Low-Power Dynamic Voltage Scaling System

Taeg Sang Cho, Ravi Palakodety
Massachusetts Institute of Technology, Cambridge, MA

Introduction

Dynamic Voltage Scaling (DVS) is a method of reducing the supply voltage to exploit savings in power dissipation by finishing critical operations just-in-time. This paper proposes modifications to optimize a DVS scheme proposed by Kuroda, et al.[1] for low-power applications, allowing the circuit to operate at the minimum-energy point. These changes include designing an efficient PWM controller, an efficient test-vector generation scheme for the speed detector, and an adaptive-delay scheme in the speed detector to ensure that the replica tracks the real critical path in light of local process variations.

Section 1: Speed Detector Circuitry

The speed detector circuitry detects whether V_{DDL} is at the lowest possible level to finish a critical path operation in one clock cycle of F_{EXT}. The speed detector compares the output (CP) and delayed output (CPD) of a critical path replica to determine the appropriate level of VDDL. When V_{DDL} is in an acceptable range, CP will finish in one cycle of F_{EXT}, while CPD will not. The critical path is a 32-bit ripple-carry adder, while the delayed critical path (CPD) has an extra full adder supplying a 3% delay. A pseudo-differential level converter with built-in register is used to transition from V_{DDL} to V_{DDH}.

V_t variations can drastically change the delay of a circuit running in the subthreshold region. Since the replica is made of the same blocks as the critical path, global variations in V_t can be tracked closely. Random dopant fluctuations (RDF) can cause local variations in V_t, which can lead to the replica not tracking the real path. A Monte Carlo simulation (Fig 2) shows the delay distribution in a 1-bit subthreshold adder with a 30% spread in V_t variation. The delay distribution for a 32-bit adder is roughly estimated by convolving the 1-bit adder distribution 32 times, by assuming independence. As a rough estimate, we ignore the correlation between the delays for different adders.

A simple adaptive-delay scheme is proposed to correct for these variations. A delay line made of buffers and multiplexor replaces the last 3 full adders in the critical path replica. At startup, the adaptive-delay scheme matches the replica delay to the real path delay by sampling the replica output on a delayed rising edge of the real path, and modifying the number of buffers used in the delay line until the replica finishes slightly after the delayed real path. The real critical path is temporarily used as an oscillator using tristate buffers that are disabled once the alignment phase is finished. Back-to-back registers are used to decrease the probability of a metastable output. This alignment phase takes about 200ns at V_{DDL}=0.9V.

Section 2: Test Vector Generator

Kuroda's speed detection scheme suffers from a major drawback: when V_{DDL} is too low, output latched on the rising edge of F_{EXT} can actually be the signal that was sent from the test vector generator multiple-cycles beforehand. Kuroda et al try to address the problem by having 3 bit counter and ANDing them together to generate the test vector, but this does not completely solve the problem, and could lead to high power dissipation and spurious testing of the circuit, which will not contribute any change at the counter, thus duty cycle.
In this work, a Finite State Machine (Fig. 4) is introduced to address the problem. By removing the high speed counter, we can greatly reduce the power dissipation. The test vector generator in this work consumes 490nW, while Kuroda's test vector generator consumes about 4.3uW (simulated at 65nm process node). Moreover, power is further reduced as the scheme in this paper tests the speed of the circuit once per count-up cycle ($1/F_N$), while Kuroda's scheme tests F_{EXT}/F_N times per count-up cycle.

Section 3: Pulse Width Modulation (PWM) Controller

The duty-cycled pulse generator in Kuroda's paper is not suited for low power applications because it entails the use of a counter operating at 64MHz. This counter alone consumes several milli-watts; thus a new scheme to generate the duty-cycled pulse is called for. In this work, PWM scheme proposed in [2] is adopted with modifications. PWM consists of a counter, a Delay-Locked Loop, a 64-to-1 multiplexer, and an edge-sensitive RS latch (Fig 5). In the proposed scheme, reference clock is delayed by a delay-line, which consists of 64 buffers. DLL is implemented in order to ensure that the delay through the chain of 64 buffers equals the period of the reference frequency signal. Several circuit techniques are employed to limit the power dissipation in DLL. The charge pump circuitry in DLL is biased at sub-threshold region so that the static power dissipated by the current sources is minimized.

