Proc. Int. Conf. of Computer Aided Design (ICCAD-2005), Nov. 2005, pp. 535-540.

Total Power-Optimal Pipelining and Parallel Processing under Process Variations in Nanometer Technology

Nam Sung Kim[1], Taeho Kgil[2], Keith Bowman[1], Vivek De[1], and Trevor Mudge[2]
[1] Intel Corporation, Hillsboro, Oregon, {nam.sung.kim, keith.a.bowman, vivek.de} @intel.com
[2] University of Michigan, EECS Department, Ann Arbor, MI 48109 {tkgil, tnm} @eecs.umich.edu

Abstract

This paper explores the effectiveness of the simultaneous application of pipelining and parallel processing as a total power (static plus dynamic) reduction technique in digital systems. Previous studies have been limited to either pipelining or parallel processing, but both techniques can be used together to reduce supply voltage at a fixed throughput point. According to our first-order analyses, there exist optimal combinations of pipelining depth and parallel processing width to minimize total power consumption. We show that the leakage power from both subthreshold and gate-oxide tunneling plays a significant role in determining the optimal combination of pipelining depth and parallel processing width. Our experiments are conducted with timing information derived from a 65nm technology and fanout-of-four (FO4) inverter chains. The experiments show that the optimal combinations of both pipelining and parallel processing—8~12×FO4 logic depth pipelining with 2~3-wide parallel processing—can reduce the total power by as much as 40% compared to an optimal system using only pipelining or parallel processing alone. We extend our study to show how process parameter variations—an increasingly important factor in nanometer technologies—affects these results. Our analyses reveal that the variations shift the optimal points to shallower pipelining and narrower parallel processing—12×FO4 logic depth with 2-wide parallel processing—at a fixed yield point.

1. Introduction

Traditionally, two architectural techniques have been used to improve the performance of processing systems at a fixed voltage—increasing pipelining depth or parallel processing width. Pipelining reduces the number of logic levels between timing elements to increase operating frequency, while parallel processing increases the number of parallel logic instances to increase throughput at a fixed frequency. However, the operating frequency or throughput increase offered by these techniques is often used instead to reduce power (or supply voltage) while maintaining a fixed throughput. Supply voltage scaling is one of the most effective techniques for trading timing slack for power, and it leads to a quadratic reduction in switching power and a super-linear reduction in leakage power as leakage current has a very strong dependency on supply voltage in nanometer technologies.

Pipelining and parallel processing impact power and performance in quite different ways. Deeper pipelining increases timing as well as both switching and leakage power overheads by the increased number of timing elements (latches or flip-flops). The power overhead from the timing elements grows super-linearly with the pipelining depth, because the number of timing elements increases super-linearly with the depth [1]. On the other hand, wider parallel processing only increases leakage power overhead from parallel copies of the logic. Due to the scaled frequency proportional to parallel processing width, switching power of the system remains constant at a fixed voltage (assuming perfectly parallelizable computation) but the leakage power overhead from the parallel copies of logic grows linearly with the width. This becomes important and because both the subthreshold leakage and gate-oxide tunneling leakage power consume a substantial amount of total power in nanometer technologies.

As the transistor feature size becomes smaller, circuits show an increased sensitivity to the fluctuations of process parameters such as threshold voltage V_{th}, channel length L_{eff}, and oxide thickness t_{ox}. Understanding of the effects of those variations become more important in the design of high performance digital system [2], because these significantly affect the critical path delay or the maximum operating frequency. As pipelining and parallel processing become deeper and wider, the critical path delay variations become even more significant, because these increase the number of the critical paths. Furthermore, as the pipelining depth—inversely proportional to the logic depth—increases, the critical path delays become more sensitive to the random variations of the processor parameters [3]. These two factors make it more difficult to achieve a target operating frequency at a target yield point and accordingly they will limit both the pipelining depth and the parallel processing width.

