Minimizing the Energy Cost of Throughput in a Linear Pipeline by Opportunistic Time Borrowing

Mohammad Ghasemazar and Massoud Pedram
University of Southern California
Department of Electrical Engineering
Los Angeles, CA 90089 U.S.A.
{*ghasemaz,pedram*}*@usc.edu*

Abstract - In this paper, we present a technique to optimize the energy-delay product of a synchronous linear pipeline circuit with dynamic error detection and correction capability running. The technique dynamically adjusts the supply voltage level and clock frequency of the design by exploiting slacks that are present in various stages of the pipeline. The key enabler is the utilization of soft-edge flip-flops to allow time borrowing between consecutive stages of the pipeline in order to provide the timing-critical stages with more time to complete their computations resulting in lower error probability. This raises the effective throughput of the pipeline for a fixed energy consumption level, or alternatively, lowers the energy consumption for the same effective throughput. We formulate the problem of optimally selecting the transparency window sizes of the soft-edge flip-flops and the frequency level of the pipeline circuit at different voltage levels so as to optimize the energy cost of the achieved throughput. Experimental results show the efficacy of the problem formulation and solution technique.

1. INTRODUCTION

With the increase in demand for battery-operated personal computing devices and wireless communication equipment, the need for energy efficient design has increased. Increasing levels of power dissipation and the resulting thermal problems have become key limiting factors to processor performance. Due to their high utilization, pipeline data path in a modern microprocessor is a major contributor to the power consumption of the processor, and consequently, one of the main sources of heat generation on the chip [1]. Many techniques have been proposed to reduce the power consumption of a microprocessor's pipeline among which pipeline gating [1], clock gating [2, 3], and voltage scaling [4] are notable.

Dynamic voltage and frequency scaling (DVFS) technique is being used in modern microprocessors to reduce the energy consumption of an executed task while ensuring the performance requirements [5]. The key idea behind DVFS techniques is to dynamically scale the supply voltage level so as to provide "just-enough" circuit speed to process the system workload while meeting the total compute time and/or throughput constraints and, thereby, reducing the energy dissipation.

In this paper we present a technique to address the problem of simultaneously reducing the energy consumption and increasing the throughput in a synchronous linear pipeline i.e., one with the following properties: (i) processing stages are linearly connected, (ii) it performs a fixed function, and (iii) stages are separated by flip-flops which are clocked with the same CLK signal. Note that in a synchronous linear pipeline, new data may be introduced into the pipeline in each cycle (latency between data initiations is one). Our technique is based on the idea of utilizing *soft-edge flip-flops* (SEFF) for slack passing and decreasing the error rate in the pipeline stages.

Soft-edge flip-flops have a small transparency window which allows time borrowing across pipeline stages. Soft-edge flip-flops have been traditionally used for minimizing the effect of clock skew on static and dynamic circuits [6, 7]. Recently, the authors of [8] proposed an interesting approach to utilize soft-edge flip-flops in sequential circuits in order to minimize the effect of process variation on the yield. They formulated the problem of statistically aware SEFF assignment which maximizes the gain in timing yield as an integer linear program (ILP) and proposed a heuristic algorithm to solve the problem.

We describe a unified methodology for optimally selecting the transparency window of the SEFF in a linear pipeline so as to achieve the minimum energy over throughput for the whole pipeline to operate at the assigned voltage and frequency sets. Our technique can also be applied to a synchronous non-linear pipeline [18], although that falls outside the scope of the present paper.

The remainder of this paper is organized as follows. In Section 2 we provide some background on pipeline design and soft-edge flip-flops. Section 3 describes our techniques for reducing the energy consumption per data. Section 4 is dedicated to simulation results and Section 5 concludes the paper.

2. BACKGROUND

2.1 Preliminaries

A simple (synchronous) 2-stage linear pipeline circuit is depicted in Figure 1. We call the set of flip-flops that separate consecutive stages of the linear pipeline as a *FF-set*, for example, $FF_0 \dots FF_2$ are the FF-sets. Let's assume for now that the FF-sets used in this design are all conventional (hard-edge) FF's.

