THE INTERNATIONAL JOURNAL OF SCIENCE & TECHNOLEDGE

A 64-Bit RISC Processor Design and Implementation Using VHDL

Andrew Obida Yoila
Teacher, Government Senior Secondary School, Minjibir Kano State, Nigeria

Abstract:
In this paper design and implementation of 64-bit RISC processor using VHDL is presented. The main idea is its history; architecture design, its data part, and how the instruction set are introduced. The structure of the design control using VHDL which means Vhsic Hardware Descriptive Language an industry standard language used to describe hardware from the abstract to concrete level and it's verified on the XILINX ISE simulator. This processor is design for a specific application anywhere automation and control is necessary such as in bottle industries and control robotic program, Automatic teller machines (ATM) machines etc. also the 64 bit can also link vast amount of memory almost up to 16 Exabyte.

1. Introduction

In today technology digital hardware plays a very important role in field of electronic and computer engineering products today. Due to fast growing and competition in the technological world and rapid rise of transistor demand and speediness of joined circuits and steeps declines of the price cause by the improvement in micro-electronics application Machineries. The introduction of computer to the society has affected so many things in the society in which almost all problems can be solve using computers. Many industries today are requesting for system developers that have the skills and technical knowhow of designing the program logics. VHDL is one of the most popular design applications used by designer to implement such task. Reduce instruction set computing (RISC) processor play a vital role with RISC AND BIST features which most dominants patterns can provide, in systems testing of the circuits below the tests which is important to the quality component of testing [1]. Although the Reduced instruction set have few instructions sets, as its bit's processing's sizes increase then the test's patterns become denser and the structure's faults is kept great. In view to enable the Operation of the most instructions as registers to registers operation, Arithmetic logic unit is studied and a detail test patterns is being develop. This report is prepaid keeping in mind where specific application is automated and controlled. This report has 33 instruction set with MICA architecture. This report will focus mainly on the meaning of

i. RISC processor,
ii. the design,
iii. the architecture,
iv. the data part and the instruction set of the design.
v. VHDL.

2. RISC Processor

RISC simple know as Reduces instructions set's computer is a micro-processors design to carry out small numbers of system instruction in order to perform in high rate. Because the instruction's types a system should performs need more transistor and circuitry. Although RICS has simpler instruction sets than CISC processors, by the mid 90's many of those RISC processor where considerable more complex than some of the CISC the replace [5]. As technology advances in chips fabrications, technology permit inventors to put together layers upon layers to the micro-processor main core. Intel's 8086 family explain the movement in a particular way, since intel acquired the unique 16-bit processors and put extra feature into each continual generation. With that, it gives way to more cumbersome architecture with incompetent instructions set, but then requires much commercials advantages that end user don't have to pay for new software when they buy the latest reincarnation of a micro-processors. RISC conception brought a considerate scheme of the micro-processor and how good an instruction can be point out to the clocks swiftness of the micro-processor, how modest an architecture is needed and how much works can be done by the micro-chip itself without resorting to soft-ware help [1].RISC processor simply know as Reduce Instructions Set Computing's. It's a set of micro-processor architecture's that manages minor, vastly improved sets of instruction instead of other particular sets of instruction frequently set up in other forms of architecture. This reduce instruction set chip is faster than its CISC counterpart and its design and its build more economically. Firstly, reduce instruction set project was introduced by IBM, Stanford, and UC-Berkeley in the year 1970 and beginning of 80s. The IBM 801, Stanford MIPS and Berkeley RISC 1 and RISC 2 was designs through related view point that is known as reduce instruction set [1].

3. Characteristic of RISC

1. Single cycle's executions time, which RISC processor has a clocked per instruction's (CPI) of single sequence which is due for optimization of each instruction on the micro-processor.
2. System call pipelining agree to real-time performance parts.
3. also the large number of register usually joins a huge amount of registers to avert in great numbers of dealings with memory.
4. Instructions which change the movement of controls such as branch's instruction which are executed well since they compress about 20 to 30% of distinctive programs.
5. RISC processor don't attempt to implement infrequent used instructions.

4. Design

There are so many RISC processors but we are going to discuss about the design considering the MICA architecture and its implementation using VHDL.MICA is known as (minimal integrated computer architecture.) is one of the design that have a simple RISC processor architecture features. ALU architecture design consist of two parts which are,

1: The operational architectures that performs the action of the ALU.

