

Passive Precharge and Rippled Power Logic (PPRPL)

Samuel B. Schaevitz, Christopher Lin

Department of Electrical Engineering, Massachusetts Institute of Technology
sams@mit.edu, linc99@mit.edu

1. ABSTRACT

A low-power, high-speed logic style using Passive Precharge and Rippled Power is proposed. Ultra-low threshold voltage (V_t) devices permit high speed operation, while the heavy leakage current pre-charges dynamic nodes. High V_t devices prevent leakage through the logic. The high V_t devices provide power to evaluate a sequence of logic gates and are activated in series for periods of time which are short relative to the clock period. The power effectively ripples through the logic path. These innovations combine to produce low power circuits that maintain very high speeds. A 16 bit by 16 bit multiplier was simulated in HSPICE using this logic style. We achieved a clock rate of 1 GHz with a latency of 1.3 ns. At that clock frequency the power dissipation is 10.9 mW.

2. INTRODUCTION

Power consumption is of paramount importance in the design of modern electronics. Low-power techniques that do not sacrifice performance are needed to meet the computational needs of contemporary and future applications.

Lowering the V_t of devices enables designers to achieve very high-speed computation and reduce the circuit supply voltage. However, these benefits are limited by the resultant increase in sub-threshold leakage current and, therefore, standby power consumption. Multi-Threshold CMOS (MTCMOS) attempts to save this sub-threshold leakage power by using high V_t devices to disconnect logic blocks from power nodes when they are not needed. Unfortunately, MTCMOS is limited by a number of problems [1]-[3].

Shutting off modules, using gated clocks or high V_t devices, only reduces average, not peak, power. In stationary applications, heat dissipation is the foremost concern with regards to power. Therefore, peak power consumption is the relevant metric. In addition, gated clocks do nothing to limit standby power due to sub-threshold leakage current, and they cause other problems including variable clock load and clock skew.

Our Passive Precharge and Rippled Power Logic (PPRPL) reduces peak and average power while maintaining low latency and high throughput. While it is fully compatible with existing fabrication technology and existing logic styles, PPRPL is a distinct innovation.

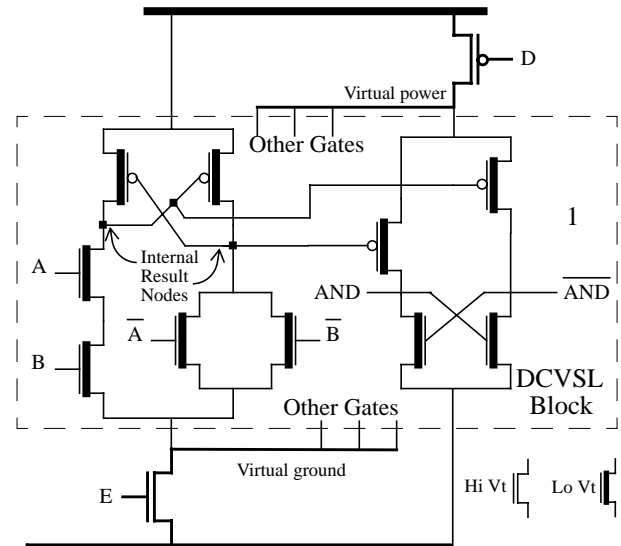


Figure 1: AND gate implemented in PPRPL.

3. PPRPL

Figure 1 shows a two input AND gate implemented in this style. The main body of the logic is a static DCVSL (Differential Cascode Voltage Switch Logic) implementation of AND followed by a complementary DCVSL inverter. These transistors are all ultra-low V_t devices. The high V_t devices insulating the logic from ground or power are enable switches. When an enable transistor is conductive, the corresponding logic is active, and the outputs are computed.

Although there is no explicit precharge transistor, nor is there an explicit precharge phase, PPRPL is a dynamic logic style. When a subcircuit's enable transistor is off, the internal and output nodes of the logic drift to the supply voltage (or ground) because of sub-threshold leakage current, hence "Passive Precharge."

Power is supplied to the logic by first activating the NMOS and then the PMOS high V_t devices using overlapping pulses generated by control logic (see figure 2), which is triggered by the rising edge of an enable pulse. Later stages are activated sequentially using overlapping pulses in the same manner, hence "Rippled Power."

Figure 3 depicts the operation of a single three input AND gate. All of the inputs switch high at 1 ns. Part (a) shows the control signals and virtual power and ground nodes. Part (b) shows the internal result nodes and the output nodes. Note, the line patterns (dashed and solid) correspond to nodes in the same subcircuit (the actual AND gate and the inverter buffer, respectively).

