An Implementation of a 32-bit ARM Processor Using Dual Power Supplies and Dual Threshold Voltages

Robert Bai, Sarvesh Kulkarni, Wesley Kwong, Ashish Srivastava, Dennis Sylvester, David Blaauw
University of Michigan, Ann Arbor, MI 48105
{rbai, shkulkar, wkwong, ansrivas, dennis, blauuw}@eecs.umich.edu

ABSTRACT

With the explosion of portable electronic devices, power efficient processors have become increasingly important. In this paper we present a set of circuit techniques to implement a 32-bit low-power ARM processor, found commonly in embedded systems, using a six metal layer 0.18μm TSMC process. Our methodology is based on Clustered Voltage Scaling (CVS) and dual-V_{th} techniques aiming to reduce both dynamic power and static power simultaneously.

Categories and Subject Descriptors

VLSI/Deep Submicron

General Terms

Documentation, Performance, Design, Experimentation

Keywords

Low power design, ARM processor, CVS, Dual-V_t

1. INTRODUCTION

Power consumption is a growing problem in deep submicron digital circuit design. Due to the quadratic relationship between power consumption and supply voltage (V_{dd}), significant low-power research has focused on the reduction of V_{dd}. However as we scale the supply voltage down each technology generation, in order to maintain drive current the threshold voltage (V_{th}) must be reduced commensurately. This will have a drastic impact on the leakage power of the device due to the exponential relationship between the threshold voltage and the leakage current (I_{off}). Several practical approaches have been developed to reduce the leakage current including 1) MTCMOS technology gates a high-V_{th} transistor with a sleep mode signal to minimize the leakage current during inactive mode [1], and 2) Dual-V_{th} technology [2,3], via an additional threshold adjust ion implantation step, assigns gates on the critical paths to low V_{th} for speed, while gates that are not timing critical are assigned high V_{th} since they can tolerate larger delay. Algorithms have been developed to optimally assign gates to high or low V_{th} [4,5,6]. However, dual-V_{th} only reduces the static power while assuming a constant V_{dd}. Hence this approach does little to reduce the switching power due to capacitive load charging and discharging. Similarly, if we can lower the supply voltage for the gates on the non-critical paths while still meeting the timing constraint we can also reduce the switching power considerably. One of the most effective techniques that exploits the timing slack present on non-critical paths via multiple supply voltages is clustered voltage scaling (CVS) [7], which assigns low supply voltage to circuits that have excessive slacks. In this paper, we propose a technique that explores the joint benefits of dual-V_{th} and CVS, with the expectation that, by combining both strategies, we can better explore the design space of dynamic power, static power, and timing.

2. ARCHITECTURE

In this design we applied the above techniques to the design and implementation of a 32-bit ARM processor core. The ARM processor is a 32-bit RISC processor with a register-to-register, three-operands instruction set. All operands are 32-bits wide. It has 31 user-accessible general-purpose registers, one of which is a program counter, namely, r15. For ease of design, a 2-stage pipeline is implemented, in which the first stage is used to fetch an instruction and the second stage is used to decode, execute the instruction, and write back the results from the execution. The ARM processor consists of four basic instruction types:

1) Logical and arithmetic operations
2) Data transfers between register file and memory
3) System privilege control
4) Register transfers between coprocessors and ARM

This specific ARM processor includes an integer ALU, integer multiplier, shifter, instruction and data caches, and register file. For this design we synthesized all components using Synopsys Power Compiler with the exception of cache memories which were generated by Memory Compiler.

