Abstract — This paper deals with the design of a low power pipelined RISC processor and its implementation on CPLD. This paper presents the architecture, low power unit, control unit, arithmetic logic unit and instruction set of the 64-bit RISC processor. Design, implementation and debugging are carried on a low-cost, full-featured ADM kit. RISC processor is designed using Verilog HDL. The software tool used for building and testing these modules in the present work is Xilinx ISE 9.1i and simulation results are analyzed with Modelsim software.

Keywords — CPLD, RISC processor, Modelsim, Verilog HDL

I. INTRODUCTION

To minimize the power of RISC processor clock gating technique is used in the architectural level, which is an efficient low power technique. The four stage pipelining includes different blocks namely Fetching, Decoder, Execution and Memory Read/Write blocks and these are implemented in one clock cycle. Here low power RISC processor is designed without any complexity because power reduction is done in front end process. RISC is a microprocessor it has a relatively limited number of instructions and is designed to perform millions of instructions per second. Since the instructions are so simple, RISC processor can execute their instructions very fast [1]. RISC chips require fewer transistors, which make them cheaper to design and produce. In an RISC processor, more complex instructions can be composed with its simple instruction set and basic instructions. Instructions are executed in one machine cycle since each instruction is of the same length so that the processor can handle several instructions at the same time. The pipelining is a key technique used to speed up RISC machines.

To decode the information is the important feature of RISC instruction format. It can execute one instruction per cycle. Overlapping the fetch, decode and execute phases of two or three instructions can be possible by using the pipelining technique. A fixed number of bytes of instructions can take fixed amount of time for execution.

In this processor, most instructions are of uniform length and similar structure. Arithmetic operations are restricted to CPU registers and only separate load & store instructions can access memory.

II. COMPLEX PROGRAMMABLE LOGIC DEVICE

CPLD is a programmable logic device with complexity between that of PALs and FPGAs, and architectural features of both. The predictable timing characteristics of CPLD only can make them ideal for critical, high-performance control applications [2]. CPLD is a combination of a fully programmable AND/OR array and a bank of macrocells [3]. The AND/OR array is reprogrammable and can perform a multitude of logic functions. Macrocell is the building block of a CPLD, which contains logic implementing disjunctive normal form expressions and more specialized logic operations. When compared to FPGAs and other programmable logic devices, CPLDs have a shorter and more predictable delay. CPLDs are used in cost-effective, battery-operated portable applications because they are inexpensive and require relatively small amounts of power [4]. The CPLD device used in the present work is XC9572 manufactured by Xilinx.

A. Details of XC9572

The XC9572 is providing advanced in-system programming and test capabilities for general purpose logic integration [5]. It consists of eight 36V18 Function blocks, providing 1,600 usable gates with propagation delays of 7.5ns. By configuring macrocells to standard or low-power modes of operation power dissipation can be reduced in XC9572. To minimize the power dissipation unused macrocells are turned off.

B. Baseboard details

The ADM Base Board acts as a universal CPLD/FPGA/μP reconfigurable implementation platform to allow rapid and interactive prototype development. The board houses a plug-in daughter board containing a target FPGA/CPLD/μP device.
Daughter boards can be easily swapped to provide support for multi-vendor devices available in DIL, PLCC, TQFP, PQFP packages up to 240 pins [6]. Fig.1 (a) & Fig.1 (b) show the photographs of the CPLD base board and CPLD daughter board used for the present work.

C. Specifications of base board

- 16x2 Alphanumeric LCD with the backlight.
- Four digit 7-segment display & Eight general purpose LEDs.
- Three LEDs to indicate the various power voltages & 4x4 Matrix KeyPad.
- 208 I/O pins, All I/O with VCC/GND links & Onboard 10 MHz oscillator.
- 10MHz clock and four different frequency clocks (5MHz, 1MHz, 500 KHz, 100 KHz), Support’s 1.8V to 5V devices.
- JTAG Programming cable & Parallel/Serial/USB/SPI/CAN
- PS2 Mouse, Keyboard I/F, I2C EEPROM, RTC & Serial ADC/DAC with Pot.
- EM Relay of DC5V, RS232 serial interface through the 9-PIN D-type connector.

D. Daughter Board Details

The Daughter board contains Xilinx CPLD (XC9572), which can be plugged onto the base board. The xc9572 device supports serial configurations, using the master/slave serial and JTAG modes, as well as byte-wide configuration employing the slave parallel mode [7].

Daughter Board includes the following items

- Xilinx CPLD (XC9572)
- Seven 10- pin Headers, intended for use as a GPIO
- Four sets of 20x2 female berg connector for plugging onto the baseboard
- JTAG pins for in-circuit programming
- Power supply 5V, 3.3V, and 2.5V are provided from the baseboard

III. ARCHITECTURE OF 64-BIT RISC PROCESSOR

The low power pipelined 64-bit RISC processor is a single cycle pipelined processor. It has small instruction set, load/store architecture, fixed length coding and hardware decoding and large register set. By using dedicated buses, it will get instructions on a regular basis to its memory and execute all its native instructions in stages with pipelining. External devices can be communicated with its dedicated parallel IO interface [8]. In this design, all the arithmetic, branch, logical operations are performed and the resultant value is stored in the memory/registers and retrieved back from memory when required[9]. The proposed architecture of RISC processor is shown in Fig.2.

The architecture of 64-bit RISC processor consists of four stage pipelining. They are Instruction Fetch, Instruction Decode, Execute, Memory Read/Write Back [10].To obtain an instruction from the instruction memory using the current value of the PC and increment the PC value for the next instruction is the function of instruction fetch unit. Fetching instruction means the instruction present in the memory is fetched from the PC and stored it in the instruction register. Opcode fetched from the memory is being decoded for the next steps and moved to appropriate registers is the function of decode unit. To perform required operation based on the opcode and store the result in the immediate register is the function of the instruction execute unit. To store the result into corresponding register or memory is the purpose of the memory read/write back unit [11].
A. Low power Technique

Different RTL and gate-level design strategies are there for reducing power. In that Clock Gating design is used for reducing dynamic power in the present work. In the clock gating design method, the modules which are working at that instant for them only clock is applied [12]. It is a dynamic power reduction method in which the clock signals are stopped for selected register banks during the time when the stored logic values are not changing [13]. One possible implementation of clock gating is shown in Fig.3. The clock pulse for low power technique is shown in Fig.4. The input to low power unit is a global clock and its output is gated clock since the module will block the main clock in the following conditions.

- When an instruction is a halt.
- When there is a continuous Nop operation.
- When program counter fails to increment.

IV. SIMULATION RESULTS

By using Modelsim, the simulation results have been verified. The Fig.5 shows simulation results of the low power unit. The Fig.6 shows the simulation results of instruction fetch unit. The Fig.7 shows the simulation results of instruction decode unit. The Fig.8 shows the simulation results of the execution unit. The flow chart of RISC processor is shown in Fig.9.
V. CONCLUSION

CPLD based low power pipelined 64-bit RISC processor is designed. The modules: Instruction fetch unit, Decode unit, Execution unit and Low power unit are implemented on XC9572. Therefore, the arithmetic operations, branch operations, and logical functions are verified. With a single instruction the proposed architecture is able to prevent pipelining to multiple executions. By using low power technique, CLOCK is switched off through sleep mode when the processor is idle. For low power applications to enhance the battery life of the devices this design can be preferable.

This architecture can prevent pipelining from flushing when branch instruction occurs and can also provide halt support.

REFERENCES

[8] [9] http://elearning.vtu.ac.in/12/enotes/Adv_Com_Arch/Pipeline/Unit2-KGM.pdf