To guarantee the correct operation of the pipeline, the following timing constraints must be always satisfied in all stages of the pipeline:

$$t_{s,ij} \le T_{clk,j} - d_{ij} - t_{cq,(i-1)j} \qquad 1 \le i \le M, 1 \le j \le S \qquad (1)$$

$$t_{h,ij} \le \delta_{ij} + t_{cq,(i-1)j} \qquad 1 \le i \le M, 1 \le j \le S \qquad (2)$$

where d_{ij} and δ_{ij} denote the maximum and minimum delays of combinational logic in stage i under voltage setting v_j, $T_{clk,j}$ denotes the clock cycle time at the specified voltage level, $t_{s,ij}$ and $t_{h,ij}$ are the setup and hold times for the flip-flops in the i^{th} FF-set under v_j whereas $t_{cq,(i-1)j}$ denotes the clock-to-q propagation delay of the flip-flops in i-1st FF-set under v_j. M denotes the number of pipelines stages whereas S denotes the number of supply voltage levels. Clearly $T_{clk,j}$ depends on v_j. Inequality (1)

This research was sponsored in part by a grant from the National Science Foundation under award number CNS-0509564.

gives the constraint set on the maximum delays of the combinational logic and the flip-flop timing characteristics to prevent setup time violations. Conversely, inequality (2) specifies the constraint set on the minimum delay of the pipeline stages in order to prevent data race hazards. Notice that to account for the effect of clock skew, t_{skew}, we can simply add t_{skew} to the left side of inequality (1) and subtract it from the left side of inequality (2).

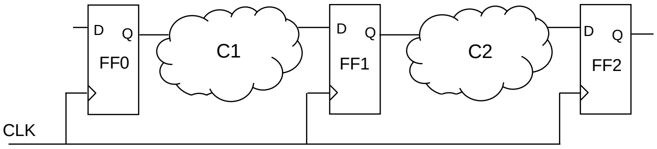

Figure 1. A simple linear pipeline.

The delay of any logic gates and interconnects are affected by process, supply voltage, and temperature (PVT) variations. For example according to [17], a 40°C junction temperature rise results in a logic and wire delay increase of roughly 5% in a 130-nm industrial process. The variation of critical path delay of a logic block may be modeled by a probability distribution function (PDF) such as a Gaussian (Normal) distribution [15] with some mean, μ, and some standard deviation, σ, as is done in [8]. Accounting for the variability of path delays in Equations (1)-(2) leads to a probability of violating the setup or hold conditions. Since the delay of combinational logic is typically much larger than the clock-to-q or setup/hold times of the input and output flip-flops for the logic, to a first order, we can ignore the effect of variability of the flip-flop timing characteristics and only focus on the effect of the variability of the combinational logic delays.

Let f_j denote the clock frequency of the pipelined circuit under a supply voltage of v_j. The probability of satisfying the setup time in pipeline stage i with voltage v_j for a given clock cycle time $T_{clk,j} = f_j^{-1}$ can be written as the probability of the *longest* path delay of the combinational logic in that stage, d_i, with a normal PDF of $N\left(\mu_{d,ij} = \mu(d_i, v_j), \sigma_{d,ij} = \sigma(d_i, v_j)\right)$, being less than the available slack time. This probability can be expressed in terms of a special function called the error function [9]:

$$P(d_{ij} < T_{Clk,j} - t_{s,ij} - t_{cq,(i-1)j}) = \frac{1}{2}\left(1 + erf\left(\frac{T_{Clk,j} - t_{s,ij} - t_{cq,(i-1)j} - \mu_{d,ij}}{\sigma_{d,ij}\sqrt{2}}\right)\right) \quad (3)$$

where the monotonically-increasing *erf* function is defined as [15]:

$$erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt \quad (4)$$

The probability of setup time constraint violation is $1-p$, where p is the value found by (3). A similar derivation can be performed for hold time constraint violation. Consequently, the probabilities of failing the setup time and hold time constraints in each stage are obtained as follows:

$$\varepsilon_{setup,ij} = \frac{1}{2}\left(1 - erf(\frac{T_{clk,j} - t_{s,ij} - t_{cq,(i-1)j} - \mu_{d,ij}}{\sigma_{d,ij}\sqrt{2}})\right) \quad (5)$$

$$\varepsilon_{hold,ij} = \frac{1}{2}\left(1 + erf(\frac{t_{h,ij} - t_{cq,(i-1)j} - \mu_{\delta,ij}}{\sigma_{\delta,ij}\sqrt{2}})\right) \quad (6)$$

where $\mu_{d,ij}$ and $\mu_{\delta,ij}$ denote the mean values of the longest path delay and the shortest path delay of the i^{th} logic stage under the j^{th} voltage setting, respectively, while $\sigma_{d,ij}$ and $\sigma_{\delta,ij}$ are the standard deviations of the corresponding longest and shortest path delay distributions.