3. DESIGN METHODOLOGIES

As described in the introduction, cells with four types will be required - high V_{dd}/high V_{th}, high V_{dd}/low V_{th}, low V_{dd}/high V_{th}, and low V_{dd}/low V_{th}. However due to layout

Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI'03)
0-7695-1904-0/03 $17.00 © 2003 IEEE

considerations, to be explained in detail later, we used three flavors of each cell. The design flow for our design can be broken down into several stages. The first stage is the augmentation of an existing standard cell library to incorporate an additional power rail to support dual V_{dd}'s [8]. This additional power rail provides the high V_{dd} to the design while the existing power rail provides low V_{dd}. Secondly, the cells must have their geometries systematically characterized (i.e. pin placement, location of various layers within the cell, etc). This information is critical for the automatic place-and-route (APR) tool, since the tool must know precisely where existing routes are located in order to route other required signals appropriately. Also, the cells must have their electrical switching properties characterized so that a new .lib file can be generated and used in synthesis. The characterization is done by running SPICE simulations on the extracted netlists of the modified cells repeatedly for different input slew rates and output loads, and then measuring the rise and fall times, propagation delays, and static and dynamic power for each cell.

Power and timing constraints are then input into Power Compiler for synthesis in order to optimize for power and timing. Note we only use the high V_{dd} cells in the synthesis run since we subsequently want to assign the high V_{dd} gates on paths that have excessive slack to low V_{dd} after we have run static timing analysis on the design. This approach, which forces the design to start out with high V_{dd} cells exclusively and then selectively changes gates to low V_{dd} cells based on a backwards traversal from primary outputs, obviates the need for asynchronous level converters since the only allowable gate type change is from a high V_{dd} gate to low V_{dd}. Finally we send the optimized gate-level netlist to the APR tool along with the geometric characteristics of the standard cells to produce the final layout.

3.1 Standard cell issues

3.1.1 Standard cell library modification

The standard cell library modified is from Artisan Components, and is designed for fabrication in the TSMC 0.18 μm process. A six metal layer version of the library is used.

Standard cell library providers make available the standard cell layouts of the library in GDS-II format. In addition to this, the foundry provides technology mapping files (different files for different vendor's CAD software), which define the available layers and their associated design rules, and are process specific. CAD tools can then import the layout into their respective layout editor tools. The general flow of importing the standard cells, modifying the cells, and exporting the cell geometry can be succinctly illustrated as shown in Figure 1:

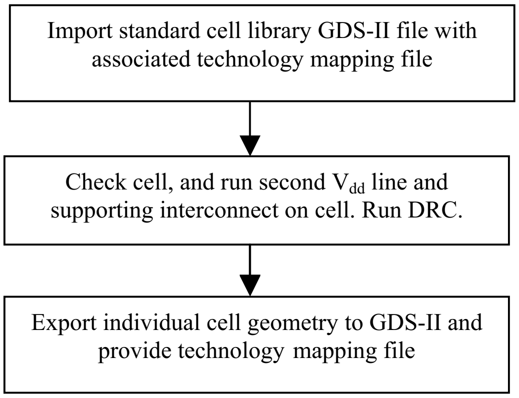
Import standard cell library GDS-II file with associated technology mapping file
Check cell, and run second V_{dd} line and supporting interconnect on cell. Run DRC.
Export individual cell geometry to GDS-II and provide technology mapping file

Figure 1. Flow for importing and exporting cells.

This design used the Cadence suite of CAD tools (Virtuoso, Abstract) for layout, and Diva for DRC of the standard cells. After the cells are imported, the metal line is placed, and then a DRC check is executed. If these are satisfied, export of the cells is accomplished by converting the individual modified layout to GDS-II using layer map files provided by the foundry.

The major challenge faced during the standard cell library modification process was that the width of the top metal line had to be changed to 0.62 μm from 0.80 μm to allow the standard cells during the APR phase of the design to be abutted with each other, since there are geometric limitations on how Silicon Ensemble can place the cells. Some cells could not be converted, because of the placement of certain polysilicon lines. The interconnect employed to connect the higher V_{dd} rail to the rest of the circuit is active (diffusion layer). This clearly rules out any cells with polysilicon running parallel to the power rails. The use of polysilicon to connect to high V_{dd} was also considered (instead of active), due to its lower resistance however this still would not solve the aforementioned problem.