The counter accumulates the output of the speed detector to set the desired duty-cycle of the pulse driving the buck converter. The speed at which the counter accumulates is critical for the stability of the system; F_N is thus set as 50kHz, which is the cut-off frequency of the buck converter. The counter has an internal circuitry so that when the output of the speed detector is 00, the duty cycle is increased by one; when the output of the speed detector is 11. By multiplexing the delayed reference signal, for example at 'N'th stage of the delay-line, signal with duty-cycle proportional to 'N' can be generated. To generate the duty-cycled signal, an edge-sensitive RS-latch is used. On the rising edge of the reference clock, a set pulse is generated to set the RS latch, while on the rising edge of the multiplexed signal, reset pulse is triggered to reset the RS latch. This exploits the fact that the RS latch holds the value at RS = 00.

The power consumption of the PWM controller is 9.75uW, which is many orders of magnitude smaller than Kuroda's scheme. DLL exhibits a delay-error of 4%, which can be fixed by increasing the loop-gain, thereby increasing the power dissipation, of DLL. However, 4% error can be compensated by the DVS feedback loop. The power-consumption could further be reduced by limiting the voltage swing in the delay-line, but such technique was not exploited in this work.

Conclusion

In this work, modifications are proposed to address techniques to reduce the sensitivity to process variations and to reduce the power dissipation of the DVS system. System specifications can be found in Fig 7. Further work could be directed to reducing the power dissipation in DLL, providing continuous adaptive-delay in speed-detector, and speeding up the response of the system while not sacrificing the stability.

Reference

1. Variable Supply-Voltage Scheme for Low-Power High-Speed CMOS Digital Design – Kuroda et al. JSSC 1998
2. High-Efficiency Multiple-Output DC-DC Conversion for Low-Voltage Systems – Dancy et al. IEEE. VLSI 2000