Due to non-deterministic nature of delay variations, error detection and correction mechanisms should be added to the design to guarantee correct computation in the pipeline. We have adopted a *multi-sampling technique* in the pipeline registers [4]. In this method, a secondary latch, called *shadow latch*, is added to each flip-flop in order to sample its input on the edge of a delayed clock. Hence, the input will be sampled and stored in output latches at the triggering edges of the normal clock and the delayed clock. Contents of these latches are then compared with one another to detect any late arrival of data that causes an erroneous data to be latched and used in the main flip-flop. To correct the error, the pipeline is stalled while the content of the shadow latch is directed to the pipeline. The error correction process takes an extra time and thus decreases the throughput. We define the *effective throughput* of a linear pipeline, or simply *thruput*, as the number of valid data outputs of the pipeline per time, which is the number of clock cycle times T_{clk}:

$$thruput = \frac{\text{valid output data count}}{\text{clock cycle count} \times T_{clk}} \quad (7)$$

Assuming each error detection and correction takes γ (≥ 1) clock cycles, encountering n errors while producing N $(\geq n)$ valid data results in a throughput of:

$$thruput = \frac{Nf}{(N-n)+n\gamma} \quad (8)$$

where $f = 1/T_{clk}$ .Clearly, $\gamma^{-1}f \leq thruput \leq f$.

2.2 Soft-Edge Flip-Flops

The key idea in designing a soft-edge (master-slave) flip-flop (SEFF) [5] is to create a window during which both master and slave latches are ON, e.g., by delaying the clock of the master latch (cf. Figure 2). This window is called the *transparency window* of the SEFF and allows passing of timing slacks between adjacent pipeline stages which are separated by the flip-flops [10].

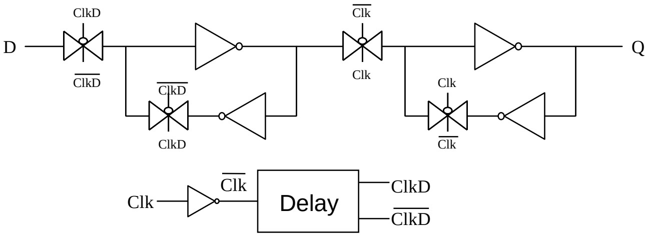

Figure 2. A positive-edge triggered soft-edge master slave flip-flop.

The delayed clock can be produced by utilizing some inverters and appropriately sizing them in order to achieve the desired window size, delay between clock of master and slave latches.

Referring back to Figure 1, for the sake of consistency with the input and output environments and to avoid imposing constraints on the sender or receiver of data for the linear pipeline circuit in question, we shall impose the *boundary condition* that the first and last FF-sets in the pipeline are composed of conventional hard-edge FF's whereas the intervening FF-sets may be SEFF's.

In a SEFF, the transparency window size is a key parameter in the timing constraints since it changes the characteristics of the flip-flop. More precisely, the setup time, hold time, and clock-to-q delay of a soft-edge flip-flop are all functions of the transparency window width. As confirmed by SEFF characterization data provided in [10], the setup time, hold time, and clock-to-q delay of a SEFF are linear functions of window size,

$$\begin{cases} t_{setup}(w) = a_1 w + a_0 \\ t_{hold}(w) = b_1 w + b_0 \\ t_{cq}(w) = c_1 w + c_0 \end{cases} \qquad (9)$$

where w is the transparency window size and a_0 through c_1 are technology and design specific coefficients.

Energy consumption per output transition of a SEFF varies with w. This is due to the fact that increasing the window size is performed by increasing the size or the number of inverters in the delayed clock path; both methods for altering w result in an *increase* in the energy consumption per output transition of the SEFF; i.e., energy consumption is a monotonically increasing function of window size. One can conclude that the energy dissipation of the SEFF may be approximated as a quadratic function of the transparency window width, i.e.,

$$E_{SEFF} = \left(d_2 w^2 + d_1 w + d_0\right) v^2 \qquad (10)$$

where d_0 through d_2 are technology and design specific coefficients and v denotes the supply voltage level.

The energy model of (10) represents the total energy consumption due to both dynamic and leakage power consumption of the SEFF circuit. As seen in Figure 3, total energy dissipation of master-slave SEFF adheres to the quadratic model presented above. The discontinuities (jumps) in the curve are due to a change in the number of inverters in the delay path.

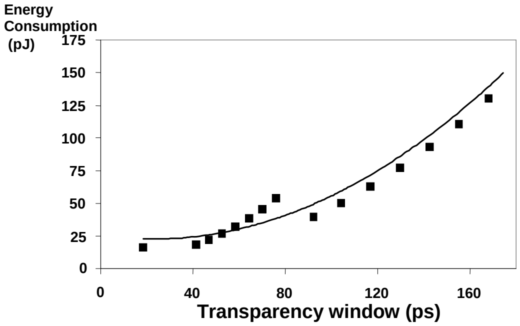

Figure 3. Energy consumption as a function of the transparency window size for a SEFF.

