ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 40-44 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

DESIGN OF A SYNTHESIZABLE PIPELINED MIPS CORE FOR EMBEDDED APPLICATIONS

Venkateswarlu Kota[1], D.Tirumala Rao[2]
[1]Department of ECE, GMR Institute of Technology, Rajam, India, Venky084c6@gmail.com
[2]Department of ECE,GMR Institute of Technology, Rajam, India, tirumalarao.d@gmrit.org

Abstract

The use of soft core processors has become inevitable in electronic and mechanical industries for embedded applications. A soft-core processor is a model of hardware description language (HDL) of a specific processor which can be customized for a given application and synthesized for Application Specific Integrated Circuits (ASIC)/FPGA. Usually they contain embedded processors that are often in the form of soft-core processors that execute software code. In this paper soft-core reconfigurable pipelined MIPS processor is developed, which is used in all embedded on-chip applications. Specifically, synthesizable VHDL model for un-pipelined and pipelined MIPS Processor is simulated. This design is useful for all synthesized to verify the synthesizability. Modelsim is used for logical verification and synthesizing is carried out using Xilinx-ISE14.2 tool. The results prove the proper functioning of the processor and found to occupy less area compared to previous one.

Key Words: Low Power, Speed,MIPS Processor,Pipelining Mechanism,ASIC, Modelsim, VHDL, Xilinx Tool.

INTRODUCTION

Soft-core processor is a model of hardware description language (HDL) of a specific processor and has found to be more advantageous than the hardcore processor. Soft-coreprocessor is platform independent, immune to obsolescence, flexible and of lower cost [1-2].

Microprocessor without hardware Interlock Processing System (MIPS) Technologies is a computer manufacturer that has designed and sold several RISC microprocessors starting with the MIPS processors [3]. The key point in the Reduced Instruction Set Computing (RISC)philosophy has been the emphasis on simplicity: having only simple basic operations, simplifying instruction formats, reducing the number of addressing modes, and eliminating complex operations. This computing paradigm could have been called Simple Instruction Set Computing (SISC). So, the design of reconfigurable MIPS processor is an interesting problem of research.

XizhiLi,Tiecai Li[4] have proposed ECOMIPS which focuses on economic resource utilization on modern chips. Gautham P, Parthasarathy R, KarthiBalasubramanian[5] have proposed design and implementation of a low power five-stage parallel pipelined structure of a MIPS 32 compatible CPU. Jong Hyuk Lee, SeungEun Lee, Heon Chang Yu, and TaeweonSuh[6] have proposed a pipelined CPU design project with a field programmable gate array (FPGA) system in a computer architecture course.

In this paper soft-core reconfigurable pipelined MIPS processor is developed, which can be used in all embedded on-chip applications. The components Register file, memory, ALU and processor CPU are built by using VHDL model. These components are integrated by using port-map statements. The processor module and the memory are integrated to yield the complete pipelined MIPS processor. Later they are simulated and synthesized. All simulations are carried out using VHDL.

The rest of the paper is organized as follows. In section II, this gives a few hints on standard MIPS processor. In Section III, Pipelined MIPS Processor (Proposed) and its Implementationis explained and simulation results, conclusions are presented in Section IV

MIPS PROCESSOR

Overall Data Path of standard MIPS processor is shown in Fig.1. It integrates the fetch and execute hardware and adds other required elements for correct operation.

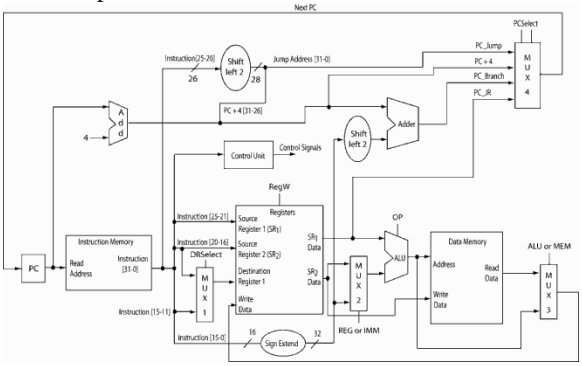

Fig.1. Overall Data Path of standard MIPS processor

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 40-44 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

The basic MIPS processor was only in the hardcore chip form and was not suitable for reconfiguration. The individual components need to be assembled with other peripherals. First, PCB is to be built with basic MIPS processor & others. But this led to all increase in, size, area, power and other complexities. That's to all the custom microprocessors are un-pipelined. Hence it cannot be operate in real time environment.

In order to overcome this, proposed MIPS processor is developed by soft-core. There is the advantage of soft-core that can be integrated with peripherals in software. The primary advantage of soft-core is integration of both ASIC and CPU into a single chip. This type of chip will reduce the power, area and also increase the systems speed, performance and reliability.Hence real time applications can be run on pipelined MIPS processor. The pipelined MIPS processor can also be optionally included with other peripherals.

