Transparent Mode Flip-Flops for Collapsible Pipelines

Eric L. Hill and Mikko H. Lipasti
University of Wisconsin - Madison
{elhill, mikko}@ece.wisc.edu

Abstract

Prior work has shown that collapsible pipelining techniques have the potential to significantly reduce clocking activity, which can consume up to 70% of the dynamic power in modern high performance microprocessors. Previous collapsible pipeline proposals either rely on single phase clocking (by forcing latches into transparent state) or do not discuss the mechanisms by which stages are merged. In this work two flip-flop designs featuring an additional transparent state suitable for collapsing stages are presented. Transparency is achieved either by decoupling the master and slave clocks to keep both latches transparent, or by using a bypass mux that routes around the flip-flop. Both of these designs are evaluated in the context of transparently gated pipelines, an ad-hoc collapsible pipelining technique. Detailed analysis shows that the decoupled clock flip-flop is the most attractive in terms of energy and delay.

1. Introduction

Transparent pipeline gating was originally proposed to reduce switching activity in pipelined data paths by forcing subsets of timing elements into a transparent state during periods of low utilization. This technique most readily lends itself to pipelines clocked using latches, as the clock signal can simply be held high to keep a latch in a transparent state. The original work proposing this idea applied the technique to a 2-phase master slave pipeline [3]. While transparent pipeline gating is a promising technique as presented, there are significant challenges associated with distributing multiple clocks across an entire chip. Furthermore, other studies on adaptive pipeline depth scaling techniques [6][9] either rely on single phase clocking, as discussed above, or do not discuss the mechanism by which stages are merged. For this reason, we explore the application of transparent pipeline gating to systems with single phase clocking. To avoid difficulty satisfying min-delay constraints, timing elements in such a pipeline require hard-edges to propagate data. In order to be used in a transparent pipeline, edge-triggered flip-flops must feature an additional mode where they behave similar to a latch in transparent mode. In this work two possible transparent mode flip-flop designs are presented, and evaluated in terms of energy consumption and delay.

The remainder of this paper is organized as follows. Section 2 provides a brief introduction to transparent pipeline gating. Despite the fact that the flip-flops presented in this work are applicable to all collapsible pipelines, the evaluation presented is specifically in the context of transparent pipeline gating. Section 3 describes the attributes of each flip-flop design evaluated. The methodology used to characterize the energy and delay characteristics of each design are discussed in Section 4. Detailed energy and delay data is presented and discussed in Sections 5, 6, and 7, while Section 8 concludes the paper.

2. Transparent pipeline gating

Transparent pipeline gating is an ad-hoc pipeline scaling technique capable of adapting pipeline depth on a cycle-by-cycle basis in order to accommodate changing data arrival rates. The basic idea behind this technique is that given a pipelined data path, and a low arrival rate, values need only be stored in a subset of the available pipeline registers while the others remain in a transparent state. In order to guarantee functional correctness (prevent data races), only enough registers need to be enabled such that each unit of work propagating through the pipeline is separated by one enabled (non-transparent) timing element. In order for the pipeline to adapt on a cycle-by-cycle basis, there also needs to be an additional one-bit control pipeline (which is also necessary for valid-bit clock gating) and additional control logic for each pipeline register to look back at previous pipeline stages to make a decision on which state the register should be in. More details about transparent pipeline gating can be found in [3].

1-4244-1258-7/07/$25.00 ©2007 IEEE

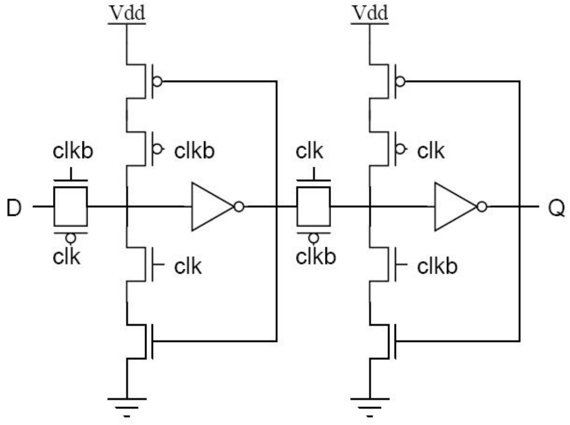

Figure 1. PowerPC flip-flop.