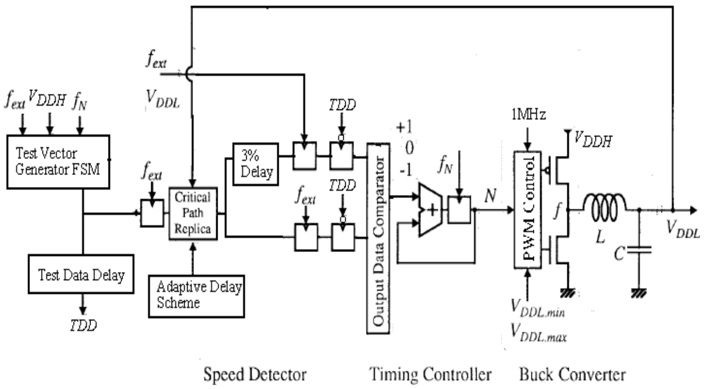

Fig 1. System Block Diagram

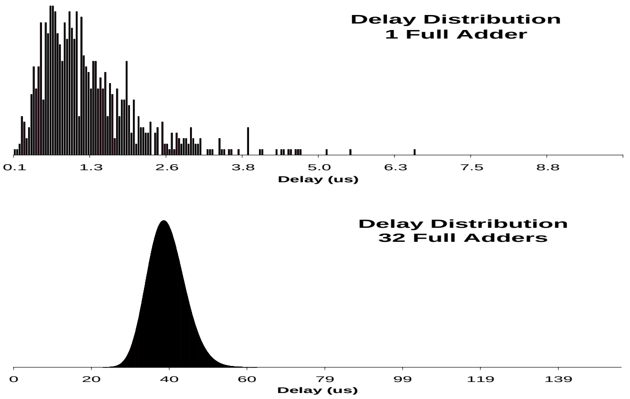

Fig 2: Monte Carlo Delay Analysis of adders with RDF

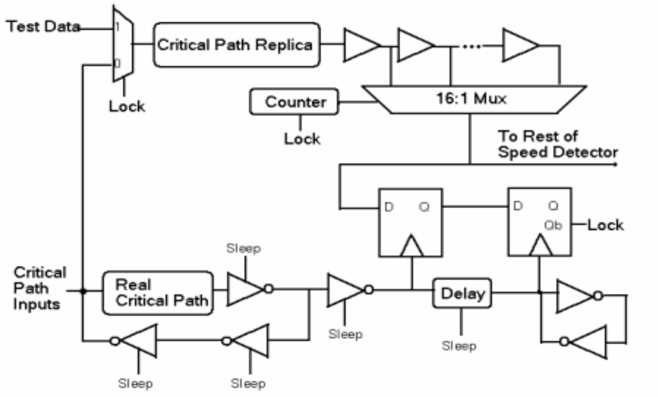

Fig 3. Adaptive Delay Scheme

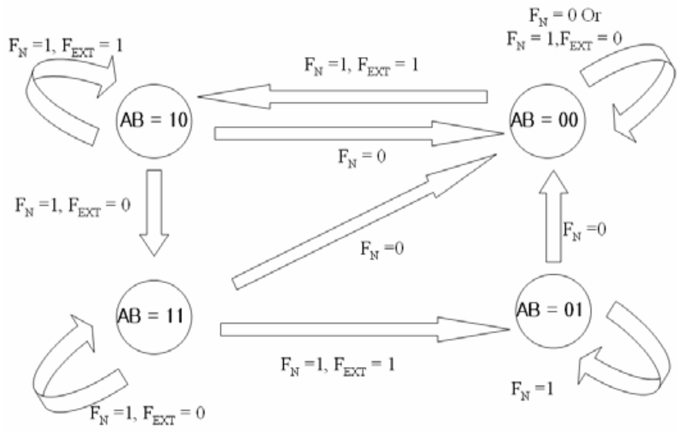

Fig. 4 FSM used in the Test Vector Generator

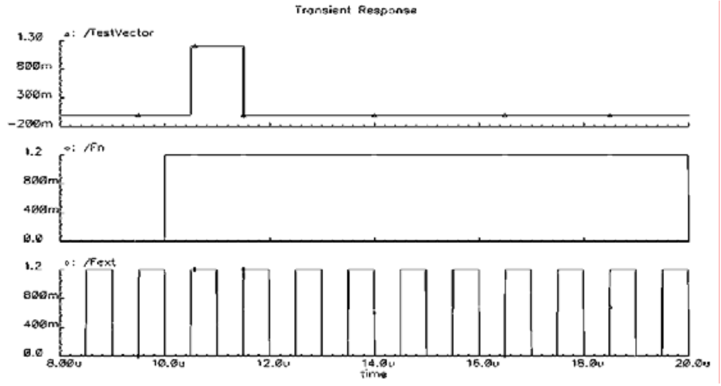

Fig 5. Test Vector Generator Output

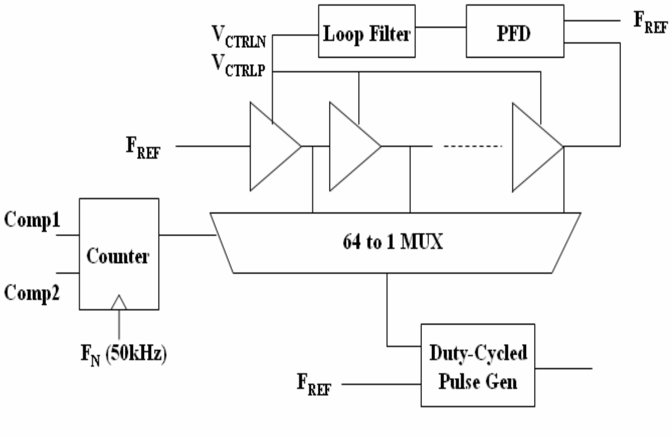

Fig 6. PWM Controller

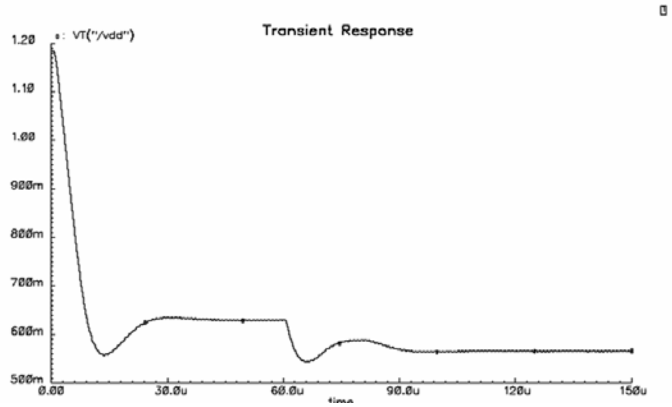

Fig 7. Buck Converter Output when N=32 changes to N=28

Metric	Performance
Load Power Range	**400uW – 5mW**
Output Voltage Range	**0.3 V – 0.9V**
Output Voltage Ripple	**V_{DDL} * 0.02 >**
Efficiency (Including Speed Detector)	**> 88.5 %**

Fig 8. DVS System Specifications