2.3 Soft-Edge Flip-Flops with Built-in Error Correction

As stated earlier, to enable the SEFF's to detect and correct errors, a secondary latch is added to the design of a conventional master-slave FF. This shadow re-samples the input data at a later time by utilizing a *phase-shifted global clock* signal, PS-Clk. As a result, if there is a setup time violation in the pipeline stage, comparison of these two data values would discover the error. Generally, hold time violations are fixed by adding buffers to increase the short path delays. Figure 4 shows the internal architecture of such a flip-flop. Handling error situations is discussed in detail in [4]. Here we are only interested in its effects on energy and throughput.

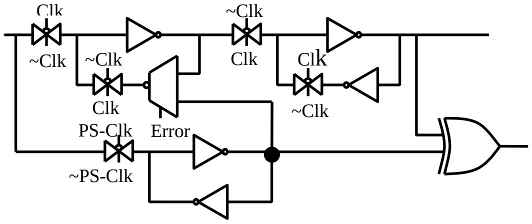

Figure 4. A positive edge SEFF with built-in error correction.

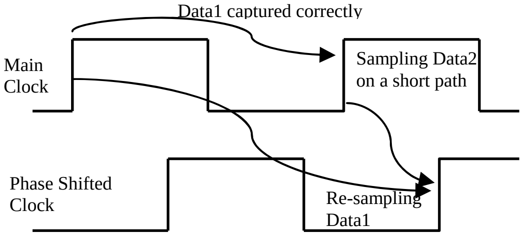

Figure 5. Timing waveforms for the SEFF.

Having introduced the phase-shifted clock signal to design, some additional timing constraints are required to be held. First, if the longest path delay of the preceding logic block is so large that it exceeds the triggering edges both the Main and PS Clock edges, then the error goes undetected. Second, there is a constraint on the minimum delay of the combinational logic circuit succeeding this flip flop, to avoid capturing short path delays as error. This is shown in the timing diagram of Figure 5. Therefore, to avoid undetected errors in the flip-flop and races due to short paths in the preceding logic, the following conditions must be met:

$$PS \geq t_{s,ij} + d_{ij} + t_{cq,(i-1)j} - T_{clk,j} \quad 1 \leq i \leq M, 1 \leq j \leq S \qquad (11)$$

$$PS \leq \delta_{ij} + t_{cq,(i-1)j} - t_{h,ij} \quad 1 \leq i \leq M, 1 \leq j \leq S \qquad (12)$$

where *PS* denotes the *phase shift* (delay) of the PS-Clk relative to the Main Clock.

3. MINIMIZING ENERGY PER THROUGHPUT

Dynamic Voltage and Frequency Scaling (DVFS) is widely used to minimize the energy consumption in microprocessors. For correct functionality, the entire pipeline should be able to complete its computations in every *circuit state* (where each such state is uniquely identified by some supply voltage and clock frequency setting which is simultaneously applied to all stages of the pipeline). Changing the voltage and frequency to bring about a new circuit state affects the combinational delay and the time budget of the combinational circuit, and hence, changes the failure rate and effective throughput of the pipeline. Of course, it also changes the energy consumption per output data value produced by the pipeline.

The key motivation for using SEFF's in a pipeline circuit is that it enables opportunistic time borrowing across adjacent stages of the pipeline in order to provide the timing-critical stages with more time to complete their computations and thereby, reduce the probability of timing errors and avoid the subsequent delay penalty of the error correction step. Consequently, the energy dissipation per produced data is reduced while maintaining the same effective throughput for the pipeline circuit with hard-edge FF's.

Higher frequency results in higher failure rate and is not a straight forward way to improve effective throughput. Also, at higher frequency larger transparent sizes are required to reduce error probabilities. As mentioned earlier, the energy consumption is an increasing function of the transparency window size and therefore higher frequency needs higher energy. It can be seen that the optimum solution of MECT balances the trade-offs between error rate and operation frequency and energy consumption.

Different circuit delay distributions under different supply voltages create the need to design the SEFF's so as to minimize the expected energy consumption over all DVS states. Further optimizing the pipeline for minimum cost in a specific voltage state may cause excessive energy consumption in another state. That is why, given the probability values for being in various voltage states during active mode of pipeline operation, we attempt to minimize the energy consumption averaged over all such states.