In addition, for low V_{dd} cells the well contacts had to be placed within the high V_{dd} rail. If such contacts were placed in the low V_{dd} rail, any attempt to abut the cells would have resulted in a short circuit between the high and low V_{dd} power supplies. This has the unfortunate side effect of reducing the performance of such cells because of the increase in the V_{th} of the PMOS transistors due to the body effect. It also leads to unbalanced V_{th}'s of the PMOS and NMOS devices within the cell. The latter effect implies unbalanced rise and fall times and precludes the use of the same V_{th}'s for both types of transistors (PMOS and NMOS) in low V_{dd} cells. Hence, we could not use the first two of the four flavors we had intended to use, i.e., (*low V_{dd}/low V_{th}, low V_{dd}/high V_{th}, high V_{dd}/high V_{th}, high V_{dd}/low V_{th}*). Instead, in the case of low V_{dd} cells we used high V_{th} NMOS and low V_{th} PMOS transistors since the body effect will result in an increase in V_{th} of the PMOSFETs and hence overall we expect the V_{th}'s of the PMOSFETs and NMOSFETs to be nearly equal. Therefore, we had three variants for each cell selected from the Artisan Components library: high V_{dd}/high V_{th}, high V_{dd}/low V_{th} and low V_{dd}/(high V_{th} NMOS, low V_{th} PMOS). In total, there are approximately 80 different cells (from the Artisan library) of varying drivability, fan-in, speed, and size – mostly variations on AOI, AND, OAI, OR, NOR, NAND, XOR, and XNOR gates, buffers, inverters, half and full adders, multiplexers, tri-state buffers, scannable D flip-flops and normal D set/reset flip-flops. This resulted in a standard cell library consisting of 240 cells.

3.1.2 Abstract Generation and Sample Place & Route Run

The flow depicting the LEF file generation process is shown in Figure 2. In generating the final LEF files for all the cells in our design, some deviations from the standard Abstract Generator flow were needed in order for the P&R tool to recognize both the power rails separately and create the correct output.

3.1.3 Layout Verification Environment

Diva was used to run a rudimentary DRC check on each standard cell as it is being laid out; however, the final layout uses Mentor Graphics' Calibre to perform checking, due to its greater performance, especially with large designs, and also due to its ability to cope with certain geometric configurations that Diva cannot (e.g. bent gates).

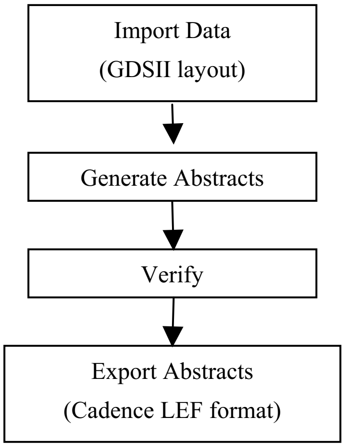

Figure 2. LEF file generation flow.

3.2 Electrical characterization of the modified cells

Electrical characterization of the modified cells is an integral part of the synthesis process. For the tool to properly synthesize the design under the power and timing constraints, it must know the switching characteristics of the cells available to it. Perl scripts were written to automate the process of running SPICE simulations on the cells and obtaining their rise and fall times, propagation delay, and power information based on the input slew rates and output capacitive loads. The scripts were verified by running them on the same library cells used by Artisan, and then comparing the results between our runs and those provided by Artisan. Our delay results were within 10-15% of Artisan. Dynamic power results exhibited the same trend, and were closely correlated. However during the leakage power correlation process, it was found that the original library underestimated the leakage power because it ignored the state-dependency of the leakage. The original library computed leakage for both an output high and output low condition but it is unclear whether the worst-case state vector was used in either run. Without any *a priori* knowledge of the circuit topology in which the cells are placed, we assume that all states of the cell are equally likely to occur. Therefore the total leakage is measured by averaging the leakage power for each state.