3. Flip-flop designs

3.1 Baseline flip-flop

All designs considered in this work were derived from the PowerPC master-slave flip-flop shown in Figure 1. A prior study on low power and high performance sequential elements compared several industrial flip-flops, and identified the PowerPC design as being attractive in terms of both power and delay characteristics [2]. For this reason, the PowerPC flip-flop was chosen as a baseline. This flip-flop has two operating modes – a clocked mode (where input transitions are propagated with clock edges) and an opaque mode (where the clock is held low, preventing D from propagating to Q) used during cycles where the logic following the flip-flop is not performing any useful computation.

3.2 Decoupled clock transparent flip-flop

In order to create a flip-flop with a transparent mode (where changes in D are immediately reflected in the output Q) both the master and slave latches need to be made transparent simultaneously. This cannot be achieved with conventional edge-triggered flip-flops as complementary clock signals are used to control master and slave latches. One straight-forward method for allowing an additional transparent mode is to allow the master and slave clocks to be controlled independently. Such a design is shown in Figure 2. Throughout the remainder of this paper, this design will be referred to as the decoupled clock or DCLK design. This design is similar to the baseline design except that the master and slave latches are now controlled by different clock signals (clock1 and clock2, respectively). Clock1 and clock2 are complements of one another when the flip-flop is in the clocked or gated modes, but in transparent mode both clock signals are gated low, allowing the timing element to behave like an open latch. While both clocks are capable of operating independently, their signals can be derived locally from a single global clock. Figure 3 shows the required clock generation logic, the cost of which is amortized over an entire pipestage register (usually at least several dozen single-bit flip-flops). In clocked and opaque modes, clock1 and clock2 are complements of one another, mimicking the behavior of a conventional edge-triggered flip-flop. In transparent mode (when gate_trans is high), clock1 and clock2 have identical polarities and are gated low. This converts the flip flop into a transparent structure where changes in D immediately affect the output of Q.

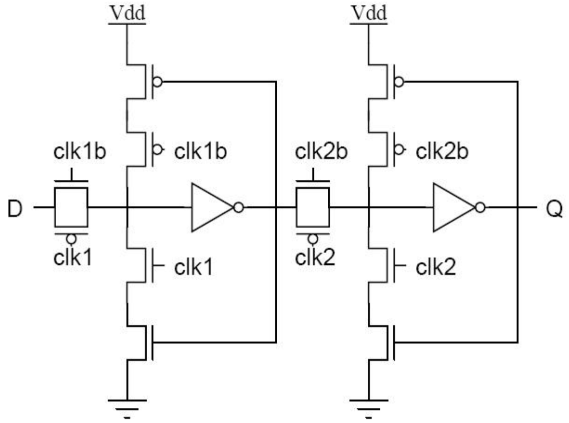

Figure 2. DCLK flip-flop design.

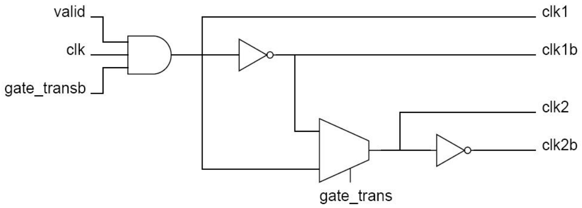

Figure 3. Clock generation logic at local buffer for DCLK flip-flop.

3.3 Transparent flip-flop with bypass

A second possible transparent mode flip-flop design is shown in Figure 4. This design is identical to the baseline flip-flop except for two transmission gates that are used to implement a bypass path connecting D and Q in transparent mode. Throughout the remainder of this paper, this flip-flop will be referred to as the bypass or BYP design. This design was inspired by the work presented in [5]. While using this design introduces an area penalty (due to the bypass path), it also has less control overhead than the previously presented flip-flop, since only one signal (driving the additional transmission gates) needs to be toggled to bring the timing element into or out of transparent mode. This flip-flop can be in either of the first two operating modes (clocked or gated) when transparent gating signal is low (meaning the bypass path is disabled), and in its transparent mode when clock is gated low and the transparent gating signal is high.