2: The testing architecture that plays first in testing.

Operational architectures perform the realaction of the ALU which Stands for Arithmetic and Logic Unit is a central component of nearly all computer machine and CPUs since the origination of early computers. Which lead to the design and implementations of 64 bit Reduce Instruction Sets processors built on double reformed ALU architecture. The ALU architecture is separated based on what consumer's need, the cost, speed inessalso the power intake. Firstly, design ALU is not a very expense's design, power consumption is less and complex. The second design is a high speed, low power model using Carry Look-Ahead Adders (CLAs) and Vedic multiplier [2]. Extensive parallelism is seen in them. While the third design is mostly used for low power and compactness and the uses of Pre-fix adders and Booth multiplier. The design of the RISC processors also involves the design or use of memory and development of opcodes, which are also included in this report. ALU has five units, that are 4bit carried look's Ahead adder (CLA), A 4 bit AND, OR, XOR, and an Inverters gate. And also a PreCLA that will organize the input based on the arithmetic operations that will be perform. Multiplexer (MUX) that make use of selects pin to select one of the result as of the above five units. MIPS processor is an example of Early RISC architecture which we are going to use to fully understand the features and design of RISC architecture. MIPS processors executed a smaller, simple instructions sets where by each of the instruction set built-in in the chips design ran into a single clocks circles. The processors use system called pipelining to more efficiently process instruction.

5. Instructions Set of MIPS

The mips64 bit architectures are built on fix size mostly programmed instructions sets and it uses load/store data models. Meant to supports optimize executions of high level language. Arithmetic and logic operation uses a three operands formats, permitting compiler to optimized complex's expression invention. The architectures drive the advantaged mode exceptions handlings and memory's management's function from the R4000 and R5000 class processors [4]. A set of the registers reflects the configuration the caches. Memory Management Unit (MMU), Translation Look aside buffer (TLB), and other privileged features implemented in each core. The MIPS64 architectures allow real-time operations and applications codes to be executed once and re-used with futures member of both the MIPS32 and MIPS64 Processors family [2].

6. MIPS Instruction Set Architecture

The MIPS is another type of RISC architecture that the data bus size is 32bits, with the address bus is 32 too. It also supports 4-integer data type of 8-bit byte, 16-bit byte, 32-bit byte and 64-bit double words and also MIPS provide 32 general purpose 32-bit register with a floating point that is supported with 32 –bit single precision and 64-bit double word precision values [4].

MIPS performs the following operations such as load/store architectures: which data has to be loaded from memory into registers first before it can be manipulated. Results are stored into registers, and they have to be stored out to memory if required.

MIP perform this instruction operation which loads registers with values either from the RAM and store register values out to RAM location does basic arithmetic logic such as ADD,

SUBTRACK, MULTIPLY, DIVIDE with Remainder. And does basic floating-point arithmetic: add, subtract, multiply, divide. Logical operations: AND, OR, NOR, exclusive OR (XOR) and shift operations: shift left, shift right. Comparison operations: ==, !=, <, >, <=, >=. Mips has three instructional format which are R-type instructions that perform arithmetic and logic operation on register, I-type instruction that does load/store operation and

J-instruction that perform jumps and function calls.

Figure 1: MIPS Processor RICS Architecture

RISC is given a special side in the computer architecture that was established due to the result that lead to 801 tasks that began in 1975 at the IBM T.J. Watson Research Center that remained accomplished at the early 1980s. Developments of RISC architecture came to past due to the analysis of in what way the instruction is really used at the actual program. RISC architectures start through a minor set of often used instruction that regulates the pipeline structures of the machines allowing fast performance of those instruction in single cycles. If adding a new simple instruction rises the "serious path" (normally 12-18 gates stages) for one access level, then the new instructions would give at most 6-8% to the overall performances of the machines [5]. Single set per instructions is realized by exploitations of parallelism through the use of pipelining. There are so many RISC type which varies by company such as those scalable processors architectures. MIPS outgrow of Stanford MIPS project, used by Silicon Graphics, and a superscalar implementations of RISC architectures, IBM RS/6000 (PowerPC architectures) [4].