3.3 Synthesis (Two-level approach)

The primary hurdle for synthesis using multiple voltage supplies is converting the low voltage outputs of a cell to higher voltage when they act as inputs to higher V_{dd} cells. For our dual V_{dd} case this occurs whenever a low V_{dd} cell feeds a high V_{dd} cell. Thus a level converter circuit is required whenever we try to elevate the voltage levels. Level converters do not perform any logical operation and hence a synthesis tool will not select a level converter cell for the design even if present in the library. A simple approach of providing the entire dual V_{dd} and dual threshold library to the synthesis tool will not work.

Therefore we divide synthesis into two levels. The first level of synthesis uses the dual threshold library at the higher V_{dd} with constraints on leakage and dynamic power. The timing constraints would be tighter than required so that the first level of synthesis provides enough margins for the second part of synthesis to yield reasonable savings in power. The second part of synthesis is based on clustered voltage scaling (CVS) which allows for only one change in going from high V_{dd} cell to low V_{dd} cells along a path. Since asynchronous level converters are not required in going from high to low V_{dd}, only the flip flops need to be level converting i.e. synchronous level converters. A timing analysis is carried out on the design obtained from Power Compiler, and gates along the path with the maximum slack are set to low V_{dd} iteratively, starting from the end of the path at the flip-flop. A similar scheme where all gates are set to low V_{dd} to start with, and then gates are changed to high V_{dd} starting from the start of the path is also possible, but may lead to timing violations even after all possible gates have been set to high V_{dd}. In the approach taken for this design we are guaranteed to meet timing, although neither approach is optimal in terms of the performance gains they provide. The approach to set gates to the lower V_{dd} is iterative – in each iteration a set of paths are selected through PrimeTime and candidate gates that can be set to low V_{dd} are identified. Each time a gate is set to low V_{dd} we search for the fan-outs of nodes at the input of this gate to other high V_{dd} gates. If there are such fan-outs then timing through the inputs of the gate is disabled. This prevents the gate which fan-outs to both high and low V_{dd} gates from being set to low V_{dd}. Also, the timing paths through the fan-out of the output nodes are enabled whenever a gate is set to low V_{dd}. The approach can be easily extended to multiple output gates where the gate is not set to low V_{dd} until that gate is found to be a candidate for low V_{dd} by tracing paths through all of its outputs. This second level of synthesis would give us quadratic savings in dynamic power, which is the goal of a dual V_{dd} design. Also since we are only using one type of low V_{dd} cell (with high threshold voltages), the replacement of high V_{dd} cells by low V_{dd} cells will reduce the static power dissipation as well. After making an estimate of the power savings obtained through the second level of synthesis, we relax the power constraint from our first level of synthesis and maintain a tight timing constraint in order to obtain better synthesis results in the succeeding second level optimization.

4. SELECTION OF V_{DD} AND V_{TH}

For the TSMC 0.18 μm process at the typical process corner, $V_{dd} = 1.8V$, V_{th} (NMOS) = 0.481V, V_{th} (PMOS) = -0.434V. Letting this threshold voltage be the lower of the two projected V_{th} values, we select the value for $V_{th,high}$ based on knowledge that ($V_{th,high}$ - $V_{th,low}$) should not be too large or too small. If the difference is too large, the effect of using $V_{th,high}$ to curb leakage power becomes very prominent. In this case, however, the number of $V_{th,high}$ transistors used will typically be small due to the large delay penalty. On the other hand, if the difference is too small, the benefit of using $V_{th,high}$ becomes suspect. There is an optimal value for the Vth differential. After running simulations, we found that if we set $V_{th,high}$ = $V_{th,low}$ + 0.1V, we are able to reduce the leakage power appreciable while keeping the delay in check, which is consistent with typical industry processes [2,3]. The simulation results for an inverter and 2-input NAND, each driving a load of 168fF with an input slew of 0.4ns, are shown in Tables 1 and 2, respectively.

