting area is much less than duplicating the complete design. The result is a much smaller ASIC or lesser FPGA size.

Introduction

The RISC architecture is an attempt to produce more CPU power by simplifying the instruction set of the CPU. Both RISC and CISC architectures have been developed as an attempt to cover the semantic gap.

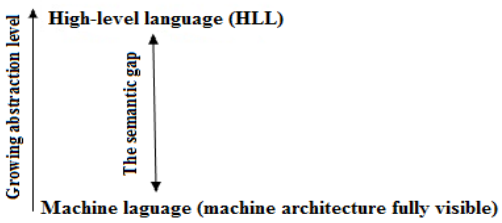

Fig.1.1 Description of semantic gap for the high level language to be converted to the machine level languages [2]

RISC stands for "Reduced Instruction Set Computer". The IBM was the first company to define the RISC architecture in the 1970's. The RISC processor used for mobile phones will be known as Low end and mobile systems & by the beginning of the 21st century, the majority of low end and mobile systems relied on RISC architectures. The RISC Processor used for the design of super computer is known as High end RISC and supercomputing. [1]

1.1 Main Characteristics of RISC Architectures

1. RISC machines require lesser time for its design implementation. 2. RISC processor consume less power 3. RISC instructions are executed in single clock cycle, while most of CISC requires more than one clock cycle. 4. RISC system is more popular. 5. The current technology RISC processors support the floating point data type. 6. RISC machines mostly uses hardwired unit. 7. RISC machines use load and store instruction to access data from memory [2]

The principles of the RISC architecture guide the design of the previous generation of processors. These principles have accelerated the performance gains of these processors over their predecessors. The changes that spurred the performance gains of RISC were clear because they were changes in instruction set architecture (ISA). In the Post-RISC changes are relied on the compiler in order to increase performance to the instructions and hardware checked the legality of multiple simultaneous instruction issue to the implementation of the architecture. The example Post-RISC CPU is a 4-way superscalar processor so each pipeline step processes groups of four instructions. Instructions are first

fetched, decoded, and then buffered. Instructions can be dispatched to execution units out of program order as resources and operands become available. General Block Diagram of the Generic Post-RISC Processor with its different blocks is shown below: -

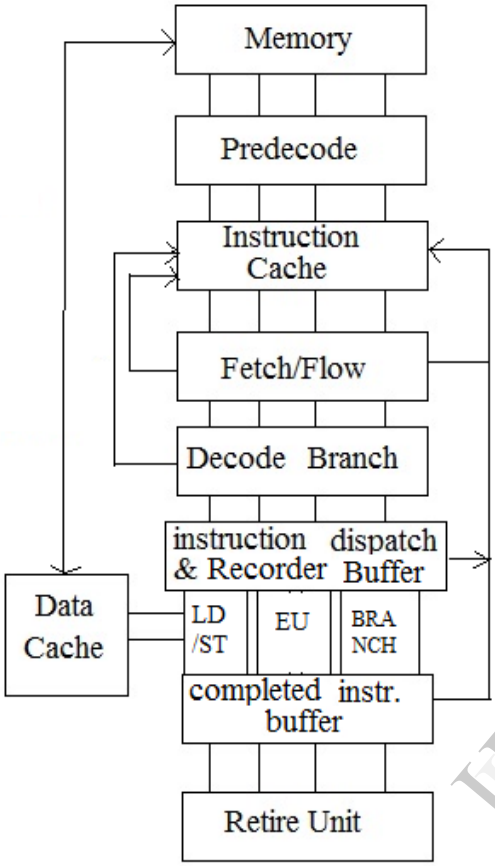

Figure 1.2 General Block Diagram of the Generic Post-RISC Processor [3]

2. Basics of Pipelining

Pipelining is a technique of decomposing a sequential process into sub-operations, with each sub process being executed in a special dedicated segment that operates concurrently with all other segments.