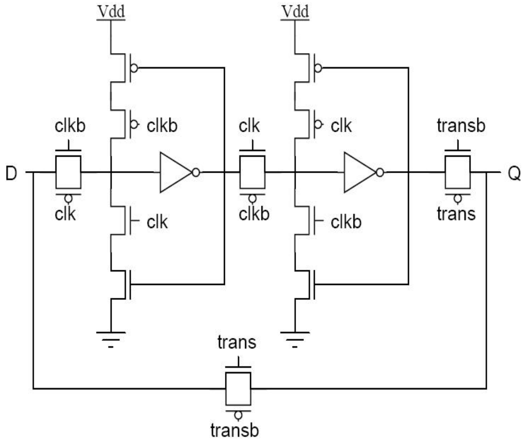

Figure 4. BYP flip-flop design.

4. Measurement methodology

4.1 Test Bench

Each flip-flop design considered was evaluated using an environment similar to the test bench setup described in [2]. One notable difference in our setup is that our test bench does not include the buffering for clock signals. This choice was made because the cost of the clock generation logic will be amortized across an entire rank of flip-flops (comprising a pipeline register) rather than for each single-bit element, and consists of an identical fixed overhead for each of the designs we evaluate. This implies that the transistors in the generation logic can be sized to meet any desired delay constraints.

4.2 Energy and delay estimation

The transition graph methodology proposed in [12] was used to measure the energy consumption of the various flip-flops studied. In this method, all possible circuit states (combinations of input signals and discrete internal node voltages) are first enumerated. Next, starting with a known reachable state, all other reachable states are computed, resulting in the construction of a state transition graph. Figure 5 shows the canonical state transitions graphs from each of the flip-flops studied. These graphs are simplified in that multiple states with differing internal node voltage combinations but identical input and output voltages are merged. Dynamic energy is consumed when the circuit transitions from one stable state to another. The paths formed by the bold arrows between states in each diagram represent the state transitions that need to occur to propagate a low-to-high input transition from D to Q. Transparent mode flip-flops save energy by propagating input transitions without toggling the clock. For each flip-flop studied each transition in the associated state transition diagram was simulated in HSPICE. All designs were modeled in 65nm technology using the Predictive Technology Model (PTM) [8].

In addition to using the metric of minimum D-to-Q delay (the measured setup time plus the associated clock-to-Q delay) as was done in [2], the propagation delay (in transparent mode) was also measured for each of the proposed flip-flops.

5. Results

5.1 Energy characterization

The energy costs for propagating all possible input transitions in both normal and transparent modes are shown in Table 1. The propagation of an input change (in D) from low to high (the upper left hand box in the table) is represented by the sequence of state transitions highlighted in section (a) of Figure 5. In order for transparent pipeline gating to be an effective technique for energy efficiency, input transitions need to be less expensive to propagate in transparent mode than they are in clocked mode. It is clear from Table 1 that this is true for both transparent flip-flop designs presented.

It is also important to note that the cost of propagating an input transition in clocked mode is actually more expensive for BYP than it is for the baseline design. This means that for pipelines that are heavily utilized (implying the majority of input transitions are propagated in clocked mode), the energy consumption of a pipeline with registers composed of BYP flip-flops will exceed that of a pipeline with flip-flops of the baseline design.

