Low Power MIPS Processor Design

Yang Zheng, Taru Doodi, Lalitha Geddapu, Qinkan Kang, Yixue Zhang
Department of Electrical & Computer Engineering
University of Florida, Gainesville, FL, USA

***Abstract*— This paper presents a power-optimized MIPS architecture that uses three techniques – clock gating, data gating and multiple-voltage supply (MVS) and compare the performance of these three techniques with our basic MIPS design. We implemented the design based on Verilog code, obtained simulations, performed synthesis, placement and routing, and finally generated a layout. The layout has been tested for DRC and LVS. The simulations and results for the MIPS architecture have been presented.**

***Index Terms*—MIPS, MVS, Clock gating, Data gating, Level shifter, Power, Area.**

I. Introduction

While the performance of processors reach a limit due to transistor technology, the focus has shifted from attaining more performance to reduced power consumption of existing processor technology. As the technology scales lower, the power dissipation due to leakage current becomes more significant. Thus, in order to reduce power consumption, the leakage currents need to be reduced. The demand for low power design became more significant with the emergence of portable systems like laptops, mobiles and tablets that run on battery supply. Emphasis shifted from speed to reduced power consumption as the users of portable systems demand more stand-by time and longer life for the battery. Moreover, low power designs help in reducing packaging and cooling costs, noise immunity and in addressing environmental concerns. In this paper, we optimize the MIPS architecture using various power optimization techniques. A lot of prior work has been done in low power CMOS design by Chandrakasan et. al[1]. Calhoun[2] worked on reducing leakage current by exploring MTCMOS techniques. Zhai et. al. have worked on reducing power consumption by utilizing sub-threshold techniques.

MIPS is an acronym for Million Instructions Per Second. The MIPS architecture is the baseline for many architecture implementations. MIPS architecture has a load-store instruction set architecture. It is designed for pipelining efficiency. Its architecture is easy to understand making it a good model to study. The registers in MIPS architecture are 32-bit wide. There are 32 such General Purpose Registers (GPRs) also called as integer registers. In addition to these integer registers there are 32 floating point registers.

In our implementation we mainly focus on low power optimization, so we use simplified MIPS architecture as our baseline. It is an 8-bit subset of the MIPS architecture that uses an 8-bit datapath, 8 registers ($0 - $7) and an 8-bit program counter (PC). The MIPS architecture can be divided into 3 blocks –datapath, controller and ALUcontroller. This block level design helps us in determining the wiring between modules, providing a top-level module for writing the system Verilog code to simulate and synthesize the architecture. The delay of these three modules can be determined to identify the critical path of the circuit.

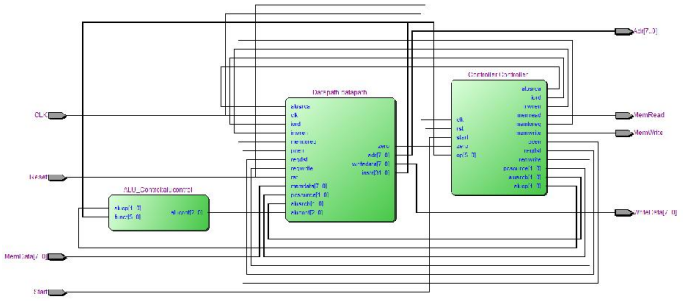

Figure 1: MIPS RTL Schematic

We have optimized this MIPS architecture for power using three techniques – clock gating, data gating and multiple supply voltages (MVS) [7]. Due to division of the architecture among different modules and different supply voltages, we would require a voltage shifter in places where low voltage domain is driving a high voltage domain. If voltage shifter is not used at voltage domain crossings, it leads to increased leakage problem.

Further sections discuss the power optimization techniques in detail. Section 2 discusses the design methodology and implementation. Section 3 discusses and analyses the results obtained from the implementation. Section 4 provides conclusion to the project.

II. Design Methodology

In this low-power design, we have realized MVS, clock and data gating in the MIPS architecture. However, there are many effective ways to maintain performance with reduced power, for example, parallelism and pipelining. With these implementations, power can be reduced by 50% or more, however, significant area overhead due to additional registers and paralleled logic is incurred.

Power gating is another technique that is used to reduce leakage currents by inserting a switch transistor (usually high threshold voltage) into the logic stack (usually low threshold voltage), which is most effective for systems with standby operational modes. It is possible to achieve 1 to 3 order reduction in leakage, but switches add complications and the tradeoffs between area, performance and leakage have to be considered.