3.1 Soft-Edge Flip-Flop Modeling

To *optimally* select the transparency window of the SEFF's, we must accurately account not only for the effect of the transparency window on the setup/hold times and clock-to-q delay, but also on the energy consumption of SEFF's. In Section 2.2, it was shown that for a SEFF, the setup/hold times and the clock-to-q delay can be modeled as linear functions of transparency window size (c.f. equation set (9)). If the supply voltage of the flip-flop can also be adjusted to a new voltage level, v, then the coefficients of these linear models will become voltage-dependent parameters, i.e.,

$$\begin{cases} t_{setup}(w,v) = a_1(v)w + a_0(v) \\ t_{hold}(w,v) = b_1(v)w + b_0(v) \\ t_{cq}(w,v) = c_1(v)w + c_0(v) \end{cases} \qquad (13)$$

Characteristics of flip-flops can be measured by simulations to determine the coefficients in (13). Figure 6 shows SPICE simulations of the setup time as functions of transparency window size and the supply voltage level for the SEFF of Figure 2. Similarly, equation (10) can be used to approximately model the effect of adjusting the supply voltage level, v, on the SEFF energy consumption as follows:

$$E_{SEFF} = \left(d_2(v)w^2 + d_1(v)w + d_0(v)\right)v^2 \qquad (14)$$

where $d_0(v)$ through $d_2(v)$ are voltage-dependent parameters.

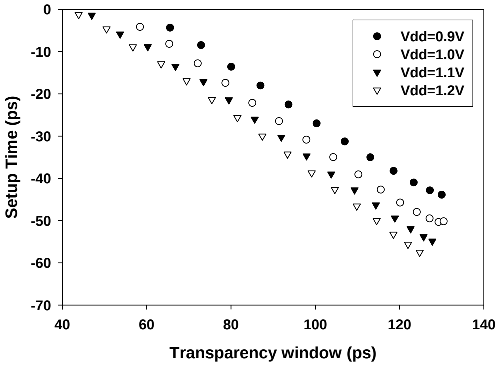

Figure 6. Setup time as a function of the supply voltage level and the transparency window width.

3.2 Problem Formulation

The problem of minimizing the energy cost of throughput (MECT) in a pipeline circuit is defined as that of finding optimal values of the operating frequency in each supply voltage state and the transparency window sizes of the individual soft-edge FF-sets in the design so as to minimize the total energy consumption per valid data produced in an linear pipeline circuit with M pipeline stages and S voltage states. In the other words, MECT tries to find a set of optimum frequencies, $f_{opt,j}$ $(j = 1,...,S)$, for each supply voltage level and a set of optimum window sizes, w_i $(i = 1,...,M)$, for each FF-set. Notice that for S circuit states and M pipeline stages, there are only $S + M - 1$ optimization variables because, in each supply voltage state, we apply the calculated optimum frequency to all stages of the pipeline while, for each soft-edge FF-set (recall that the first and last FF-sets use hard-edge FF's), we implement the calculated optimum window size at the pipeline circuit design time (which makes these size assignment independent of the voltage state of the pipeline circuit).

Given an error probability of ε_j in some circuit state s_j, the number of erroneous data computations that need to be corrected is $n = N\varepsilon_j$. From (8), the effective throughput in state s_j is:

$$thruput(s_j) = \frac{f_j}{1+(\gamma-1)\varepsilon_j} \qquad (15)$$

where f_j is the frequency setting in the circuit state, s_j. Let's assume the energy consumption of the pipeline circuit (including energy dissipations in the combinational logic and flip-flops) for detection and correction of an error is β times that of simply producing a data value without encountering an error (The value of β parameter is obtained from circuit simulations). Consequently, with an error rate of ε_j, producing N valid data values in state s_j consumes total energy of:

$$ene(s_j) = E_j\left((1-\varepsilon_j) + \varepsilon_j\beta\right)N \qquad (16)$$

E_j in equation (16) denotes the sum of energy consumptions of the combinational logic blocks and flip-flops, both of which are circuit (voltage) state dependent, when producing one data value at the output of the pipeline circuit without encountering an error. Remember from (14) that E_j is a function of the SEFF's window sizes, i.e.,

$$\begin{aligned} E_j &= \sum_{i=1}^{M} \alpha_{comb,i} C_{comb,i} v_j^{\,2} + \sum_{i=1}^{M-1} \alpha_{out,i} C_{SEFF} v_j^{\,2} \\ &= C_{sw,tot} v_j^{\,2} + \sum_{i=1}^{M-1} \alpha_{out,i} \left(d_2\left(v_j\right)w_i^{\,2} + d_1\left(v_j\right)w_i + d_0\left(v_j\right)\right)v_j^{\,2} \end{aligned}$$

Or

$$E_j = A_j + \sum_{i=1}^{M-1} B_{i,j}\left(d_{2,j}w_i^{\,2} + d_{1,j}w_i + d_{0,j}\right) \qquad (17)$$

With various E_j values denoting the average switching activity inside the combinational logic and at the inputs of the FF's in every pipeline stage. These activity factors can be calculated by circuit profiling or assumed to be typical values (e.g., in the range of 0.1 to 0.2). The point of above equation is to show that E_j is a function of various w_i's. Notice that v_j is given for each voltage state and that E_j is independent of f_j (which itself is an optimization variable).