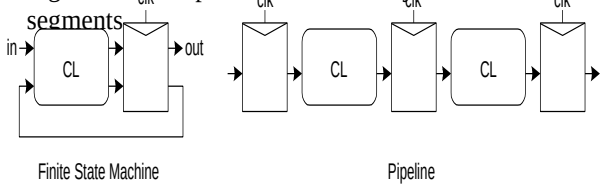

Figure 1.3 Comparison of FSM (finite state machine) with Pipelined sequential structure

3. Types of pipelining

3.1 Super Pipelining
3.2 Super scalar pipelining
3.3 Hyper pipelining technique

3.1 Super pipelining

Pipelining is the concept of overlapping of multiple instructions during execution time. Pipeline splits one task into multiple subtasks. These subtasks of two or more different tasks are executed parallel by hardware unit. The concept of pipeline can be same as the water tab.

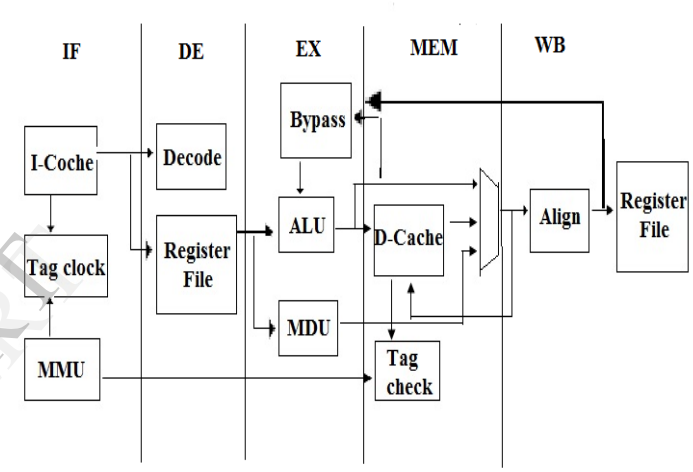

Figure 1.4, Five-Stage pipelined structure of RISC (super pipelining) [4]

3.2 Super scalar processor/Machine

The scalar machine executes one instruction one set of operands at a time. The super scalar architecture allows on the execution of multiple instruction, at the same time in different pipelined hardware. Here multiple processing elements are used for different instruction at the same time. Pipeline is also implemented in each processing elements. The instruction fetching units fetch multiple instructions at a time from cache. There should be multiple execution units so that multiple instructions can be executed at the same time. The slowest stage among fetch, decode and execute will determine the overall performance of system. Ideally these three stages should be equal fast.[5]

3.3 Hyper pipelining

In this Thesis, a method is discussed that how the functionality of a core can be multiplied by adding registers to the core. It does not only provide the less area usage compared to its individual instantiations, but it can also provide impact on the system performance as a whole. This method is called "hyper pipelining" here the hyper pipelined complex RISC core (OR1200 from Open Cores) is discussed. Hyper pipelining is a technique where the Core Multiplier multiplies the functionality of cores, bus-systems or complete sub designs. It implements registers (called pipes) in the design to create CMF independent designs, whereas CMF can be any number greater than 1[5]

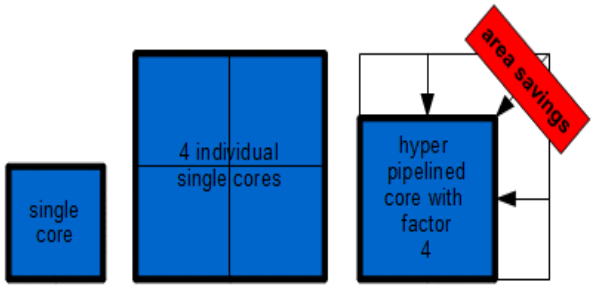

Figure 1.5 Comparison of combined 4 single RISC core with hyper pipelined RISC core with CMF=4(Core Multiplication Factor) [5]

4. Applications of RISC Core

Low-power embedded processors are used in a wide variety of applications including: Cars, Phones, Digital cameras Printers, and other such devices. Applications may include its use in vending machines, ATMs, mobile phones, portable gaming kits and control of robotics problems. The RISC Processor used for the design of cell phones are known as Low end RISC and supercomputing. Some examples are given below-

1. The ARM architecture dominates the market for low power and low cost embedded systems (typically 100–1200 MHz in 2011). It is used in a number of systems such as most Android-based systems, the Apple iPhone and iPad, RIM devices, Nintendo Game Boy Advance and Nintendo DS, etc.
2. The MIPS line, (at one point used in many SGI computers) and now in the PlayStation, PlayStation 2, Nintendo 64, PlayStation Portable game consoles, and residential gateways like Linksys WRT54G series.
3. Hitachi's SuperH, originally in wide use in the Sega Super 32X, Saturn and Dreamcast,
4. Atmel AVR used in a variety of products ranging from Xbox handheld controllers to BMW cars.