A. Multi-voltage Supply Design

MVS techniques operate different blocks at different voltages. Running at a lower voltage reduces power consumption, but at the expense of speed. Designers use different supply voltages for different parts of the chip based on their performance requirements. MVS implementation is key to reducing power since lowering the voltage has a squared effect on active power consumption [6].

We have written a CPF (Common Power Format) file to create two power domain, PD_L (low power domain) and PD_H (high power domain) which is used as supply voltage for the critical path section.

Our MIPS processor is divided into three parts, which are datapath, controller and ALU controller. Since datapath has memory and ALU which form the main critical path, we have put it in the high power domain and the others are placed in low power domain.

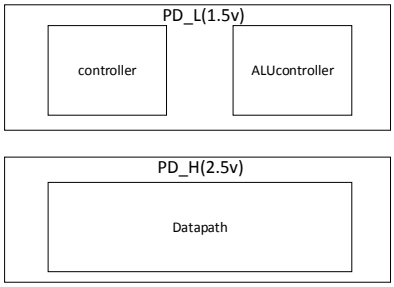

Figure 2: MIPS Power Domain

Voltage shifter is another important issue while designing multiple voltage supply. It includes two kinds of shifter, one is from low voltage to high voltage domain, which is necessary because raised voltage is required in order to make sure signals that cross multiple voltage domains can be sampled correctly, and have enough high voltage to switch the gate, the other kind is opposite from high voltage to low voltage domain, which is optional. In our design, we have simplified the voltage shifter and only considered them for signals going from low power domain to high power domain.

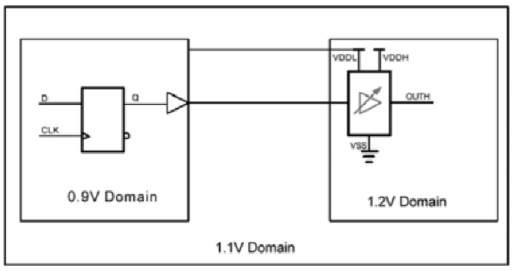

Figure 3: Low to High Level shifter [8]

B. Clock Gating Design

Clock gating is one of the most frequently used techniques in many synchronous circuits to reduce dynamic power consumption without affecting the functionality of the design. Clock gating saves power by adding more logic to a circuit to prune the clock tree, thus disabling portions of the circuitry so that the flip-flops in them do not have to switch states. The switching power consumption goes to zero, and only leakage currents are incurred.Power is reduced by two mechanisms: clock net toggles less frequently, and registers' internal clock buffering switches less often.

The key idea is to reduce or inhibit unnecessary clocking to decrease the switching activity such that only leakage current losses are incurred. We traced the clock tree in the basic MIPS architecture to find that cache is the main load of the clock net and it is not necessary to drive the cache unless write enable signal is asserted. There are mainly two kind of cache in our original circuit, as you can see in the figure 4, one is Instr/Data memory and the other is Register File. We implement the two clock gates, and use MemWrite and RegWrite signal as the Enable signal for clock gate, by doing this so we do not need to add additional control logic.

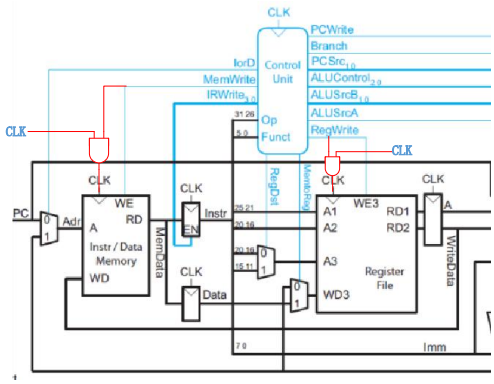

Figure 4: Clock Gating inside MIPS architecture

C. Data Gating Design

Data gating is one of the preferred glitch-resistant way of providing an enable control on a synthesizable register. Data gating is so-called because it controls the data input of the register instead of the clock input. Therefore, the register is continuously clocked. Data gating works by feeding back the register output to its input whilst the enable is in inactive state. We use the same RTL code to explicitly specify clock gating. Inputs are prevented from rippling through multiplier if multiplier output is not selected. However, data gating leads to extra logic in data path that slows timing, and additional area due to gating cells like clock gating.