The behavior of the gate is readily apparent. When "e" switches high, virtual ground starts to be pulled towards ground. Only one of the internal result nodes of the first subcircuit switches. When "e" switches back down, both virtual ground and the internal nodes passively precharge back to their initial values. The pulse "d" overlaps slightly with pulse "e" allowing the second subcircuit to begin computing while the first subcircuit result is becoming valid. Similar to before, the virtual power node is driven towards power while one of the outputs switch. When "d" drops back to ground, both nodes passively precharge to their initial values.

Because we are using a dynamic and differential gate design, our outputs do not need to switch fully. Low swing speeds computation

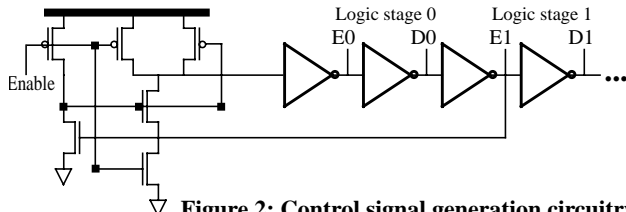


Figure 2: Control signal generation circuitry.

and improves power consumption. Also, the nature of our logic precludes erroneous switching, further checking power waste.

The reader may note that virtual power and ground both begin this simulation fully discharged, but they are not fully discharged at the end. Virtual power and ground may discharge completely to ground and power respectively in cases when the inputs remain valid long after the stage has been shut off. As that is the worst case scenario, it is the initial condition chosen for simulations.

Complete discharge of the virtual power and ground nodes does not normally happen. This is due to the body effect. In the case of the NMOS tree, the transistors experience a much higher threshold voltage as virtual ground rises. As a result, their leakage current drops off exponentially. Virtual power may actually drift slightly towards ground due to capacitive coupling with the precharging inputs. Similar effects influence virtual ground's behavior.

The complementary inverter subcircuit is needed to prevent race conditions from arising, and it allows us to run the logic at a much higher speed. Ensuring correct timing of cascaded non-complementary gates is possible. However, doing so is prohibitively slow. Instead we use a Domino-like variation of the NORA, a.k.a. np-CMOS, logic style to ensure that the outputs from each subcircuit are precharged correctly for the next subcircuit. We chose to put only an inverter (as in Domino) in the PMOS pull-up trees because anything more complicated unacceptably slowed down the computation. The static inverters of a pure Domino scheme would have to be built out of high Vt transistors to prevent heavy leakage. Unfortunately, that too would severely degrade performance.

Within the Domino-NORA framework, DCVSL is not the only possible gate design. We chose to use DCVSL because the complementary outputs are well suited to the function of the multiplier. In situations where differential outputs are of no use, variations which more closely resemble pseudo-NMOS logic may be preferred (see figure 4).

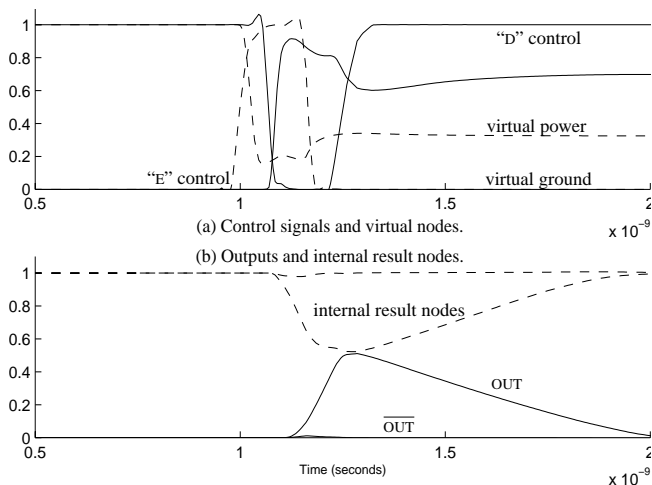


Figure 3: Interesting nodes in a three input AND gate.

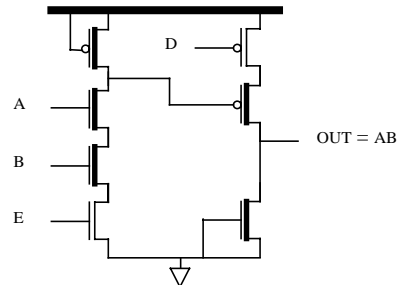


Figure 4: PPRPL built around a pseudo-NMOS-like gate.