Table 1. Inverter delay and power characteristics for $V_{th,low}$, and $V_{th,high}$

	Dynamic Energy (fJ)	Static Power (pW)	Rise Delay (ns)	Fall Delay (ns)
V_{th}= 0.458V	18.7	20	0.925	0.553
V_{th}= 0.558V	15.6	7.2	1	0.589

Table 2. 2-input NAND delay and power characteristics for $V_{th,low}$, and $V_{th,high}$

	Dynamic Energy (fJ)	Static Power (pW)	Rise Delay (ns)	Fall Delay (ns)
V_{th}= 0.458V	55.2	65.8	0.544	0.351
V_{th}= 0.558V	47.2	22.8	0.579	0.379

The impact of moving from $V_{th,high}$ to $V_{th,low}$ is to increase the delay by approximately 7-8% and reduce the static power by a factor of 2.8. This is due to the exponential relationship between threshold voltage and subthreshold current. Next we look at the selection of the low V_{dd}. We adopted the results presented in [9] stating that power is minimized in dual-supply systems when the lower V_{dd} is ~30% smaller than the higher V_{dd}. Using this rule of thumb, simulation results for two inverters (one driven by low V_{dd} and using $V_{th,high}$ ($V_{th,low}$) for NMOS (PMOS), and the other by high V_{dd} and $V_{th,low}$ for both NMOS and PMOS), each driving a load of 168fF and with an input slew of 0.4ns, have been obtained (Table 3).

Table 3. Inverter switching and power characteristics as a function of V_{dd} ($V_{th} = V_{th,low}$)

	Dynamic Energy (fJ)	Static Power (pW)	Rise Delay (ns)	Fall Delay (ns)
V_{dd}= 1.8V	18.7	20	0.925	0.553
V_{dd}= 1.26V	7.1	5	1.31	0.789

In this case the dynamic energy goes down by a factor of 2.6 while the delay increases by 42% as we move to a lower supply voltage. More interestingly, the static power drops by a factor of 4. This is due to the presence of high V_{th} MOSFETS in the low V_{dd} cells.

5. LEVEL CONVERSION

When implementing dual power supply integrated circuits, care must be taken to avoid scenarios where gates at low V_{dd} drive gates at high V_{dd}. This is because the output of the low V_{dd} gate will never be raised higher than the value of low V_{dd} which will cause the PMOS device at high V_{dd} to become weakly on, conducting static current from the supply to ground. We hence need to insert a level converting circuit at the interface between low V_{dd} and high V_{dd} cells that converts low V_{dd} to high V_{dd}. Since we are implementing CVS, we need to add these circuits only at the outputs of flip-flops driven by low V_{dd}. We used the traditional circuit in Figure 3 for level conversion [10]. The circuit was optimized for delay using the built-in circuit optimizer in HSPICE.

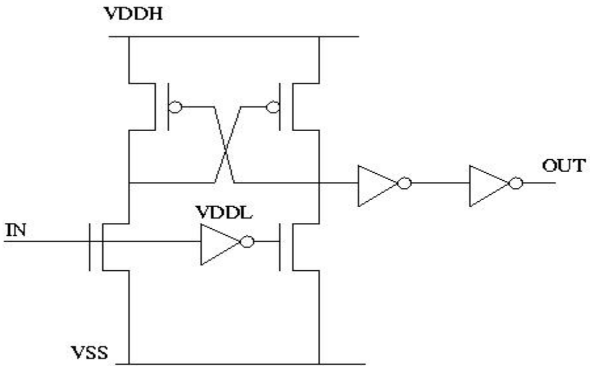

Figure 3. Conventional level converter circuit topology.

6. CACHE MEMORY DESIGN

The ARM that we have implemented has two caches: the data cache and the instruction cache. The memories needed here were generated using the Artisan Components High-Speed Dual-Port SRAM Generator and run at the typical process paramenters of the 0.18µm TSMC process. The areas of instruction and data cache controllers were measured to be 7 mm^2 and 25 mm^2 respectively.

7. SIMULATION RESULTS

We applied our dual-V_{dd} and dual-V_{th} techniques to synthesizing the ARM9 processor core running at 20MHz. The results for leakage power are shown in Table 4, while internal switching power, signal nets switching power, and total dynamic power results are shown in Table 5.