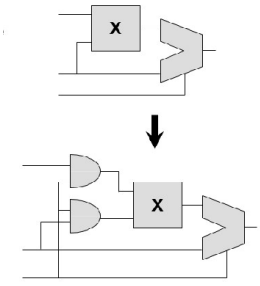

Figure 5: Data Gating

III. Results and Discussions

A. Simulation

We have simulated the design to confirm the functional simulation. The simulation was done on ModelSim with a test-bench written in Verilog. The test-bench tested the MIPS design for the implementation of a Fibonacci code. The module produces output and starts writing when the calculations finish and write is asserted. For example when the test-bench finishes the last numbers 0x05 and 0x08 are added to produce 0x0d, which is then written to memory at address 63.

Figure 7. The simulation of the Basic MIPS pipeline

B. Area and Layout

The layout area for the different designs is shown in table I. The layouts have been generated using the area constraint provided by dc_shell. And they all show similar results as only a few extra gates were added in design. This increased the area by only 100 μm^2, which is less than 1% in terms of overall design are. The dual supply voltage design uses the same code as basic MIPS and hence has the same area.

TABLE I COMPARISON OF LAYOUT AREA FOR THREE DIFFERENT IMPLEMENTATIONS

Design	Area (μm^2)
Basic MIPS	76242
Clock Gating	76386
Data Gating	76134
Two voltage supply	76242

The auto generated layout for the two voltage supply design was created manually in two steps in encounter and hence did not have exact n-well separations as per the design rules. This problem was corrected in layout in cadence. The layout has a ring structure which features the different voltage supplies, shown in figure 9. The outer ring supplies VDD_HIGH, the internal ring supplies VDD_LOW and the innermost ring is the ground supply rail. The lower half of the layout houses the low power domain which essentially contains the main controller and the ALU controller.

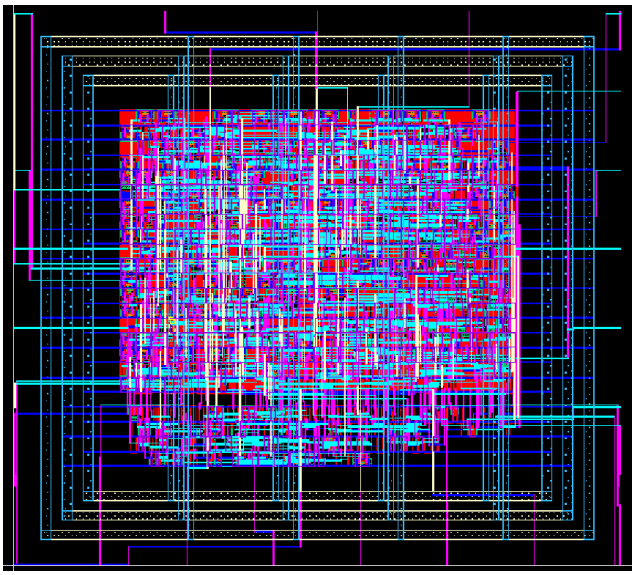

Figure 7: Layout of basic MIPS

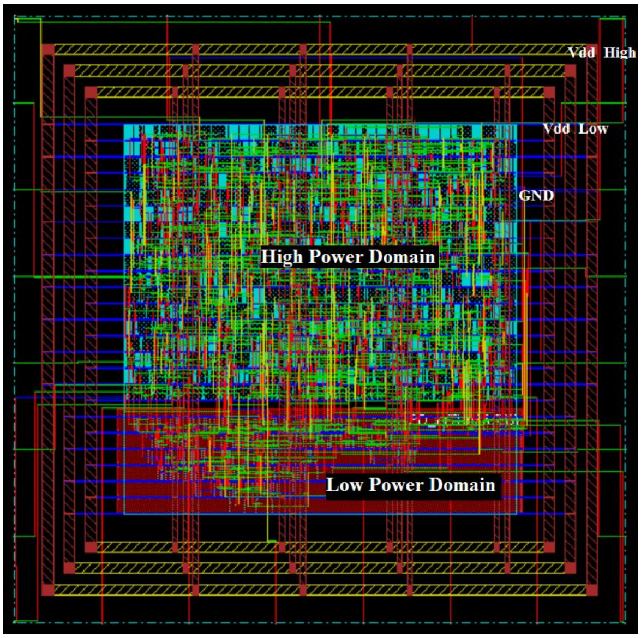

Figure 9. The encounter layout showing the different power supply domains.