Reduce instruction sets performances is necessary for every computer model and architecture which lead to every fresh architectures or scheme organizations. Each instructions of the RISC machines that is perform is simple and straightforward. With this, the period necessary to implement each instruction should be shortened and the numbers of cycles decrease. Usually the instruction simple mentation time is shared in five stages, machines cycle, and as soon as execution of one phase is finish, machines proceed with implementing the instant stage [4]. Moreover, if the phase become free it is used to execute the same operations that belong to the next instructions. Operations of the instruction is done in a pipeline's fashions that is related to the meeting line in the factory's procedure. Usually this five pipeline phases include: IF-Instructions fetch

ID-Instructions Decodes

EX-Executes

MA-Memory Access

WB-Write Back

Main idea of RISC is to archive the execution rate of one cycle per instructions that would be the case when no interruption in the pipeline occurs.

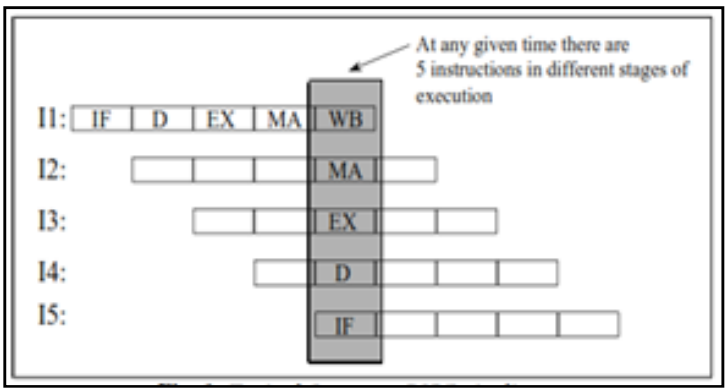

Figure 2: Typical Five Stage RISC Pipeline

At any time, there are five instructional in different stages of execution. That means each execution of instruction finished before the next. The key features of RISC are the architectures' maintenance aimed at the utilization of parallelism on the instructions' level. Hence all well-known feature of RISC architectures could be considerer's in graceful of their supports for the RISC pipeline. Adding to that, RISC takes benefit of the principles of locality: spatial and temporal. What that means is that the data that was used recently is more likely to be used again [3]. This justify the implementations of a comparatively large general purpose register file found in RISC machine as opposed to CISC. Spatial section means that the data most likely to be referenced is in the neighbourhoods of a location that has been reference. It is not clearly specified, but that imply the use of cache in RISC.

7. What Is VHDL

VHDL (high speed integrated circuit) came into light in early 1980 as a program high speed integrated circuit research
It was founded by the U.S department of defense. It is a programming language that was design to enhanced, describe the behavior of digital systems. Which also the project is executed via VHDL and it's tested on Xilinx ISE. VHDL is an overall drive programming languages like high-level programming languages that allowed hard designed concept to be uttered as computers program, VHDL permits the conduct of multipart electronics circuit to be taken into a project system for automatics circuits' fusion or for system simulations. E.g. Pascal, C and C++, VHDL include feature necessary for structures design technique, and gives a rich set of control and information representations feature. Unlike these other programming languages, VHDL provides feature that allowed simultaneous event to be describes. This is necessary since they hardwire describes using VHDL is inherently concur rents in its operations [7].