The RISC Processor used for the design of super computer are known as High end RISC and supercomputing. Some examples are given below

1. SPARC, by Sun Microsystems (now Oracle), and Fujitsu
2. Hewlett-Packard's PA-RISC, also known as HP-PA, (discontinued at the end of 2008)
3. Alpha, used in single-board computers, workstations, servers and supercomputers from Digital Equipment Corporation, Compaq and HP, discontinued as of 2007.
4. IBM's Power Architecture, used in many of IBM's supercomputers, midrange servers and workstations.[2][6].

5. Processor based applications

All the applications which are mentioned above are broadly based on the different types of RISC cores and these are

5.1 Scalar RISC Architecture Core

Here the authors worked on Graph Modelling of Parallelism in super scalar RISC Architecture. Here the SGM (System Graph Model) is used which is a fundamental source of information regarding the resource dependencies, intra-instruction; inter instruction parallelism, and the cost-performance of the architecture. In addition to this the paper shows new technique called Hierarchical Identification of Parallelism (HIP), which is a systematic approach to identify parallelism.[7] In this paper the Software based self-testing of embedded processor cores was discussed, which effectively partitions the testing effort between low-speed external equipment and internal processor resources. It is has been proposed as an alternative way against the classical hardware built-in self-test techniques. It provides significant advantages. This methodology for processor cores is used for producing compact test code sequences developed with a limited engineering effort. A low-cost software-based self-testing methodology for processor cores is presented with the aim of achieving high fault coverage for the processor core. [8] Here the implementation of 32 bit RISC processor

with microprocessor without interlocked pipeline stages (MIPS) is presented. Here a VHDL implementation is used to reduce the instruction set present in the programmable memory. Because of this the processor will contain the necessary logics for the implementation that requires fewer gates to be synthesized in the programmable matrix and the speed of the target processor is increased. It is now possible to implement a complete system on single chips (SOCs)[9]. The authors has worked on Design & performance analysis of 8-bit RISC Processor using Xilinx Tool. RISC or Reduced Instruction Set Computer is a design philosophy that has become a mainstream in Scientific and engineering applications. The main objective of this paper is to design and implement an 8-bit Reduced Instruction Set (RISC) processor using XILINX Spartan 3E tool. . The module functionality and performance issues like area, power dissipation and propagation delay are analyzed at 90 nm process technology using SPARTAN 3E XCS500E XILINX tool.[10]

5.2 Comparison of DSP Processor with RISC

In this paper it has been proved that a DSP processor can be implemented using strictly the RISC design philosophy. The architecture implements a set of DSP instructions that support an efficient mapping of common DSP algorithms to the processor. The basic 32 bit RISC architecture was introduced which provides the following characteristics: fast DSP processing: simple two stage decode/execute pipeline with single cycle delayed branches , almost all ALU instructions are single cycle providing fast loop control, address calculation and index updates.[11] In this paper the features of Digital signal processors are combined with microcontrollers. Various attempts have been made to combine the two processor functions in single architecture, but two problems are always being unsolved. These are different data and program memory hierarchy choices in speed and size, and different real-time control needs. This paper reviews the basic processing requirements for digital signal processing (DSP) and controllers and shows how a new 32-bit RISC architecture has resolved these conflicts and successfully integrated the two functions seamlessly into one processor core.[12]. In this paper the authors discussed about mathematical analysis and empirical evaluation of the solid state equation $Power_{cmos}$ = P = C V2 f N %N. In this paper they identify a measurable metric for evaluating relative advantages of ASIC, DSP, and RISC architectures for embedded applications. Relationships are examined which can help predict relative future architecture performance as new generations of CMOS solid state technology become available.[13] Here the authors worked on a combined architecture. Here the YS-RDSP which is a scalar processor is designed, which merges a RISC microprocessor with a DSP processor to be suitable for embedded applications. The YS-RDSP can execute maximum 4 instructions in parallel at the same time. In order to reduce the size of programs the YS-RDSP has variable instruction length of 16-bit and 32-bit. The YS-RDSP has 8-kbyte ROM and 8-kbyte RAM on chip. [14]