The use of pipelining for power reduction was proposed by Chandrakasan et al. [4]. Several microarchitectural level studies have examined optimal pipelining depth for both performance and power in uni-processors [6-10]. In addition, there were several studies comparing pipelining with parallel processing to determine which technique is more efficient for minimizing total switching power at a functional-block level [11-12]. These previous studies focus on either pipelining depth at a fixed parallel processing width or parallel processing width at a fixed pipelining depth. The greater number of degrees of freedom that results from the simultaneous optimization of both variables—pipelining depth and parallel processing width—allows for lower power solutions as we will see. Furthermore, the inclusion of leakage power and process parameter variations introduce further important dimensions to this optimization problem.

In our study, first, we present the first-order performance and power analysis models. Second, to optimize total power consumption, we explore switching and leakage power trade-offs of various pipelining depths and parallel processing widths based on a future nanometer technology. In this investigation, we assume a fixed-throughput design having idealized unlimited parallelism. This almost never occurs in real applications, but is approximated by the workload of some digital signal processors, network processors, and graphics engines. In this case, we do not include any performance loss from pipeline stalls as pipelining depth or parallel processing width increases. Third, we extend our analyses to provide the impact on the optimal depth and width by limited parallelism. Finally, we investigate the impact on the optimization by process parameter variations.

The remainder of this paper is organized as follows. Section 2 explains simulation methodology used in this study. Section 3 presents supply voltage scaling analyses for given pipelining depth and parallel processing width for idealized applications. Section 4 analyzes switching, leakage, and total power trade-offs from pipelining and parallel processing based on the results presented in Section 3. Section 5 and 6 analyze the implication of limited parallelism and process parameter variations on total power optimal pipelining and

0-7803-9254-X/05/$20.00 ©2005 IEEE.

parallel processing. Finally, Section 7 summarizes our contributions.

2. Methodology

To model logic path in a design, we use a simple static inverter chain with each inverter driving four copies of itself to yield a FO4 load. FO4 is a common metric for the first-order analysis and evaluation of digital circuit performance in given process technologies [5]. We use 24×FO4 delays as a baseline computational element logic depth to represent a current high-performance processor circuit; the high-frequency Pentium-4 has a 20×FO4 cycle time [6], and most other designs have somewhat shallower pipelines. Even though different circuit styles and logic gates might lead to different power-optimal pipelining depth and parallel processing width, we assume that our FO4 inverter chain model is fairly representative for *first-order* analyses. The insights gathered from the simulation results can be applied to other cases. For HSPICE simulations, a 65*nm* technology model is used.

Global wire delay does not scale as fast as gate delay in scaled technology. However, one or more pipeline stages—determined by the clock cycle time and the delay of the interconnect—are assigned to long global communication wires such as the buses between L1 and L2 caches in modern microprocessors. Furthermore, the drivers for the global wires have the same structure as the inverter chains. Hence, we assume those global wires as separate pipelining stages can be represented by the FO4 delay model, too. In considering logic delay, we do not include local wire capacitance effect, because we can only guess at the details of circuit layouts. Instead, we assume that the local wire delay effects can be embedded in the FO4 delay number [5]. Nevertheless, we are aware that the local wires increase switching power. To resolve this issue, we make an assumption about the ratio between switching and leakage power of the entire system and leave it as a parameter in this study. Finally, we leave the impact of glitching power for a future study. We do not expect it to change our results significantly, because its impact is a diminishing one that reduces linearly with pipelining depth.

3. Supply Voltage Scaling Trends

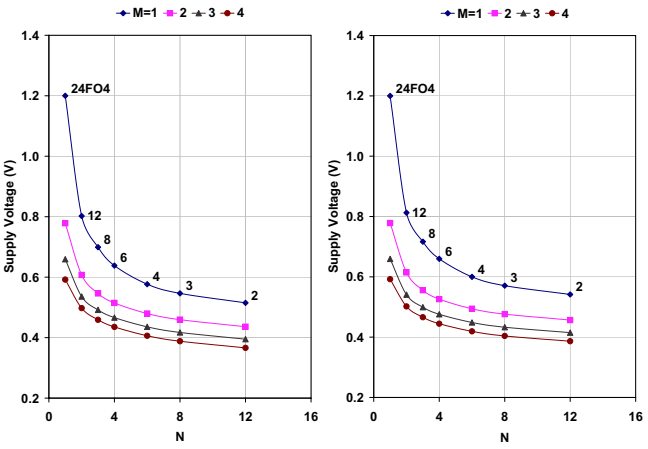
In this section, we present supply voltage scaling trends as a function of logic depth in each pipeline stage and parallel processing width. For our analyses in this section, we assume that applications are *embarrassingly parallel*—the term widely used in the parallel computing area when computations can be highly parallelized—and can be pipelined and parallelized without any inefficiencies due to data or control dependencies. In other words, the performance of applications scales linearly with the width of parallel processing and increasing pipelining depth does not incur any performance penalty except from the increasing overhead of timing elements. We will discuss the effects by the limited parallelism in Section 5.