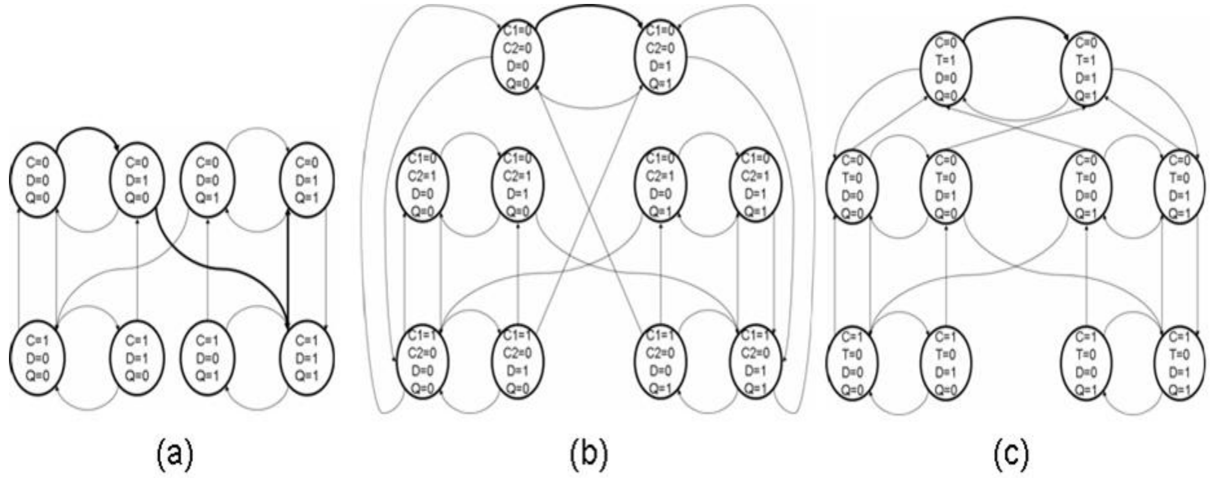

Figure 5. Canonical state transition diagrams for the baseline PPC flip-flop (a), the DCLK flip-flop (b), and the BYP flip-flop (c). Paths connected with bold lines represent the transitions required to propagate a logic 1 from D to Q.

The energy values shown in Table 1 are for single-bit flip-flops. Pipeline registers are typically composed of multiple single-bit flip-flops driven by a common clock signal. In addition to this, when a unit of work is propagated through a pipeline register, some of the bits may not change values. For the case of a single bit flip-flop, propagating a non-transitioning input signal consumes significantly less energy in the conventional case (since only the clock switches), and consumes no energy in transparent mode. Therefore, the fraction of bits switching has a large influence on the amount of energy consumed by the pipeline register. If the fraction of bits switching is small, energy consumption is dominated by clock activity. In contrast, for large fractions of switching bits, energy consumption is dominated by data activity.

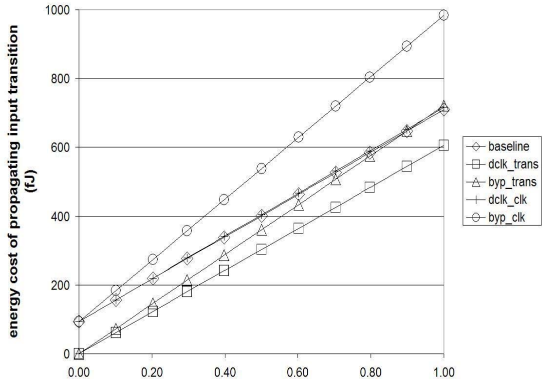

Figure 6. Energy cost of propagating input transition through 128-bit register.

In order to quantify the effect that the fraction of switching bits can have on propagation energy, a 128-bit register was evaluated using each considered design. Figure 6 presents the results of this experiment. The x-axis represents the fraction of bits that switched, while the y-axis is the propagation energy. Of the bits that are switching, half are rising, and the other half are falling. This graph illustrates how much energy can be saved propagating data items in transparent mode. It is clear from Figure 6 that while both the DCLK and BYP transparent flip-flops save energy in transparent mode, the DCLK saves more at higher switching fractions. In addition to this, at higher switching fractions the BYP flip-flop consumes significantly more energy when transitions are propagated in clocked mode.

6. Delay results

In addition to evaluating the presented designs in terms of energy consumption, it is also important to make sure that these flip-flops have delay characteristics such that timing closure can be achieved. Consider an arbitrary design pipelined into N stages. Let c_Q be the maximum clock-to-q delay for all flip-flops in the design. Further, let d_{MAX} and t_{SETUP} be the maximum delay through combinational logic in any stage, and the flip-flop setup time, respectively. With these terms defined, the lower bound on the clock period (denoted by C_{MIN}) can be expressed by Equation 1.

Table 1. Flip-flop energy consumption. The energy consumption (in fJ) is shown for propagating input transitions in both clocked and transparent modes.