5.3 Floating point RISC Processor

They have worked on the Design of a 32-bit floating-point RISC microprocessor. LS RISC (Li Shan RISC) microprocessor is a 32bits floating-point embedded microprocessor designed by Xi'an Microelectronic Technology Institute independently. LS RISC processor supports six type's fixed-point number and the single precision floating-point number specified by IEEE754. The main features of the LS RISC that it is having thirty 43bits three-ported general registers, a 32bits PC register, a PSR register and a SR register in LS RISC microprocessor.[15] This paper presents the design and implementation of a novel VLIW digital signal processor (DSP) for multimedia applications. The DSP core embodies a distributed & ping-pong register file, which saves 76.8% silicon area and improves 46.9% access time of centralized ones found in most VLIW processors by restricting its access patterns. A hierarchical instruction encoding scheme is also adopted to reduce the program sizes to 24.1ò26.0%.[16]

5.4 DLX RISC Processor or base RISC processor

This paper describes the algorithm for a tool which is used for interlock & forwarding logic for sequential implementation and the correctness is formally verified. They used a standard DLX RISC processor as a Base processor. The DLX is essentially a RISC Processor design. The DLX has simple 32-bit load/store architecture, somewhat unlike the modern MIPS CPU. The interlock and forwarding logic is considered the tricky part of a fully-featured pipelined microprocessor and especially debugging these parts delays the hardware design process considerably. [17] In this paper, hybrid neural network processor (HANNP) is designed in VLSI. The HANNP has RISC based architecture leading to an effective general digital signal processing and

artificial neural networks computation. The architecture of a HANNP including the general digital processing units such as 64-bit floating-point arithmetic unit (FPU), a control unit (CU) and neural network processing units such as Neural Network Processing Unit (NNPU), specialized neural data bus and interface unit, etc. The HANNP is modeled in Verilog HDL and implemented with FPGA. Character recognition problems and Kohonen self-organization problems are applied to the proposed HANNP to justify its applicability to real engineering problems.[18] A true 16-bit RISC processor has been designed using VHDL. Four stage Address Line Input Port (viz. instruction fetch stage, instruction decode stage, execution stage and memory/lO-write back stage) pipelining is used to Processor Read improve the overall CPI (Clock Cycles per Instruction). The processor has one input port, one output port and six hardware vectored interrupts along with 16-bit address bus and 16-bit data bus. The design has been implemented on FPGA for verification purpose.[19] Here the authors discussed that a further speed up can be achieved by RISC-based extensible processors if the incorporated custom functional units (CFUs) can execute functions with more than two inputs and one output which is based on recent study. However, there is no mechanisms has been discussed to execute multiple-input, multiple-output (MIMO) custom functions in a RISC processor. This paper presents an extension for single-issue RISC processors based on a CFU that can execute custom functions with up to six inputs and three outputs.[20] This paper provides an approach on pipeline design and optimization for a 32-bit embedded RISC3200 processor with configurable multimedia extension instructions. In order to incorporate & enhance the reduced instruction set and the multimedia extension instruction set in a unifying pipeline, there is a scalable super-pipeline technique is used. Here a data flow graph (DFG) with delay information, is helpful for us to locate the critical path of the pipeline stage. This paper presents a distributed data bypass unit and a centralized pipeline control scheme for achieving lower CPI.[21] In this paper the authors discussed 32-bit RISC processor for embedded application. They presented the paper with respect to the limitation of power and area in the embedded system. Here a Dual-issue technology is adopted to improve the performance. In the different pipeline stages to improve the frequency the complex logic of the dynamic scheduling algorithm is allocated. Whereas the lower power designs method is used to decrease the whole power. The processor is implemented by SMIC 0.18um CMOS technology which contains almost 5 million transistors.[22] in the present paper, authors presented the design and implementation of a 64-bit reduced instruction set (RISC) processor with built-in-self test (BIST) features. A built-in self-test (BIST) or built-in test (BIT) is a mechanism that permits a machine to test itself. Key features of the design including its architecture, data path, and instruction set are presented. The design is implemented using VHDL and verified on Xilinx ISE simulator. The proposed design may find applications where automation and control processing is required.[23] In this paper the authors discussed an LED Array Color Temperature Controlled Lighting System has been implemented using an 8 bit RISC IP Core for the lighting control system, as well as a Color Temperature controlling IP and Delta-Sigma DAC IP designed to control the system. The light sources are made of an LED Array, and the LEDs are configured to have 10 stick bars, such as 3 chips of white (30EA), daylight (30EA), red (30EA), green (30EA), and blue (30EA) 0.1W SMD.[24] In this paper the Design and Implementation of 64 Bit RISC Processor using system on chip (SOC) was done. In this paper, we present the design and implementation of a 64-bit reduced instruction set (RISC) processor with built-in-self test (BIST) features. A built-in self-test (BIST) or built-in test (BIT) is a mechanism that permits a machine to test itself. Key features of the design including its architecture, data path, and instruction set are presented. The design is implemented using VERILOG and verification is done on IUS (Intensive Unified Simulator) (CADENCE) simulator.[25] In this paper the System is built around high performance VLSI technology. Matrix manipulation is a parallel architecture of logical expressions. When VLSI is used to implement High Performance Gate Arrays it becomes most suitable for implementing parallel architecture. Advanced RISC Machine (ARM) is the Implementation result of the RISC microprocessor architecture.[26] Here authors have worked on 8-bit RISC Processor using Harvard Architecture. Usually a car is not considered as to be an electronic device. However, it certainly has many complex, smart electronic systems, such as the anti-lock brakes and the fuel-injection systems. The design of ALU was optimized so that it consumes fewer resources. Traditional approach to develop a digital system was to use a set of interconnected digital integrated circuits like counters, buffers, logic gates and memory.[27] In this paper, FPGA based implementation of 32-bit RISC processor was done. Here the design of general purpose processor with a 5 stage pipeline, to incorporate programmable resources in to a processor is used. As RISC processors have a CPI (clock per instruction) of one cycle, which was achieved due to the optimization of