Table 4. Leakage power results

Original Artisan library (µW)	Our library w/o any optimization (µW)	Our library w/ dual-V_{th} optimization (µW)	Our library w/ dual-V_{th} & dual-V_{dd} optimization (µW)
2.54	2.86	1.62	1.52

Table 5. Dynamic power results

	Original Artisan library (mW)	Our library w/o any optimization (mW)	Our library w/ dual-V_{th} & dual-V_{dd} optimization (mW)
Internal Switching Power	117.71	96.74	95.44
Signal Net Switching Power	858.77	835.28	721.54
Total Dynamic Power	976.48	932.02	816.98

Using the 2-level optimization, we were able to reduce the leakage power by 40% when compared to the design using the Artisan Component's original full .lib file. If compared to the same design using our abridged version of .lib without any optimization, we reduced leakage power by 46%. We expect that, with a richer library, we could save additional leakage power. We also reduced total dynamic power by 15% at the same time. The penalty for these power savings is an increase in the chip area after optimization of 14%. This is unavoidable due to the fact our design methodology calls for a secondary power supply and hence an additional power ring. The area problem is further exacerbated by the fact that we could not abut the high V_{dd} rails due to layout constraints imposed on the location of the rails.

8. CONCLUSION

In this paper, we demonstrated reductions in both static and dynamic power by applying CVS and dual-V_{th} circuit techniques simultaneously. For an ARM processor core we reduce the static power by 40% and dynamic power by 15% with a fixed performance level. For future work, we are investigating the design of faster and lower energy level-converting flip-flops, which can be the bottleneck on a path going from low-V_{dd} gates to high-V_{dd} gates. We are also working on issues of transistor sizing in conjunction with CVS and dual-V_{th} to allow more gates to be assigned low V_{dd} and further reduce the dynamic power consumption.

REFERENCES

[1] S. Mutoh, *et al*., "1V high-speed digital circuit technology with 0.5µm multi-threshold CMOS," in *Proc. IEEE International ASIC Conference*, pp. 186-189, 1993.

[2] Z. Chen, *et al*., "0.18µm dual V_t MOSFET process and energy-delay measurement," in *International Electron Devices Meeting (IEDM)*, pp. 851-854, 1996.

[3] N. Rohrer, *et al*., "A 480MHz RISC microprocessor in a 0.12µm L_{eff} CMOS technology with copper interconnects," in *IEEE Journal of Solid-State Circuits*, v. 33, pp.1609-1616, Nov. 1998.

[4] S. Sirichotiyakul, *et al.*, "Standby power minimization through simultaneous threshold voltage selection and circuit sizing," *Proc. ACM/IEEE Design Automation Conference*, pp. 436-441, 1999.

[5] Q. Wang and S.Vrudhula, "Algorithms for minimizing standby power in deep submicron, dual-V_t CMOS circuits," *IEEE Transactions on CAD,* v. 21, 2002.

[6] L. Wei, K. Roy, and C.–K. Koh, "Power minimization by simultaneous dual-V_{th} assignment and gate sizing," *Proc. IEEE Custom Integrated Circuits Conference*, pp. 413-416, 2000.

[7] K. Usami and M. Horowitz, "Clustered voltage scaling technique for low-power design," *Proc. International Symposium on Low-Power Electronics and Design*, pp. 3-8, 1995.

[8] C. Yeh, Y. Kang, S. Shieh and J. Wang, **"**Layout techniques supporting the use of dual supply voltages for cell-based designs," *Proc. ACM/IEEE Design Automation Conference,* pp. 62-67, 1999.

[9] M. Hamada, Y. Ootaguro and T. Kuroda, "Utilizing surplus timing for power reduction," *Proc. IEEE Custom Integrated Circuits Conference*, pp. 89-92, 2001.

[10] M. Hamada, *et. al*., "A top-down low power design technique using clustered voltage scaling with variable supply-voltage scheme," *Proc. IEEE Custom Integrated Circuits Conference*, pp. 495-498, 1998.