Mode	Clocked				Transparent	
	D rises	D falls	D stays at zero	D stays at one	D rises	D falls
BASELINE	5.68	5.42	0.72	0.73	N/A	N/A
DCLK	5.71	5.48	0.72	0.73	4.83	4.63
BYPASS	8.47	6.91	0.73	0.73	6.70	4.57

$$C_{MIN} \geq c_Q + d_{MAX} + t_{SETUP}$$

Equation 1. Long path timing constraint.

For a design pipelined into N stages, the total delay (denoted by T_D) is given by Equation 2.

$$T_D = N(c_Q + d_{MAX} + t_{SETUP})$$

Equation 2. Total propagation delay.

The rewriting of this equation to differentiate the clock-to-q and setup delays of the first and last pipeline registers, which are composed of normal flip-flops even in a transparent pipeline (discussed by [3]), is shown in Equation 3.

$$T_D = 2(c_Q + t_{SETUP}) + N(d_{MAX}) + (N\text{-}2)(c_Q + t_{SETUP})$$

Equation 3. Total propagation delay (rewritten).

The rightmost term in Equation 3 represents the delay through the internal pipestage registers. When a transparent pipeline is in shallow mode the delays represented by this term are replaced by the transparent propagation delay, meaning simply the propagation delay through both the master and slave flip-flops in transparent mode. Let the transparent propagation delay be denoted by Tr_D. In order for timing closure to be achieved in transparent mode, the inequality shown in Equation 4 must hold.

$$Tr_D \leq (c_Q + t_{SETUP})$$

Equation 4. Requirements for timing closure in transparent mode.

In addition to this, care must be taken to ensure that timing closure can still be achieved in clocked mode. If the proposed transparent flip-designs have higher setup or clock-to-q delays, the clock period may need to be lengthened to satisfy the long path constraint. This is undesirable as it would result in performance loss. Let $\{c_{Q\_DCLK}, c_{Q\_BYP}\}$ and $\{t_{SETUP\_DCLK}, t_{SETUP\_BYP}\}$ represent the clock-to-q and setup delays for the DCLK and BYP designs, respectively. In order for a transparent pipeline to satisfy the long path requirement in clocked mode, the inequality given in Equation 5 must be satisfied.

$$c_{Q\{DCLK,BYP\}} + t_{SETUP\{DCLK,BYP\}} \leq c_Q + t_{SETUP}$$

Equation 5. Requirements for timing closure in clocked mode.

Table 2. Measured flip-flop delays. The setup, clock-to-q and transparent propagation delays are shown in picoseconds.

	t_{SETUP}	c_Q	Tr_D
BASELINE	115	139	n/a
DCLK	115	139	217
BYP	110	254	35

The measured setup, clock-to-q, and transparent propagation delays are shown in Table 2. All values shown are in picoseconds. The bypass flip-flop fails to satisfy the timing requirements in clocked mode. The resistance associated with the additional transmission gate at the output of the bypass flip-flop is responsible for increasing the clock-to-q delay, preventing the design from reaching timing closure.

In order to meet timing requirements with BYP design, the clock period would need to be extended. Because the BYP flip-flop is inferior to the DCLK flip-flop in terms of both energy consumption and delay, only the DCLK and BASELINE designs are compared in the remainder of this paper.

7. Floating point adder analysis

In order to further illustrate the utility of the proposed design, the energy consumption of the DCLK flip-flop was evaluated in the context of a transparently pipelined floating point adder datapath. The floating point adder design provided in the OpenSPARC Verilog Model was used for this experiment (to obtain accurate latch counts) [7]. The adder is pipelined into 4 stages, meaning that the 3 internal registers are composed of flip-flops with the new transparent mode. A detailed microarchitectural simulator was used to collect input switching factors and operand arrival rate statistics for several SPEC floating point benchmarks [1]. The energy consumption for each pipeline register is calculated by first determining from the arrival rate statistics how many units of work must be propagated through each register in both (transparent and clocked) modes. These propagation counts are then multiplied by the energy cost per input transition (which is dependent on the percentage of bits that switch value). The results of this experiment are shown in Figure 7. The y-axis represents the energy consumption of the datapath pipeline registers normalized to a design where all pipeline registers are composed of flip-flops of the baseline type with conventional (valid-bit based) clock gating.