each instruction on the CPU and a technique called pipelining. This technique allows each instruction to be processed in a set of number of stages. Which allows for the simultaneous execution of a number of different instructions, each instruction will be at a different stage in pipeline.[28]

5.5 ARM RISC Processor

Iin this paper a high performance 32-bit RISC processor was implemented using VLSI Technology. Some of the 32-bit RISC microprocessor like ARM, MIPS, and PowerPC etc. are used commercially. ARM is the most widely used because of its abundant instructions, high code density, small transistor counts and low power consumption. A 32-bit RISC microprocessor, FDU32, with instruction Sets compatible with ARM7TDMI is presented in this paper. They have changed the pipeline from 3-stage to 5-stage and modified the total system architecture, which results in notable improvement in max clock rate, CPI (clock per instruction).[29] In this paper the Architecture comparison of different aspects of popular embedded RISC was done. The comparison was done at the level of internal structure component like general purpose register file and stack based three-stage pipeline and five –stages, von Neumann architecture and Harvard architecture. As to do this, they compared several different aspects of computer architecture in the state-of-the-art and choose the most suitable sets to implement our RISC32 microprocessor.[30] Here AVS-M which is the recent mobile video coding standard of China is discussed. Currently, ARM cores are widely used in mobile applications because of their low power consumption. In this paper, a scheme of the AVS-M decoder real time implementation on 32 bit MCU RISC processor ARM920T (S3C2440) is presented. The algorithm, redundancy, structure and memory optimization methods to implement AVS-M real time are discussed in detail. The ADS, MCPS and simulation results show that the proposed AVS-M decoder can decode the QVGA image sequence in real-time with high image quality and has low complexity and less memory requirement.[31] here the implementation of the ARM9TDM, a 32-bit RISC processor based on the ARM9TDMI was done by the authors. This core is considered as the successor to the ARM7TDMI-S which is used for embedded applications requiring low power, small chip area, and high processing speed. The main features of the ARM9TDM are its use of a 5-stage pipelined data path and a Harvard architecture that has separate data and instruction interfaces. It also supports the ARMv4T instruction set architecture (ISA) that uses both the 32-bit ARM instructions and 16-bit Thumb instructions.[32] This paper provides a result regarding the design and implementation of a embedded processor, xCore_AHB whose main features are precise interrupt and exception and it is also compatible with ARMv4 architecture. The precise exception mechanism of this design provides not only the quick entrance of the interrupt handle programs but also the interrupt handle programs with the right return address by an additional program counter in write back (WB) stage of the pipeline. Here in this paper exception controller has saved about 30% of area with the traditional exception controller.The ARMv4 instruction set is an RISC instruction set.[33] In this paper the implementation of ARM processor as a 32*32 Bit Matrix Coprocessor was designed which is used for different applications like image processing, satellite communication etc. was done using VHDL language which was implemented on ARM processor. Various features of the ARM processor are small size, reduced power consumption and extended battery operation, whereas the main applications of the ARM processor are PDAs (Personal Digital Assistants) & Mobile phones. ARM was designed for 32-bit processor cores. The 32*32 bit matrix coprocessor is performing the different operations of matrix.[34]