We define $\Phi(s_j)$ as the *energy cost of throughput* for state s_j and calculate it as the ratio of (16) to (15):

$$\Phi(s_j) = \frac{E_j}{f_j}\left(1-\varepsilon_j+\varepsilon_j\gamma\right)\left(1-\varepsilon_j+\varepsilon_j\beta\right) \qquad (18)$$

Because the probability of encountering an error in a specific combinational circuit stage is independent of that in other stages, the total error rate (probability of having an error in the whole pipeline), ε_j, can be written as:

$$\varepsilon_j = 1 - \prod_{i=1}^{M} \left(1 - \varepsilon_{setup,ij}\right)\left(1 - \varepsilon_{hold,ij}\right) \tag{19}$$

where $\varepsilon_{setup,ij}$ and $\varepsilon_{hold,i}$ are probabilities of setup time and hold time violations in stage i of the pipeline under circuit state s_j, as given in equations (5) and (6).

Consider now a scenario whereby the circuit has been profiled for the various tasks that utilize it in the system and hence based on the system-level performance targets and system-level power management policy that is in effect, it has been determined that the circuit will operate in each of its voltage-frequency states according to some probability distribution. More precisely, let π_j denote the probability of being in circuit state s_j (which is characterized for a given voltage level v_j and a calculated frequency level f_j). Then, the cost function is defined as follows:

$$cost = \sum_{j=1}^{S} \pi_j \Phi(s_j) \tag{20}$$

The MECT problem is thus formulated as:

$$\begin{cases} Minimize \sum_{j=1}^{S} \pi_j \dfrac{E_j}{f_j}\left(1 - \varepsilon_j + \varepsilon_j \gamma\right)\left(1 - \varepsilon_j + \varepsilon_j \beta\right) \\ \text{such that:} \\ E_j = A_j + \sum_{i=1}^{M-1} B_{i,j}\left(d_{2,j} w_i^{\,2} + d_{1,j} w_i + d_{0,j}\right) \\ \varepsilon_j = 1 - \prod_{i=1}^{M}\left(1 - \varepsilon_{setup,ij}\right)\left(1 - \varepsilon_{hold,ij}\right) \\ 0 \le w_i \le w_{\max} \end{cases} \tag{21}$$

with $\varepsilon_{setup,ij}$ and $\varepsilon_{hold,ij}$ are calculated from equations (5) and (6), respectively. The optimization variables are w_i and f_j .

To solve (21), we run a constrained minimization solver, i.e., the MATLAB optimization toolbox. The solution gives values for each SEFF-set transparency window size and frequency values for each circuit state. The difficulty of solving (21) is the complicated form of the equation for ε_j because it contains erf(.) functions. To simplify this mathematical program, we can linearly approximate the erf(.) functions for the case of small error probabilities (i.e., $\text{for } x \le 0.2 : \frac{1}{2}(1 + erf(x)) \approx 0.12 + 0.045x$, resulting in:

$$\varepsilon_{setup,ij} \approx 0.12 - 0.045\left(\frac{\dfrac{1}{f_j} - t_{s,ij} - t_{cq,(i-1)j} - \mu_{d,ij}}{\sigma_{d,ij}\sqrt{2}}\right) \tag{22}$$

$$\varepsilon_{hold,ij} \approx 0.12 + 0.045\left(\frac{t_{h,ij} - t_{cq,(i-1)j} - \mu_{\delta,ij}}{\sigma_{\delta,ij}\sqrt{2}}\right) \tag{23}$$

With these approximations, optimization problem (21) becomes a general polynomial optimization problem which is easier to solve. Of course, when a solution is obtained we must check that the condition for approximating the *erf(x)* holds, but this has always been the case in our experimental results.

In ASIC design, the transparency window of SEFF's cannot assume any arbitrary value; instead, they only take some distinct values as indicated by the SEFF's which have been included in the cell library. Therefore, we round the continuous sizing solution to the closest match in the library.

3.3 Bounding the Probability of Undetected Errors

Having an undetected error in the pipeline is caused by very long path that violates (10). This error probability is the probability of data arriving after T_{clk}+PS which is calculated by (24) – notice that this equation is similar to (5) except that we have replaced T_{clk} with T_{clk}+PS because an undetected error occurs only when the arrival time of the correct data is later than the triggering edge of the PS Clock in the current cycle. Consequently, the probability of an undetected error is:

$$\varepsilon_{\text{undetected},ij} = \frac{1}{2}\left(1 - erf\Big(\frac{T_{clk,j} + PS - t_{s,ij} - t_{cq,(i-1)j} - \mu_{d,ij}}{\sigma_{d,ij}\sqrt{2}}\Big)\right) \tag{24}$$

$\varepsilon_{\text{undetected,ij}}$ is pipeline stage (i) and circuit state (v_j) dependent. The overall rate of undetected errors for all circuit states is:

$$\varepsilon_{\text{undetected}} = 1 - \prod_{j=1}^{S}\left(1 - \prod_{i=1}^{M}\left(1 - \varepsilon_{\text{undetected},ij}\right)\right) \tag{25}$$

To impose an upper bound on the undetected-error probability, we include PS as a new variable of optimization to the MECT problem formulation with the constraint $\varepsilon_{un\det ected} \le \varepsilon_{UB}$ where ε_{UB} is user provided (typically set to a small value such as 1*e*-10).