PIPELINED MIPS PROCESSOR (PROPOSED) AND IT'S IMPLEMENTATION

Pipelining is used to reduce the delay between completed instructions (called 'throughput').However, implementation of pipelining results instructural hazards, data hazards, and control hazards.

Patterson and Hennessey [7] give the complete information of pipelining and pipelined data path of processor. The pipelined data path of the processor is givenin Fig. 2 where the different stages of the pipeline are separated by pipeline registers. These registers are used tocarry-over results from the previous to the following stages. In this paper, discussion is made to eliminate structural hazards, data hazards which appear in the diagram shown in Fig.2.

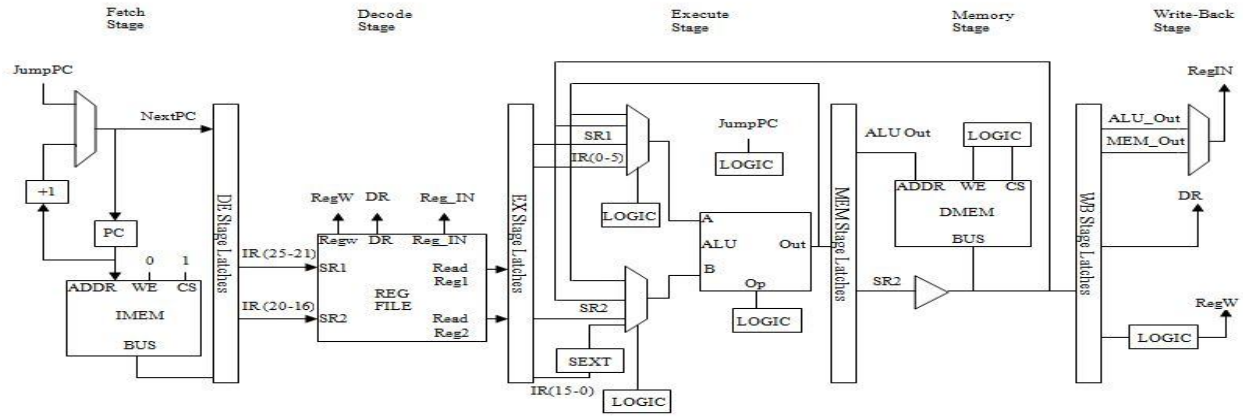

Fig .2.shows overall pipelined data path

Structural Hazards

It means that the hardware cannot support the combination of instructions that are set to execute in the same clock cycle. In MIPS pipeline with a single memory Load/store requires data access, instruction fetch would have to stall for that cycle that would cause a pipeline "bubble". These Hazards can always be resolved by waiting.Hence, pipelined data-path requires separate instruction/data memories. Fig. 3 illustrates the memory-use hazard situation [7].

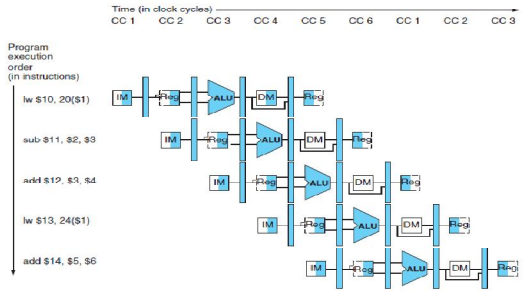

Fig. 3. Separate instruction/data memories for a 'memory-use' hazard (taken from [7])

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 40-44 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

Data Hazards

Data hazards arise from the dependence of one instruction on an earlier one in the pipeline. To avoid these hazards, data needs to be forwarded which is done by the forwarding unit. Data forwarding cannot prevent all pipeline stalls. When a data requested by a load instruction has not yet become available it leads to load-use hazards. To resolve this hazard, the pipeline is stalled by the unit for one stage and then continues with the forwarding of data. Fig. 4 illustrates the load-use hazard situation [7].

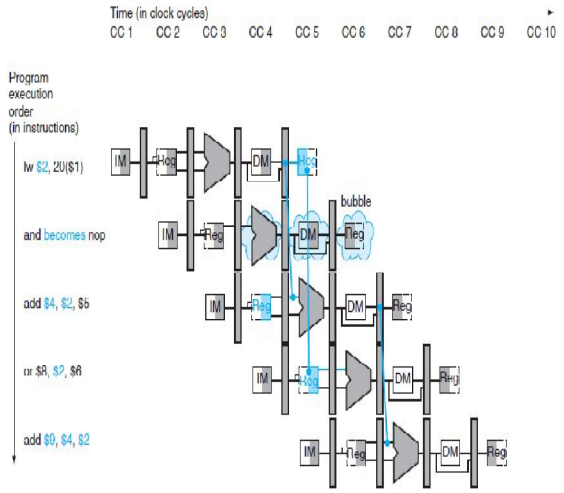

Fig. 4. Data forwarding with a stalled pipeline stage for a 'load-use' hazard (taken from [7])

SIMULATION RESULTS

The overall pipelined data path of MIPS processor shown in Fig.1 has been modeled using VHDL [8, 9-10]. If all synthesizable VHDL code of standard MIPS processor runs correctly, the simulation results obtained as shown in Fig.5.