In general, the delay T of a logic circuit as a function of supply voltage can be approximated by:

$$T(V) = A \cdot \frac{V}{(V - V_{th})^{\alpha}} \quad (1)$$

where A is the constant proportional to transistor sizes, load capacitance, etc., α is the velocity saturation effect factor, V is the supply voltage, and V_{th} is the threshold voltage. To maintain the same throughput (or computation time) as the baseline processor, the following should be satisfied:

Figure 1: Supply voltage scaling trends for two different flop timing overheads at the same throughput (M—parallel processing width and N—24×FO4 logic depth division factor).



(a) $T_{flop}(V)$ = 2×FO4 (V) (b) $T_{flop}(V)$ = 3×FO4 (V)

$$L \cdot (T_{logic}(V_{DD}) + T_{flop}(V_{DD})) \quad (2)$$

$$= \underbrace{(T_{logic}(V) + N \cdot T_{flop}(V))}_{\text{latency for the first } M \text{ operations}} + \left(\frac{L-M}{M} \cdot \left(\frac{T_{logic}(V)}{N} + T_{flop}(V)\right)\right)$$

$$= \left(\frac{L}{M} + N - 1\right) \cdot \left(\frac{T_{logic}(V)}{N} + T_{flop}(V)\right)$$

where is V_{DD} is the initial unscaled supply voltage, and V is the scaled version. The terms $T_{logic}(V)$ and $T_{flop}(V)$ are the delays of each baseline pipelining stage for *logic* and *flip-flops* at the scaled (lower) supply voltage *V*. $T_{flop}(V)$ includes *clock skew* and *jitter* overheads. The term L is the number of operations required to execute a benchmark to completion, M is the width of parallel processing, and N is a measure of the depth of pipelining. N does not represent the actual number of pipelining stages, but a logic depth division factor. For example, we divide each 24×FO4 pipelining stage into two 12×FO4-delay stages when N is equal to *2*, i.e., the number of total pipelining stages are doubled. For $L/M >> N-1$, a typical situation, (2) can be approximated by:

$$T_{logic}(V_{DD}) + T_{flop}(V_{DD}) \approx \frac{1}{M} \cdot \left(\frac{T_{logic}(V)}{N} + T_{flop}(V)\right) \quad (3)$$

Figure 1 shows the supply voltage scaling trends of *M-N* processors denoting *M*-wide parallel processing processors with *24×FO4/N* logic depth per stage. The scaled supply voltage is calculated at each M and N combination point using (3) at the same throughput point as the *1-1* based processor operating with V_{DD}= 1.2V. To analyze the impact of the timing element overhead, we calculate the scaled supply voltages for two different timing overhead models—T_{flop}(V) is 2×FO4 (*V*) [7] in Figure 1-(a) and 3×FO4(*V*) in Figure 1-(b), respectively. As can be seen in Figure 1, pipelining alone is effective in scaling the supply voltage, but the processors employing the combination of pipelining and parallel processing are far more effective than pipelining alone. On the other hand, deeper pipelines (or shallow logic depth per stage) becomes less effective in reducing supply voltage as the timing overhead increases. Likewise, due to a dramatic increase in logic circuit

delay at low voltage, excessive parallel processing is not effective, either. The effectiveness of supply voltage scaling diminishes when parallel processing is increased significantly. Hence, a balanced M and N is required to allow supply voltage scaling to maximize total power reduction.