5.6 MIPS RISC Processor

This paper describes a design methodology of a single clock cycle MIPS RISC Processor using VHDL to do the description, verification, simulation and hardware realization easily. There are three different formats for the RISC processor which is having fixed-length of 32-bit instructions. Three different formats are R-format, I-format and J-format. The RISC has 32-bit general-purpose registers with memory word of 32-bit. The MIPS processor is separated into five stages: instruction fetch, instruction decode, execution, data memory and write back.[35] In this paper an approach is developed which is used to handle & implement the 5-stage pipeline RISC processor precise exception in detail. An exception is precise is a kind of interrupt at the any stage at which the saved processor state corresponds with the sequential model of program execution where one instruction completes before the next begins. [36] In this paper authors have worked on Design and Implementation of 5 Stages Pipelined Architecture in 32 Bit RISC Processor. The design will help to improve the speed of processor, and to give the higher performance of the processor. It has 5 stages of pipeline viz. instruction fetch, instruction decode, instruction execute, memory access and write

back all in one clock cycle. The control unit controls the operations performed in these stages. All the modules in the design are coded in VHDL. Particular attention will be paid to the reduction of clock cycles as well as to improve the speed of processor.[37]-

5.7 DIVA System (Data Intensive Architecture System)

In this paper the authors worked on the increasing gap between processor and memory speeds. It is a well-known problem in computer architecture. As the processor performance increasing at a rate of 50 - 60%per year while memory access times improves only at a rate of 5-7%. However, if multithreading and pre fetching techniques are used to hide memory latency, there will actually be increase in memory band width requirements. So to overcome this problem of mismatching of speed between memory and microprocessor Data-Intensive Architecture (DIVA) system which employs Processing-In-Memory (PIM) chips as smart-memory coprocessors to a microprocessor is implemented. [38]

6. Conclusion

From the literature survey or review it is concluded that for the different types of application a particular type of RISC core with its different-2 configuration is used. The work done on different type of RISC Cores is either based on super pipelining (5 stage pipelining with same hardware) or on super scalar (3, 4, 5-stage pipelining with the individual hardware for each module). Whereas hyper pipelining is a technique with the increased number of pipelined stages after including the intermediate register. This technique uses the OR (Open RISC) 1200 RISC Core. In this paper the main objective was around the speed. Whereas the future work can done on power and area.

7. References

[1] Reduced instruction set computing - Wikipedia, the free encyclopedia en.wikipedia.org/ wiki/Reduced_instruction_set_computing.
[2] Introduction to RISC Processors by ni logic Pvt.Ltd, pp. 1-42

[3] Tobias Strauch, "Hyper Pipelined OR1200 Core Specification", OpenCores Hyper Pipelined OR1200 Core, pp.-1-19, 2010
[4] Tobias Strauch "Hyper Pipelining of Multicores and SoC Interconnects", pp. 1-16, October 2010

[5] Hyper Pipelined Open RISC OR1200 Core, opencores.com/project, OR1200_hp, (Verilog) Hyper Pipelined AVR Core, opencores.com/project,avr_hp, (VHDL)

[6] Damjan Lampret "Open RISC 1200 IP Core Specification", Preliminary Draft Rev.-0.7, pp.-1-63, 2001