3.1 Power and Performance

PPRPL maintains and improves performance while saving power in a variety of ways. The ultra-low Vt devices allow us to operate at a very high speed while reducing supply voltages significantly. They also allow us to decrease gate sizes so that the majority of logic transistors are minimum sized. This greatly diminished switched capacitance, which further reduces power consumption.

PPRPL works by activating the high Vt devices in succession to take advantage of sub-threshold leakage. That is, every section of a circuit is off until it is about to be used. This off period is used for Passive Precharge. Then, each stage is on only for the time it needs to complete its computation. This duration is only a small portion of the clock cycle. Because of the fractional activity cycle of each logic stage in a given module, PPRPL saves power even while it is running at peak power in active mode for a prolonged period of time. For example, each stage in our multiplier is on for only 10% of the clock cycle. This translates into an 10x reduction in sub-threshold loss during active mode.

PPRPL uses high Vt devices differently from typical MTCMOS. Consequently, we are able to avoid many of the problems associated with their cousins in conventional MTCMOS. MTCMOS power transistors are enormous. They require a lot of power and time to switch, much longer than a single clock period. The size of our power transistors are on the order of the active precharge transistors we have eliminated. Because they are so small, they switch quickly enough to ripple power through the stages of large logic blocks. By reducing the clock load, we offset the power needed to drive these enable transistors.

3.2 Timing

The nature of PPRPL creates a very interesting timing feature which we call "intrinsic pipelining." Because the nodes precharge continuously, with a rate controlled by transistor sizing, it is possible to pipeline a PPRPL module without any memory elements. If care is taken to ensure that there is enough delay between successive computations to allow the nodes to precharge, there will be no interference between the two tasks.

3.3 Compatibility

Since the output of our logic is almost identical to the output from a dynamic gate, integrating PPRPL into existing digital systems is a trivial task. Many of the same techniques used to interface dynamic gates with downstream dynamic or static gates can be used with our logic.

In addition, our logic enables easy integration with Complementary Pass-transistor Logic (CPL). Because we use ultra-low Vt devices, no additional engineering is required to add fast CPL. The high leakage normally associated with low Vt CPL is eliminated by the use of PPRPL to drive the logic (see figure 5). In fact, we take advantage of this natural combination in our multiplier example.

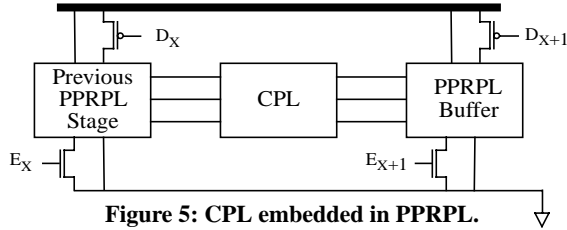


Figure 5: CPL embedded in PPRPL.

4. DEVICE ISSUES

We simulated our technology in HSPICE using 0.1 V threshold transistors for our ultra-low V_t devices and 0.6 V threshold for our high V_t devices. The effective gate length of both transistor types was 0.25 μ m. Figure 6 displays the I-V characteristic of the ultra-low V_t NMOS devices for various operating points.

Fabrication variation is of particular concern when working with ultra-low V_t devices. Differences of a few millivolts that would be inconsequential for higher V_t devices can have profound impacts on the behavior of ultra-low V_t devices. We investigated these effects by simulating a three input AND gate with five different values of V_t (see figure 7). The higher V_t configurations took much longer to precharge after switching. The lower V_t devices did not switch as high as the higher V_t devices. Interestingly, the switching time remained very constant over the values of V_t considered.

The primary concern with regards to PPRPL is the slow precharge of higher V_t devices. As stated before, switching levels do not need to be particularly high since our gates are differential. However, throughput is limited by the speed with which nodes precharge. This variation might be controlled by adjusting the body bias.

Table 1: Interesting Figures

Ultra-low V_t	0.1 Volts	Latency	1.3 ns
High V_t	0.6 Volts	Energy/cycle	10.9 pJ
Eff. gate length	0.25 μ m	Max. clock	1 GHz
# MOSFETS	11262	Power @ 1 GHz	10.9 mW

5. MULTIPLIER EXAMPLE

To demonstrate the feasibility and promise of our logic style, we used it to implement a full 16 bit by 16 bit 2's complement multiplier simulated in HSPICE. The multiplier is well-suited to testing PPRPL since it is large but can be broken up into distinct stages of computation. We use Modified Booth Encoding [4] to pre-code our inputs using CPL [5] followed by a PPRPL buffer (see figure 5). We add the partial products in a Wallace Tree [6] using mirror adders and 3-input XOR gates implemented in PPRPL, and combine the final two summands using a PPRPL carry-look-ahead adder (see figure 8).