4. Power Scaling Trends

4.1 Switching Power

Switching power remains a significant component of total power consumption, even in leaky nanometer technology. Switching power is the power consumed while charging and discharging load capacitances. The load capacitances include transistor parasitic and wire capacitances. When f is the operating frequency of the baseline 1-1 processor at V_{DD}, the switching power of an M-N processor can be expressed as:

$$P_{switching}^{M-N} = M \cdot \frac{f}{M} \cdot (C_{logic} + C_{flop} \cdot N^{\rho}) \cdot V^2 = M \cdot \frac{f}{M} \cdot C_{logic} \cdot (1 + B \cdot N^{\rho}) \cdot V^2 \quad (4)$$

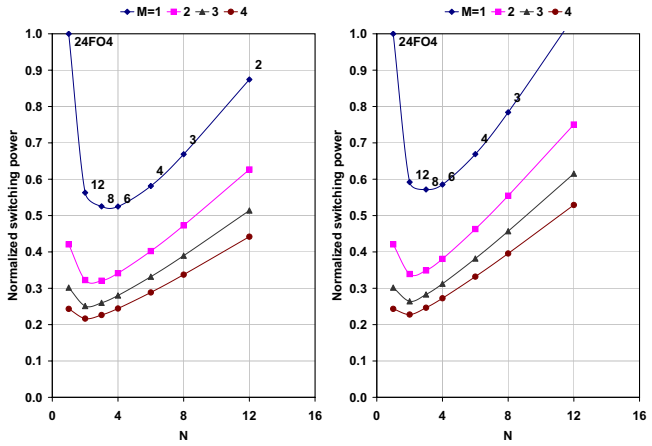
where V is the supply voltage for the M-N processor at the same throughput point as the baseline 24FO4 logic depth processor, C_{logic} and C_{flop} are the effective switching capacitances of the logic and the timing elements of the baseline *1-1* processor, B is the timing element switching power overhead factor defined as C_{flop}/C_{logic}, and ρ is the flip-flop growth factor that is super-linearly proportional to the pipelining depth N. We assume that the logic switching power includes all the local and global interconnect wire capacitance and the timing element switching power includes the clocking power by switching the input capacitances of the timing elements and the clock tree power. If the number of timing elements grows linearly with the number of stages, $\rho = 1$. However, Srinivasan, et al. [1] have argued that the number of timing elements in pipelining stages grows super-linearly with the pipelining depth—consider pipelining a multiplier or a multi-banked memory. Following them, we take $\rho = 1.2$ throughout this study, but consider ρ as a design-specific parameter. Note that increasing the parallel processing width does not increase switching power with M at a fixed supply voltage, because the operating frequency f is also scaled down with M as noted in (4).

The normalized switching power of M-N processors to the baseline *1-1* processor—our main interest of this study—can be represented as:

$$\frac{P_{switching}^{M-N}}{P_{switching}^{1-1}} = \frac{M \cdot \frac{f}{M} \cdot C_{logic} \cdot (1 + B \cdot N^{\rho}) \cdot V^2}{1 \cdot \frac{f}{1} \cdot C_{logic} \cdot (1 + B \cdot 1^{\rho}) \cdot V_{DD}^{\ 2}} = \frac{(1 + B \cdot N^{\rho}) \cdot V^2}{(1 + B) \cdot V_{DD}^{\ 2}} \quad (5)$$