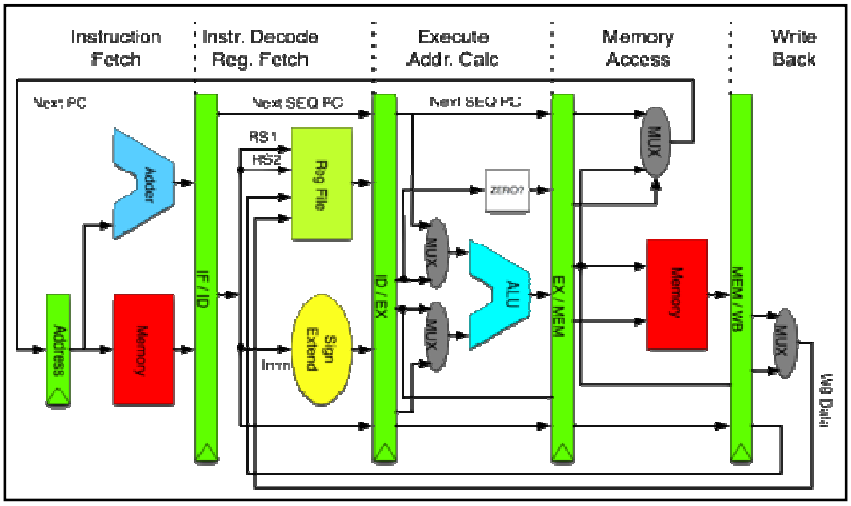

Figure 3: MIPS RISC processor architecture

Looking at the figure, procedure movevia the processor's that execute an instruction that is actually fairly big. There are four phases which are:
1; Fetch – succeeding instructions is being read by the CPU after the stack of instruction which is reserved in memory.
2; Decode- the CPU decode instructions and it figures out what to do with it. By the way, decode is the place in which the main change among complex instruction set (CISC) and reduce instruction set initially come to played. On a nutshell CISC instructions sets example x86, which an instruction can really be a micro-instruction that a numbers of micro-operations are really fixed. Example "MOV'' in assemblers. MOV change information. A "mov" that means copy is fellow by a delete. RISC instruction set break down job/ task into smaller unit and much of them are implemented. Which is the facts that RICS compilers, a program software that turns high level programming language into executables machines codes that are more complex and RISC executable where often large. E.g. BASIC or C language compilers.
3; Execute –in execute the CPU perform the next execution of the instruction. In today world super CPU, which include multiples "performance unit". Super scalar originated from the RISC word in which modest instruction permit simple performance unit that might be scales out allowing for two instructions to be implemented at the same time. Usual the execution unit are "integer unit" (that handle integer math). And "floating's" that handles, shockers, floating point math. Due to the development of super-scalar computer architectures, that pipelining start to gain fancy and a single processor can workout magic when executing two instructions at the same time. Before an instruction is finish executing it will pick up another one to be executed at the same time. Which condition has to be made that has to free an execution unit before executing next in order to make the process well-organized and less dangerous. Today's processor can execute out of order e.g. by taking instruction a little down rather than the next one and pick these instructions based on which instructions can execute together via algorithms like "branch prediction" the logic is that if the outcome of instructions A might impact what instructions B does, then it cannot be executed at the same time. But if instructions C or D could be next executed after B depending on the outcome of both A and B, the CPU logically will take either C or D to execute in parallel with A based on "branched predictions". When it is wrong it takes time and has to do it over again. If it's right, it gains some big efficiency.
4; Store – after the instructions has finished executing, the outcome is store inside the data space of memory

8. Conclusions

This report focus on A 64-bit RISC processor with 33 instruction set has been written. Every instruction is executed in two clock cycles. RISC has been analyzed to give details on how it operates and it is design, Arithmetic logic unit was also analyzed and a complete sets of test pattern was established. VHDL is being explain as the programming language that is use to verified the design and implementation of the RISC. Prospect works will be included in the increasing numbers of instruction and make a pipeline design with less clock cycles per instructions and more improvements can be included in the future's works.

9. Index

- RISC - Reduce instruction sets.
- MICA - Micro architecture-independent characterization of applications.
- VHDL - Vhisic hardware Descriptive Language.
- ALU - Arithmetic logic unit.
- MIPS - Microprocessor without interlocked pipeline stage.
- BIST - Built-in self-test.

10. References

i. D. Bhandarkar and D.W. Clark, "Performance from Architecture: Comparing a RISC and a CISC with Similar Hardware Organization," Proceedings of the 4th Int'l. Conference on ASPLOS, Santa Clara, California, April 8-11, 1991.

ii. Yen-Kunang Chen and Sun-Yuan Kung, "Trends and Challenges with System-on-chip Technology for Multimedia System Design", Emerging Information Technology Conference 2005, 15-16 August 2005.

iii. Steve Brazier, Hrvojecekoli and marcus Gillette "RISC definition" September 2005.Steve Brazier, HrvojeCekolj, and Marcus Gillette

iv. http://www.imgtec.com/mips/mips64 architecture.asp

v. http://www.answers.com/topic/risc

vi. http://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/mips/index.html

vii. http://techpubs.sgi.com/library/dynaweb_docs/0530/SGI_Developer/books/DevDriver_PG/sgi_html/ch01.html#id87076