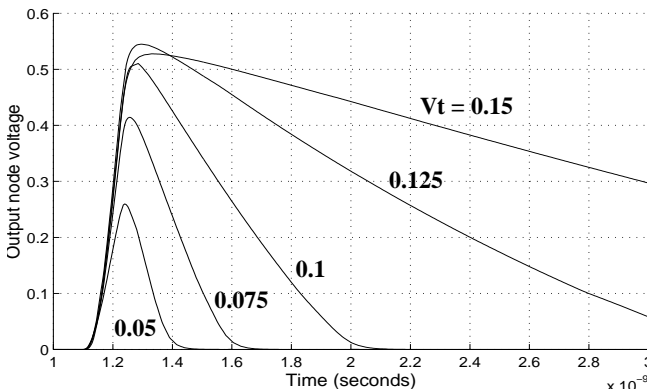


Figure 7: Output node voltage vs. time of a 3 input AND gate

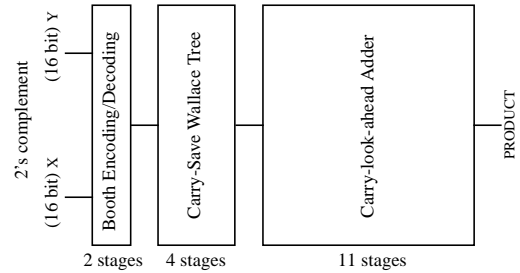


Figure 8: High-level block diagram of multiplier.

This multiplier uses 10.9 pJ and 1.3ns for a single computation, but is pre-charged and ready to begin a new computation in only 1 ns which corresponds to 10.9mW watts at 1 GHz. Table 1 organizes the interesting parameters and performance results of our multiplier.

6. CONCLUSION

Rippled Power in MTCMOS for a multi-stage module, combined with Passive Precharge using low V_t devices, enables extremely efficient power use. Power distribution in PPRPL is optimized within each clock cycle. The result is very low power use during evaluate, negligible power loss during standby, and no penalty in power or performance from switching to or from standby mode.

Our proposed methodology could certainly be used to design an entire general microprocessor. Clock load could theoretically be reduced to only the memory logic. Pipelining could be accomplished with greatly fewer registers, if any at all. All logic modules would compute only when an enable pulse was sent from the microcontroller and would need no other clock signal whatsoever. An in depth evaluation of the implications of this logic on a system level are well beyond the scope of this paper.

7. ACKNOWLEDGEMENTS

The authors would like to thank Keith M. Jackson and Professor Dimitri Antoniadis, both at MIT, for their generous and timely assistance in providing us with transistor models. This paper is the result of our final project for the course Analysis and Design of Digital Circuits (6.374) taught by Professor Anantha Chandrakasan, also at MIT, whom we would like to thank for his encouragement, support, and inspiration.

8. REFERENCES

- [1] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada. "1-V Power Supply High-Speed Digital Circuit Technology with Multithreshold-Voltage CMOS," IEEE Journal of Solid-State Circuits, pp. 847-854, August 1995.
- [2] T. Douseki, S. Shigematsu, Y. Tanabe, M. Harada, H. Inokawa, and T. Tsuchiya. "A 0.5V SIMOX-MTCMOS Circuit with 200ps Logic Gate," IEEE International Solid-State Circuits Conference, pp. 84-85, February 1996.
- [3] J. Kao, S. Narendra, and A. Chandrakasan. "MTCMOS Hierarchical Sizing Based on Mutual Exclusive Discharge Patterns," Proceedings of the 35th annual conference on Design automation conference, pp. 495-500, 1998.
- [4] N. R. Scott. Computer Number Systems & Arithmetic, Prentice Hall, 1985.
- [5] K. Yano, T. Yamanaka, T. Nishida, M. Saito, K. Shimohigashi, and A. Shimizu. "A 3.8- μ m CMOS 16x16-b Multiplier Using Complementary Pass-Transistor Logic," IEEE Journal of Solid-State Circuits, pp. 388-394, April 1990.
- [6] C. S. Wallace. "A Suggestion for a Fast Multiplier," IEEE Trans. Electronic Computers, Vol. EC-13, pp. 14-17, Feb. 1964.