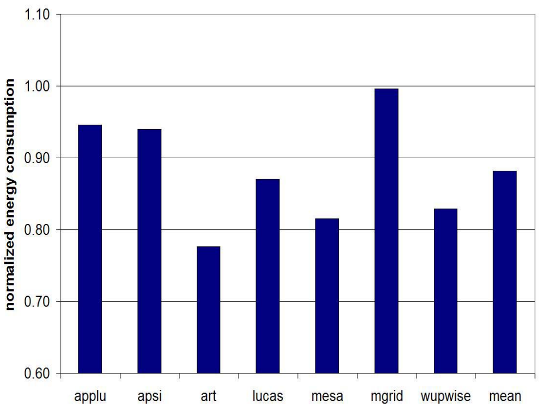

Figure 7. Floating point register energy consumption.

The amount of register energy savings achievable with transparent pipeline gating is primarily dependent on the arrival rate of instructions to the floating point unit, a characteristic that is workload dependent. Assuming the use of the floating point unit is uniformly distributed over the running time of the program, the utilization, which we define as the number of floating point operations issued to the unit divided by the simulation cycles (factoring out memory stall time) is a natural proxy for the amount of power savings achievable. The floating point utilization, along with the fraction of switching bits (used to obtain the energy costs), for each benchmark is displayed in Table 3.

Table 3. Observed switching factors and utilization.

	Switching Factor	Utilization
ammp	0.43	0.25
applu	0.19	0.40
apsi	0.23	0.10
art	0.30	0.21
lucas	0.19	0.12
mgrid	0.42	0.45
wupwise	0.18	0.14

From the results shown in Figure 7 and Table 3 it is clear that the observed energy savings do not always correlate directly with the unit utilization. For example, the benchmarks applu and apsi have similar amounts of power savings, but apsi has a utilization of 10%, while applu utilizes the floating point unit 40% of the time. The reason for this is that the issue of operations to the floating unit is not necessarily distributed evenly over time. It is possible for a workload to asymmetrically issue operations to the floating point unit such that the overall utilization is low, but the majority of operations arrive in back-to-back cycles (meaning the pipeline registers are forced to operate in clocked mode), precluding any energy savings from transparent pipeline gating.

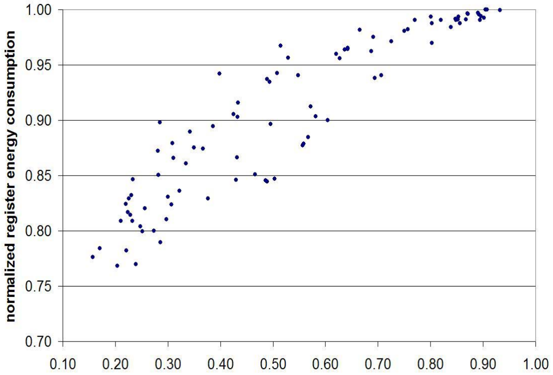

Figure 8. Results of monte carlo analysis for DCLK design.

In order to study the effects of transparent pipeline gating (with our proposed transparent flip-flop design) over a wider range of utilizations than those provided by the floating point benchmarks studied, a Monte Carlo experiment was performed. Instead of collecting arrival rate statistics from a performance simulator, randomized data arrival rates were generated to represent various utilizations. A switching factor of 25% was then assumed. The results of this experiment are shown in Figure 8. The y-axis represents the total energy consumption normalized to the baseline case where the pipeline is consumed entirely of flip-flops using the baseline design (representing a conventional pipeline with valid-bit based clock gating). From Figure 8 it is clear that while a larger amount of energy savings tend to occur at lower utilizations, it is still possible to have operations arrive at a rate such that the overall utilization is low, but the energy savings are limited. This last point is reflected in the variation in y-values (energy savings) for the lower utilizations in Figure 8.