4. EXPERIMENTAL RESULTS

To solve the mathematical problem developed in this paper, MATLAB optimization toolbox [11] has been used. To extract the parameters used in the optimization problem, we performed transistor-level simulations on soft-edge flip-flops by using HSPICE [12]. The technology used in this simulation is a 65nm predictive technology model [13] and the nominal supply voltage of this technology is 1.2V. All simulations have been done at the die temperature of 100^{o}C.

We synthesized a number of linear pipeline (datapath) circuits and some modified ISCAS89s benchmark circuits to construct a set of benchmarks. SIS and Design Compiler packages [14] were used to synthesis the set of benchmarks.

We assumed 3 voltage levels: (0.8V, 1V, 1.2V); We then performed timing simulations to extract the mean and standard deviation of longest and shortest path delays of each pipeline stage under each voltage setting. Next we set up and solved problem (21) for hard-edge FF's (i.e., we determined the energy-delay-optimum frequency levels for each voltage setting while fixing the transparency window sizes to zero everywhere). We calculated the optimum value of Energy per throughput for a uniform probability distribution function for all voltage states (π_j=1/3 for all j). This data served as our baseline data. Next we introduced SEFF's in intermediate pipeline stages and solved problem (21) this time for the optimum f_j and w_i values. Finally we calculated the optimum value of the energy per throughput for this case and report the relative savings of our approach with the SEFF compared to that with conventional FF. Note however that even our baseline data is obtained based on the formulation that we have presented and solved in this paper, that is, the hard edge FFs also uses the error correction mechanism that the SEFFs use. If we compare MECT results with the case of using hard edge FFs without any error correction mechanism (where we end up with a conservative setting of clock frequencies in order to keep the probability of error low, say below 1*e*-10), then the percentage reduction in energy per throughput figure of merit will be on average **22%**. Furthermore, the runtime of MECT for all test cases was less than 2.2seconds on a 2.8GHz Xeon processor with 2GByte of memory.

Our baseline design implements the Error Correction mechanism and thus has an area overhead for error detection and correction logic and the DLL required for generation of Phase-Shifted Clock compared to a normal linear pipeline. The area overhead of our pipeline compared to this baseline design is only due to the more complex internal circuitry of the SEFF's which produces the transparency window. This extra overhead is negligible compared to the size of the rest of circuit.

The specifications of used benchmarks are shown in Table 1. The first entry in this table shows the name of the benchmark. The second entry shows the operating voltage of the circuit. The Third and fourth entries are the distribution of maximum and minimum delays of each pipeline stage at the nominal voltage.

TABLE 1. SPECIFICATION OF BENCHMARKS

Bench-mark	Voltage	Max Delay (ps)	Min Delay (ps)

TB1	V=0.8V	N(987,42), N(961,40),
		N(1070,48), N(869,38)
			N(497,28), N(837,40)
			N(582,33), N(746,31)

	V=1.0V	N(538,23), N(523,23)
		N(586,26), N(477,18)
			N(272,16), N(455,22)
			N(318,18), N(419,17)

	V=1.2V	N(350,15), N(342,15)
		N(382,17), N(310,12)
			N(176,10), N(290,14)
			N(206,12), N(270,11)

TB2	V=0.8V	N(1152,52),N(1113,54), N(1470,56),N(1279,42), N(1180,45), N(1109,39)	N(597,28), N(841,33), N(900,42),N(989,31), N(905,36), N(849,28)

	V=1.0V	N(638,29), N(613,29)
		N(806,29), N(697,23),
		N(640,20), N(609,21)
			N(328,15), N(463,18)
			N(496,23), N(537,16),
			N(490,20), N(469,15)

	V=1.2V	N(410,19), N(400,19)
		N(522,20), N(446,16),
		N(420,16), N(389,14)
			N(210,10), N(302,12)
			N(322,15), N(350,12),
			N(320,13), N(303,10)

Table 2 reports the simulation results. The first entry denotes the circuit; the second column shows the maximum frequency determined by baseline for the test supply voltages. Next column shows the frequencies obtained by MECT, and finally the last column shows the percentage of improvement in the cost function of equation (20) that can be achieved by MECT.