[7] Miroslaw Malek, Pratap Chillakanti, Saghir A. Shaikh and Aman sinha "Graph Modeling of Parallelism in Superscalar Architecture –A case study of HP PA-RISC Microprocessor", IEEE International conference on South east con '96, Bringing Together Education, Science and Technology, **pp. 641**-648, 1996

[8] N. Kranitis, G. Xenoulis, D. Gizopoulos, A. Paschalis and Y. Zorian, "Low-cost software-based self-testing of RISC processor cores", IEEE Conference on Computers and Digital Techniques Vol.-150, Issue 5, pp. - 355-360, 2003
[9] Anjana R & Krunal Gandhi "VHDL Implementation of a MIPS RISC Processor", International Journal of Advanced Research in Computer Science and Software Engineering Vol. 2, Issue No.8, pp. 83-88, August 2012

[10] R. Uma Reasearch scholar "Design and Performance Analysis of 8-bit RISC Processor using Xilinx Tool", International Journal of Engineering Research and Applications (IJERA), Vol. 2, Issue No.2, pp. 53-58, March-April 2012

[11] Cjiras toph Baunihof, Frank Muller, Otto Muller, Manfred Schlettino "A Novel of 32 bit RISC Architecture Unifying RISC and DSP", IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 436 – 439, 1997.

[12] Daniel Martin, Robert E. Owen "A RISC Architecture with Uncompromised Digital Signal Processing and Microcontroller Operation", IEEE International Conference on Acoustics, Speech and Signal Processing, pp.-3097-3100, 1998

[13] Marc E. Campbell, Northrop Gmmman Corporation "Evaluating ASIC, DSP, and RISC Architectures for Embedded Applications", 1st merged IEEE International Conference on Parallel Processing Symposium & Distributed Processing, pp-600-603, 1998

[14] Wookyeong Jeong, Sangiun An. Moongyung Kim, Sangkyong Heo, Youngun Kim, Sangook Moon, Yongsurk Lee "Design of a combined processor containing a 32-bit RISC Microprocessor and a 16-bit fixed-point on a chip", IEEE International Conference on VLSI and CAD, pp. 615 - 620, 1999

[15] Qian Gang, Li li, Shen Xubang, Xu Qi, Zhao Ning "The Design of a 32-bit Floating-Point RISC Microprocessor", IEEE International Conference on ASIC (Application Specific Integrated Circuits) , **pp.**760 – 764, 2001

[16] Tay-jyi lin, Shin-kai Chen, Yu-ting Kuo and Chih-wei Liu, "Design and Implementation of a High-Performance and Complexity-Effective VLIW DSP for Multimedia Applications", Springer Journal on Signal Processing Systems, pp. - 209-223, 2008

[17] Daniel Kroening, Wolfgang J. Paul "Automated Pipeline Design", IEEE International Conference on Design Automation, pp. 810 – 815, 2001

[18] Dong-Sun Kim1, Hyun-Sik Kim1, and Duck-Jin Chung, "Implementation of a Neural Network Processor Based on RISC Architecture for Various Signal Processing Applications",Springer Journal on Advances In Neural Networks, pp.-1340-1349, 2006

[19] Pravin S. Mane, Indra Gupta, M. K. Vasantha "Implementation of RISC Processor on FPGA", IEEE Conference on Industrial Technology, pp.-2096-2100, 2006

[20] Xiaoyong Chen and Douglas L. Maskell "M2E: A Multiple-Input, Multiple-Output Function Extension for RISC-Based Extensible Processors", Springer Journal on Architecture of Computer Systems, pp.-191-201, 2006

[21] XIAO Zhi-bin, LIU Peng, YAO Ying-biao, YAO Qing-dong "Optimizing pipeline for a RISC processor with multimedia extension ISA", Springer Journal of Zhejiang University SCIENCE A, pp. 269-274, 2006

[22] Xiaoping Huang,Xiaoya Fan, Shengbing Zhang, "Design and Performance Analysis of One 32-bit Dual Issue RISC Processor for Embedded Application", IEEE 9th International Conference on Solid State and Integrated Circuit Technology, pp.- 978-981, 2008