7.1 Glitching

While the evaluation up to this point has focused exclusively on the energy consumption of pipeline registers, the additional energy consumed by glitching must also be considered in order to accurately evaluate the potential of transparent pipeline gating. Glitching is a significant contributor to the total energy consumption of combinational networks. Glitches form at the output of combinational logic gates as a result of variation in the arrival times of input signals. Without gate-level simulation, it is hard to determine whether transparent pipeline gating causes a larger or smaller number of glitches to form. It could be the case that a gate with unbalanced (in terms of delay) inputs in deep mode could have the fast input fed by a slow path and the slower input fed by a fast path, evening out the delays when stages are merged. The inverse of the previous situation can also occur.

Conventional (valid-bit based) clock gating schemes reduce the impact of glitches by preventing them from propagating outside of a single pipeline stage. This means that schemes that merge pipeline stages (like transparent pipeline gating) actually exacerbate the glitching problem because glitches are able to propagate through a larger number of gates before being blocked. With this in mind, it is important to make sure that the additional energy introduced because of glitching does not negate the energy savings attainable when transparent pipeline gating is used.

The impact of glitching on a transparent pipeline was studied by [3]. In this work, the author finds that for a 7-stage Multiple/Add-Accumulate (MAAC) unit, the introduced glitch power is less than 10% of the total clock power savings. While the amount of glitching that occurs in a logic network depends largely on the types of gates present, the MAAC unit studied in this work contained a large number of XOR gates, representing the worst case. XOR gates are especially susceptible to both the generation and propagation of glitches because any change in an input signal triggers a subsequent change in the output.

While the evaluation presented in this study does not explicitly include measurements of the energy introduced by glitching, the results presented in [3] imply that this additional power will not significantly erode the potential energy savings indicated in the previous sections.

8. Conclusions

In this paper, two transparent mode flip-flops were proposed and evaluated. Both of these designs are applicable to all generalized collapsible pipelining techniques. The principle contribution of this work is that these flip-flop designs enable collapsible pipelining techniques to be used in conjunction with single phase pipeline clocking. In addition, our results show that the decoupled clock flip-flop is superior in terms of both energy and delay.

9. References

[1] H. Cain, K. Lepak, B. Schwarz, and M. H. Lipasti. Precise and Accuate Processor Simulation. In Workshop on Computer Architecture Evaluation using Commercial Workloads.

[2] S. Heo, R. Krashinksy, and K. Asanovic. Activity Sensitive Flip-Flop and Latch Selection for Reduced Energy. In Proc. of the 2001 Conference on Advance Research in VLSI. March 2001.

[3] H. M. Jacobson. Improved clock-gating through transparent pipelining. In Proc. of the 2004 International Symposium of Low Power Electronics and Design. August 2004.

[4] H. M. Jacobson, P. Bose, Z. Hu, A. Buyuktosunoglu, V. V. Zyuban, R. Eickenmeyer, L. Eisen, J. Griswell, D. Logan, B. Sinharoy, J. M. Tendler. Stretching the Limits of Clock Gating Efficiency in Server-Class Processors. In Proc. of the 11th Annual International Symposium on High Performance Computer Architecture, February 2005.

[5] S. Kim, C. H. Ziesler, and M. C. Papaefthymiou. Fine-grain real-time reconfigurable pipelining. IBM J. Res. Dev. 47, 5-6, September 2003.

[6] J. Koppanalil, P. Ramrakhyani, S. Desai, A. Vaidyanathan, and E. Rotenberg. A case for dynamic pipeline scaling. In CASES '02: Proceedings of 2002 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, pages 1-8, New York, NY, USA, 2002. ACM Press.

[7] OpenSPARC T1 Microarchitecture Specification.

[8] Predictive Technology Model. http://www.eas.asu.edu/~ptm

[9] H. Shimada, H. Ando, and T. Shimada. Pipeline stage unification: a low-energy consumption technique for future mobile processors. In Ingrid Verbauwhede and Hyung Roh, editors, ISLPED, pages 326-329. ACM, 2003.

[10] J. E. Smith. An analysis of pipeline clocking. Technical report, University of Wisconsin, March 1990.

[11] N. Weste and D. Harris. CMOS VLSI Design: A Circuits and Systems Perspective. Addison-Wesley, 2005.

[12] V. Zyuban and P. Kogge. Transition Graph Methodology for Power Estimation. Notre Dame Univ., Notre Dame, IN, Notre Dame CSE Tech. Rep. 96-24, September 1996.