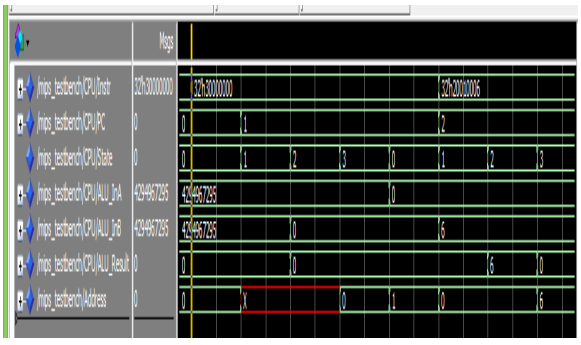

Fig. 5. Shows the simulation results for standard MIPS processor

The following command file was used to test the VHDL model of standard MIPS processor. The configure list -delta collapse command removes outputs at intermediate deltas.

```
add  list -hex sim:/mips_testbench/cpu/instr
add  list -unsigned sim:/mips_testbench/cpu/nPc
add  list -unsigned sim:/mips_testbench/cpu/pc
add  list -unsigned sim:/mips_testbeflch/cpu/state
add  list -unsigned sim:/mips_testbench/cpu/alu_.ina
add  list -unsigned sim:/mips_testbeflch/cpu/alu_inb
add  list -signed sim : /mi ps_testbench/cpu/alu_result
add list -signed sim:/mips_testbench/cpu/addr
configure list -delta collapse
run 2330
```

The simulation results of standard MIPS processor in the form of wave list are tabulated in Table.1. Simulation wave list was done using modelsim, which shows the details of each and every instruction which is written in the test bench.

Fig .6.Shows the simulation results for the proposed pipelined MIPS processor.If all programs are run correctly then output is observed on the data memory bus (DMEM_BUS) like (24, 12, 2...etc) as mentioned in test bench.

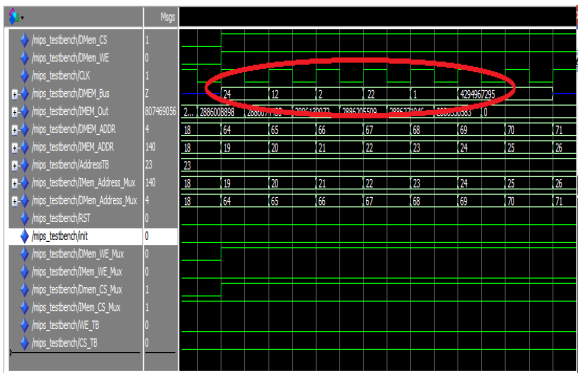

Fig. 6. Shows the simulation results for our pipelined MIPS processor

CONCLUSION

This paper presents analysis &synthesis of soft-core reconfigurable pipelined MIPS processor, which can be used in all embedded on-chip applications.The subcomponents such as Register file, memory, ALU and processor CPU are built by using VHDL model. These components are integrated by using port-map statements. The processor module and the memory are integrated to yield the complete pipelined MIPS processor.The processor was successfully designed in VHDL, simulated with Modelsim and synthesized on

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 40-44 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

to a Xilinx-ISE14.2 tool. From the results shown in Fig.5, Table.1,it is evident that the standard MIPS processor is working correctly and the proposed MIPS processor is working correctlyfrom the results shown in Fig.6. It can be further revealed that by properly designing the data path, the hazards are automatically eliminated in the proposed system. The proposed system occupies less area and consumes lesser power compared to standard MIPS processor. The results prove the efficiency of the proposed processor, and makes unavoidable in many of the real-time applications such as microprocessor, microcontroller which requires lesser time for programmed execution.