TABLE 2. MECT PERFORMANCE COMPARISON WITH A DESIGN COMPRISED OF HARD EDGE FF EQUIPPED WITH ERROR CORRECTION

Benchmark	fmax (MHz)		Energy/Thruput Reduction %
	baseline	MECT	
TB1	1.48	1.56	5.5%
	1.75	1.83	5.3%
	2.00	2.11	5.2%
TB2	1.1	1.22	8.2%
	1.34	1.45	8.4%
	1.57	1.63	8.0%

5. CONCLUSION

We presented a new technique to minimize the total energy consumption of a linear pipeline circuit by utilizing soft-edge flip-flops. We formulated the problem as a mathematical program and solved it efficiently. Our experimental results demonstrated that this technique is quite effective in reducing the energy of a pipeline circuit under a performance constraint.

A number of extensions based on the work presented in this section are possible to further reduce the energy dissipation of a pipeline circuit. These include (i) Consider the interdependency between setup and hold times. It is known that the "independent" characterization of setup, hold time, and clock-to-q delay of FF's results in pessimistic timing analysis [16]. In our problem definition, considering the interdependency between the setup and hold time provides more freedom in the optimization problem and it is expected that it improves the quality of results. (ii) Solving the MECT design problem for the non-linear pipelines with multi-stage feed forward and feedback paths. The problem setup in this case will be similar to that presented in Section 3 but the constraints will be more complex.

References

[1] S. Manne, A. Klauser, and D. Grunwald, "Pipeline gating: speculation control for energy reduction," in *Proc. Int'l Symposium on Computer Architecture*, 1998, pp. 132-141.

[2] H. M. Jacobson, "Improved clock-gating through transparent pipelining," in *Proc. Int'l Symposium on Low Power Electronics and Design*, 2004, pp. 26-31.

[3] H. Jacobson, P. Bose, H. Zhigang*, et al.*, "Stretching the limits of clock-gating efficiency in server-class processors," in *Proc. High-Performance Computer Architecture*, 2005, pp. 238-242.

[4] D. Ernst, N. Kim, S. Das*, et al.*, "Razor: a low-power pipeline based on circuit-level timing speculation," in *Proc. Int'l Symposium on Microarchitecture*, 2003, pp. 7-18.

[5] K. Choi, R. Soma, and M. Pedram, "Fine-grained dynamic voltage and frequency scaling for precise energy and performance trade-off based on the ratio of off-chip access to on-chip computation times." *IEEE Trans. on Computer Aided Design,* Vol. 24, No. 1, Jan. 2005, pp.18-28

[6] H. Partovi, R. Burd, U. Salim*, et al.*, "Flow-through latch and edge-triggered flip-flop hybrid elements," in *Proc. Int'l Solid-State Circuits Conference*, 1996, pp. 138-139.

[7] D. Harris and M. A. Horowitz, "Skew-tolerant domino circuits," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 11, Nov. 1997, pp. 1702-1711.

[8] V. Joshi, D. Blaauw, and D. Sylvester, "Soft-edge flip-flops for improved timing yield: design and optimization," in *Proc. Int'l Conference on Computer-Aided Design*, 2007, pp. 667-673.

[9] A. Papoulis, S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill 2002

[10] M. Ghasemazar, B. Amelifard, and M. Pedram, "A Mathematical Solution to Power Optimal Pipeline Design by Utilizing Soft Edge Flip-Flops," in *Proc. Int'l Symposium on Low Power Electronics and Design*, 2008.

[11] MATLAB Optimization Software, http://www.mathworks.com [online]

[12] HSPICE: The gold standard for accurate circuit simulation, http://www.synopsys.com/products/mixedsignal/hspice/hspice.html [online]

[13] Predictive Technology Model, http://www.eas.asu.edu/~ptm/ [online]

[14] E. M. Sentovich, K. J. Singh, L. Lavagno*, et al.*, "SIS: A System for Sequential Circuit Synthesis," University of California, Berkeley, Report M92/41, May 1992.

[15] Milton Abramowitz and Irene A. Stegun, eds. Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. New York: Dover, 1972

[16] E. Salman, A. Dasdan, F. Taraporevala*, et al.*, "Exploiting setup–hold time interdependence in static timing analysis," *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 26, no. 6, Jun. 2007, pp. 1114-1125.

[17] T. Sato, J. Ichimiya, N. Ono, K. Hachiya, and M. Hashimoto, "On-Chip Thermal Gradient Analysis and Temperature Flattening for SoC Design," *IEICE Trans. Fundamentals*, Vol. E88-A, No. 12, Dec. 2005, pp. 3382-3389.

[18] K. Hwang, *Advanced Computer Architecture*. New York, NY: McGraw Hill, 1993.