[23] Rohit Sharma, Vivek Kumar Sehgal, Nitin Nitin, Pranav Bhasker, Ishita Verma "Design and Implementation of a 64-bit RISC Processor using VHDL", IEEE 11th International Conference on Computer Modelling and simulation, pp. 568-571, 2009

[24]Cheol-Hong Moon and Woo-Chun Jang, "Implementation of LED Array Color Temperature Controlled Lighting System Using RISC IP Core", Springer Journal on Emerging Intelligent Computing Technology and Applications, pp. - 753-761, 2009

[25] Y.B.N.V. Bhaskar, D. Rajesh Setty, Addanki Purna Ramesh "Design and Implementation of 64 Bit RISC Processor Using System on Chip (SOC)", International Journal of Computer Science & Commn. Network, Vol.-2, Issue No.1, pp. 1-11, 2011

[26] Vandana S. Shah, Dr. R. V. Kshirsagar "Study of 32-bit RISC Processor Architecture and VHDL FPGA Implementation 32-Bit Matrix Manipulation", International Journal of Computer Applications, Vol. 32, Issue No.3, pp. 50-55, October 2011

[27] Anuruddh Sharma, Mukti Awad "8-Bit RISC Processor using Harvard Architecture", International Journal of Advanced Research in Computer Engineering & Technology, Vol. 1, Issue No.5, pp. 33-37, July 2012

[28] Viral Makwana,Vandana Shah, Roma Patel "32*32 Bit Matrix Coprocessor Design using VHDL FPGA Device for Advanced RISC Machine Operations", International Journal of Advanced Engineering Technology , Vol.3, Issue No.1, pp. 25-28, January-March 2012

[29] Xia Li', Longwei Ji, Bo Shen,Wenhong Li, Qianling Zhang "VLSI Implementation of a High-Performance 32-bit RISC Microprocessor", IEEE International conference on Communications, Circuits and Systems, pp. 1458 – 1461, 2002

[30] Xu Ke, Zhu Kejia, Li Qinng, MIN Hao "The Architecture Comparison and the VLSI Implementation of the 32-Bit Embedded RISC", IEEE International Conference on ASIC, (Application Specific Integrated Circuits), pp. 452 – 455, 2003

[31] Baiying Lei, Wenguang Jin, Huifang Zhang, Kailing Hu, "Optimization and Implementation of AVS-M Decoder on ARM", IEEE 4th International Conference on Image and Graphics, pp.-255-258, 2007

[32] Christiensen C. Arandilla, Joseph Bernard A.Constantino, Alvin Oliver M. Glova "High-Level Implementation of the 5-Stage Pipelined ARM9TDMCore", IEEE International Conference on Communication, Networking, Computing & Industry Applications, pp. 1696-1700, 2010

[33] Wenjiang Li, Song Zhang, Xiong Jiang, Yaohui Zhang "A 5-Stage Pipelined Embedded Processor with Optimized Handling Exception", IEEE International Conference on Computer Science and Network Technology, pp. 2773-2777, December 2011

[34] Viral Makwana,Vandana Shah, Roma Patel "32*32 Bit Matrix Coprocessor Design using VHDL FPGA Device for Advanced RISC Machine Operations", International Journal of Advanced Engineering Technology , Vol.3, Issue No.1, pp. 25-28, January-March 2012

[35]Mamun Bin Ibne Reaz, Md. Shabiul Islam, Mohd. S. Sulaiman, "A Single Clock Cycle MIPS RISC Processor Design using VHDL", IEEE International Conference on Semiconductor Electronics, pp.- 199-203, 2002

[36] Liu Zhenyul, Qi Jiayuez "Implementation of Precise Exception in a 5-stage Pipeline Embedded Processor", IEEE International Conference on ASIC, Vol.-1, pp. 447 – 451, 2003

[37] Sharda P. Katke, G.P. Jain "Design and Implementation of 5 Stages Pipelined Architecture in 32 Bit RISC Processor", International Journal of Emerging Technology and Advanced Engineering , Vol. 2, Issue No.4, pp. 340-346, April 2012.

[38] Draper J, Sondeen J, Mediratta S, Ihn Kim "Implementation of a 32-bit RISC Processor for the Data-Intensive Architecture Processing-In-Memory Chip", IEEE International Conference on Application specific systems, Architecture and Processor, pp. 163 – 172, 2002