REFERENCES

[1]. Parhami, Behrooz. Computer Arithmetic: Algorithms and Hardware Design. New York: Oxford University Press, 2000.
[2]. Patterson, David A., and Hennessey, John L. Computer architecture: Quantitative approach, 4rd Ed, Elsevier, San Francisco 2007.
[3]. Kane, Gerry. MIPS RISC Architecture. Upper Saddle River, N.J.: Prentice Hall,1989.
[4]. Xizhi Li and Tiecai Li, "ECOMIPS: An Economic MIPS CPU Design on FPGA,"IEEE International Workshop on System-on-Chip for Real-Time Applications, (IWSOC'04),2004.
[5]. Gautham P, Parthasarathy R and KarthiBalasubramanian,"Low-Power Pipelined MIPS Processor Design",ISIC 2009,pp 462-465,2009.
[6]. Jong Hyuk Lee, SeungEun Lee, Heon Chang Yu, and TaeweonSuh, ''Pipelined CPU Design with FPGA in Teaching Computer Architecture'' IEEE Transactions On Education, Vol. 55, No. 3,pp 341-348, August 2012.
[7]. Patterson, David A., and Hennessey, John L. Computer Organization and Design: The Hardware Software Interface, 3rd Ed, Elsevier,San Francisco 2005.
[8]. Brown, Stephen and Vranesic, Zvonko. Fundamentals of Digital Logic with VHDL Design, 2nd Ed, New York: McGraw-Hill,2005.
[9]. Chan, P., and Mourad, S. Digital Design Using Field Programmable Gate Arrays. Upper Saddle River, N.J.: Prentice Hall,1994.
[10]. Chang, K. C. Digital Design and Modeling With VHDL and Synthesis. Los Alamitos, IEEE Computer Society Press, 1997.

Table.1. Simulation results of standard MIPS processor

MIPS Instruction	ns	Instr	PC	State	ALU_InA	ALU_InB	ALU_Result	Addr
andi $0,$0,0	570	30000000	0	0	-	-	0	0
	590	30000000	1	1	-	-	0	X
	610	30000000	1	2	-	0	0	X
	630	30000000	1	3	-	0	0	0
addi $1,$0,6	650	30000000	1	0	0	0	0	1
	670	20010006	2	1	0	6	0	0
	690	20010006	2	2	0	6	6	0
	710	20010006	2	3	0	6	0	6
ori $2,$0,18	730	20010006	2	0	0	6	0	2
	750	34020012	3	1	0	18	0	6
	770	34020012	3	2	0	18	18	6
	790	34020012	3	3	0	18	0	18
add $3,$1,$2	810	34020012	3	0	0	18	0	3
	830	00221820	4	1	6	6176	0	18
	850	00221820	4	2	6	18	24	18
	870	00221820	4	3	6	18	0	24
sub $4,$2,$1	890	00221820	4	0	6	18	0	4
	910	00412022	5	1	18	6	0	24
	930	00412022	5	2	18	6	12	24
	950	00412022	5	3	18	6	0	12
and $5,$1,$2	970	00412022	5	0	18	6	0	5
	990	00222824	6	1	6	18	0	12
	1010	00222824	6	2	6	18	2	12
	1030	00222824	6	3	6	18	0	2
or $6,$1,$2	1050	00222824	6	0	6	18	0	6
	1070	00223025	7	1	6	18	0	2
	1090	00223025	7	2	6	18	22	2
	1110	00223025	7	3	6	18	0	22

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 40-44 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

slt $7,$1,$2	1130	00223025	7	0	6	18	0	7
	1150	0022382A	8	1	6	18	0	22
	1170	0022382A	8	2	6	18	1	22
	1190	0022382A	8	3	6	18	0	1
sll $8,$2,4	1210	0022382A	8	0	6	18	0	8
	1230	00024100	9	1	0	18	0	1
	1250	00024100	9	2	0	18	288	1
	1270	00024100	9	3	0	18	0	288
srl $9,$1,1	1290	00024100	9	0	0	18	0	9
	1310	00014842	10	1	0	6	0	288
	1330	00014842	10	2	0	6	3	288
	1350	00014842	10	3	0	6	0	3
beq $1,$2,1	1370	00014842	10	0	0	6	0	10
	1390	10220001	11	1	6	18	0	3
	1410	10220001	11	2	6	18	-12	3
lw $10,4($0)	1430	10220001	11	0	6	18	0	11
	1450	8C0A0004	12	1	0	-	0	-12
	1470	8C0A0004	12	2	0	4	4	-12
	1490	8C0A0004	12	3	0	4	0	4
	1510	8C0A0004	12	4	0	4	0	4
bne $1,$2,1	1530	8C0A0004	12	0	0	4	0	12
	1550	14620001	13	1	24	1	0	4
	1570	14620001	13	2	24	18	6	4
j 16	1590	14620001	14	0	24	18	6	14
	1610	08000010	15	1	0	0	0	6
jr $2	1630	08000010	16	0	0	0	0	16
	1650	00400008	17	1	18	0	0	6
	1670	00400008	17	2	18	0	0	6
sw $3,64($0)	1690	00400008	18	0	18	0	0	18
	1710	AC030040	19	1	0	64	0	0
	1730	AC030040	19	2	0	64	64	0
	1750	AC030040	19	3	0	64	0	64