The parameters such as f and C_{logic} are canceled out during the normalization process. Hence, we do not need to estimate the C_{logic} of the system. Instead, we only have to focus on the ratio between C_{logic} and C_{flop} as an analysis parameter whose trend is relatively well known from earlier designs. Figure 2 shows the normalized switching power of M-N processors at the same throughput. To analyze the total switching power impact of the timing elements, we also calculated the normalized switching power for two different timing element switching power models. Specifically, we use 0.25 and 0.33 for B (C_{flop}/C_{logic}) in Figure 2-(a) and (b), respectively—C_{flop}/C_{logic} is typically 0.2~0.3 in general-purpose microprocessors according to [13]. The calculated results in Figure 2 are based on the supply voltage scaling calculations from Section 3. As noted in Figure 2, the optimal pipelining logic depth for the *1-N* processor switching power is between 6 and 8×FO4 depending on the timing element power overhead. This agrees with [7] and [10], but increasing M shifts the optimal point to a deeper logic depth from 6 to 8 and from 8 to 12×FO4. Furthermore, increasing the power overhead by the timing elements also shifts the optimal point to a deeper logic depth (smaller N). Compared to the optimally pipelined 1-wide processor, the 2-, 3-, and 4-wide parallel processing processors at the optimal pipelining depth consume less switching power by 39/40%, 52/55%, and 59/61% when B is 0.25/0.33. The switching power reduction, however, diminishes quickly as the parallel processing width exceeds 2.

Figure 2: Normalized switching power for two different flop switching power overheads at the same through-put point.



In summary, more parallelism at an optimal logic depth helps to reduce switching power dramatically, but the power reduction caused by increasing M diminishes, because the supply voltage scaling diminishes as explained in Section 3. While switching power scales with M, leakage power does not. Hence, the increased leakage power multiplied by M will start to dominate the total power of the system considering that the leakage power consumes a significant portion of total chip power in the current and future technologies.

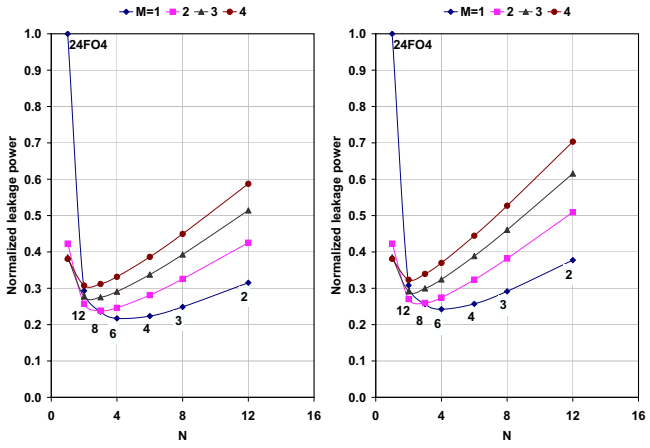
4.2 Leakage Power

The rapid reduction in gate length, oxide thickness, and accompanying down-scaled threshold voltages over the last few process generations has led to an exponential growth in both subthreshold and gate-oxide tunneling leakage power. Within a few process technology generations, it is predicted that power dissipation from both leakage current could be comparable to dynamic switching power. The leakage power of our pipeline circuit can be given by:

$$P_{leakage}^{M-N} = M \cdot (I_{logic} + I_{flops} \cdot N^{\rho}) \cdot V = M \cdot I_{logic} \cdot (1 + C \cdot N^{\rho}) \cdot V \quad (6)$$

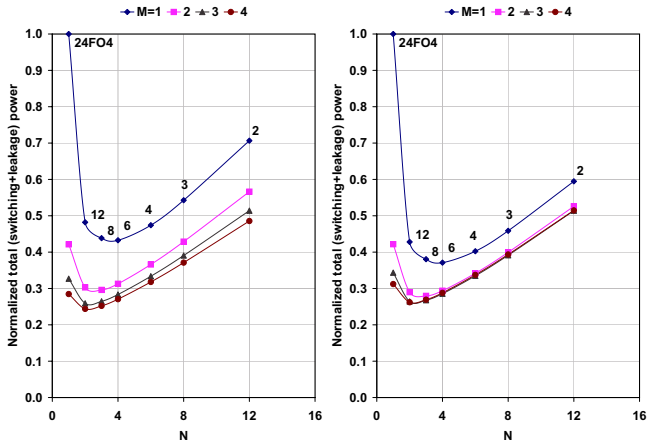
where I_{logic} and I_{flops} are the leakage current from the logic and the timing elements at supply voltage V, and C is the timing element leakage power overhead factor defined as I_{flops}/I_{logic}. As a result of subthreshold and gate-oxide tunneling leakage both I_{logic} and I_{flops} have super-linear dependency on

Figure 3: Normalized leakage power for two different flop leakage power overheads at the same throughput point.