C. Power Consumption

The various designs show significant variation in power consumption. The results have been summarized in table II. The clock gated design shows the minimum power

consumption among all the designs. It uses 33% less power than the basic design. This is expected as all the components that are not being used in the current steps are turned off.

TABLE III COMPARISON OF POWER CONSUMPTION FOR THREE DIFFERENT IMPLEMENTATIONS

Design	Power Consumption(mW)			
	Internal	Switching	Leakage	Total
MIPS Basic	23.65	16.7	9.286e-05	40.35
Clock Gating	15.67	11.4	9.311e-05	27.07
Data Gating	24.55	17.56	9.413e-05	42.11
Two voltage supply	23.51	14.59	9.287e-05	38.1

TABLE III COMPARISON OF POWER CONSUMPTION FOR BASIC MIPS AND CLOCK GATING MIPS

Design	Power Consumption(mW)			Percentage of Total Power
	Internal	Switching	Total	
MIPS Basic	1.501	3.495	4.996	12.38
Clock Gating	0.8613	1.996	2.857	10.56

The clock gating significantly reduced the switching and internal power consumed by the clock in the design. It almost reduced to half in magnitude when compared to the original design. The values can be seen in table III. Even though power consumption reduces it still remains a significant percentage of the overall power consumption.

The implementation of data gating introduced more power usage, it increased by 4% with respect to the basic MIPS design. Although it can decrease the switching activity, the data gates introduced in the implementation lead to additional power consumption. Besides, we use simplified MIPS architecture, which does not see much data change inside circuit, in other word, there is no frequent switching activity. Thus no decrease was seen in the switching power either. Hence, we decided to eliminate the data gating from final design.

We tested the design with various modifications such as changing the supply voltage domain of different components. This influences the power consumption in multiple ways. As the number of signals crossing the different voltage domain changes the number of level shifters required also change. This additional introduction of components causes increased power consumption and reduces the effect of placing blocks in separate supply voltage domains. So, this restricts us from placing most of the components in one domain. Since, datapath makes the critical path of the design, it was kept in the high voltage supply domain and the control logic was moved to the lower domain. The dual supply voltage only showed a 5% in power consumption.

The inherent functions in synopsis allows advantages such as reducing the power consumption. The leakage power observed in the design was approximately 92 nW. Although the leakage power in the design is very small, using the feature allows an additional 10% decrease in the leakage power consumptions.

IV. CONCLUSION

The various methods to decrease power consumption have different trade-offs of their own. Implementations like data gating might not prove useful for lesser data movement and can cause significant overhead. Also creating dual supply voltage domains introduces level shifters which increase area slightly and have their own power consumption. The clock gated design showed the best performance among other implementations.

ACKNOWLEDGMENT

We are thankful to Prof. Eisenstadt and the teaching assistant Shaon Yousuf for their guidance during the project. We also thank University of Florida ECE Department for providing the resources we required to complete the project.

REFERENCES

[1] **"A 2.60pJ/Inst Subthreshold Sensor Processor for Optimal Energy Efficiency",** Bo Zhai, Leyla Nazhandali, Javin Olson, Anna Reeves, Michael Minuth, Ryan Helfand,Sanjay Pant, David Blaauw and Todd Austin.

[2] **"Low-power CMOS Digital Design",** A. Chandrakasan, S. Sheng, and R. Brodersen, IEEE J. Solid-State Circuits, vol. 27, pp. 473–483, Apr. 1992.

[3] **"A Leakage Reduction Methodology for Distributed MTCMOS (May, 2004)",** B. Calhoun, et al., IEEE Journal of Solid-State Circuits, Vol. 39, No. 5.

[4] TSMC: Advanced Design for Low Power at 65nm and Below

[5] A practical guide to Low-Power Design, User Experience with CPF.

[6] http://www.cs.utah.edu/~bchong/VLSI/Resources/Low_Power/Encounter_MSV.doc

[7] CMOS VLSI Design: A Circuits and Systems Perspective (4th Edition), Neil Weste and Davis Harris.

[8] http://www.ict.kth.se/courses/IL2200/Lec/Low%20Power%20MultiVoltage%20Design.pdf