(a) C (I_{flops}/I_{logic})= 0.25 (b) C (I_{flops}/I_{logic})=0.33

Figure 4: Normalized total power for two different leakage over switching power ratios at the same throughput point (B/C=0.25).



(a) G ($P_{leakage}/P_{switching}$)= 0.3 (b) G ($P_{leakage}/P_{switching}$)= 0.5

V. The subthreshold leakage current component can be modeled as [14]:

$$I_{sub} = D \cdot e^{\frac{V_{th} - \eta V}{n \cdot (k \cdot T/q)}} \cdot \left(1 - e^{\frac{V}{k \cdot T/q}}\right) \quad (7)$$

where D is a constant proportional to transistor dimensions etc., V_{th} is the threshold voltage, n is the subthreshold swing factor, kT/q is the thermal voltage proportional to the temperature, and η is the *drain-induced barrier-lowering* (DIBL) factor. The gate-oxide tunneling leakage current component can be approximated by [15]:

$$I_{gate} = E \cdot \left(\frac{V}{t_{ox}}\right)^2 \cdot e^{\left(-F\frac{V}{t_{ox}} \cdot \left(1 - \left(1 - \frac{V}{\phi_{ox}}\right)^{3/2}\right)\right)} \quad (8)$$

where E and F are physical parameters that are intrinsic to the process, ϕ_{ox} is the barrier height for the tunneling particle (electron or hole), and t_{ox} is the oxide thickness. As can be seen in (6), (7), and (8), both the subthreshold and gate-oxide tunneling leakage power is reduced super-linearly by supply voltage scaling.

In the same way as (4), the normalized leakage power of M-N processors to the baseline *1-1* processor is represented as $M \cdot (1+C \cdot N^{\rho}) \cdot V / (1+C) \cdot V_{DD}$. Figure 3 shows the normalized (subthreshold + gate oxide tunneling) leakage power of M-N processors at the same throughput and 110°C. To analyze the total leakage power impact of the timing elements, we also calculated the normalized leakage power for two different timing element leakage power models: 0.25 and 0.33 are used for C (I_{flop}/I_{logic}) in Figure 3-(a) and (b), respectively. As C increases, the optimal leakage power point shifts to from 6 to 8 and from 8 to 12×FO4, and wider parallel processing consumes more total leakage power. As we showed in Section 3, the supply voltage scaling diminishes with M but wider parallel processing does not reduce the total leakage power, because it is multiplied by M.

4.3 Total Power

As we saw in Section 4.1 and Section 4.2, parallel processing is more effective at decreasing switching power than pipelining, while pipelining is more effective at reducing leakage power. Between these two trade-offs an optimal M and N combination can minimize the total power consumption. The total power of a multi-processor can be expressed by:

$$P_{total} = P_{switching} + P_{leakage} = P_{switching} \cdot (1 + G) \quad (9)$$

where G is the ratio between leakage and switching power.

Figure 4 shows the normalized total power of M-N processors at the same throughput. To analyze the total power impact trend of leakage power, we calculated the normalized leakage power for two different leakage power ratio models. The values 0.3 and 0.5 are used for G ($P_{leakage}/P_{switching}$) in Figure 4-(a) and (b) respectively—the fraction of leakage power is around 0.4 in the current generation high-end microprocessor [13]. The different G can account for the effects by different temperatures or different activity factors for switching power on the optimization. Wider parallel processing shifts the optimal pipelining depth from 6 to 12×FO4. As the fraction of leakage power increases, wider parallel processing than 2 does not reduce the total power noticeably. Hence, 2-wide parallel processing with 8~12×FO4 pipelining depth seems to be an optimal point considering both total power and area. This configuration reduces the total power remarkably —32%/25% when G is 0.3/0.5—compared to the 1-wide parallel processing optimal case. Compared to the system only employing either the 1-wide optimal pipelining of 6~8×FO4 logic depth or the 2-wide pipelining of 24×FO4 baseline logic depth alone, the optimal combination of pipelining and parallel processing can reduce total power by 20~44% depending on the parameter values such as B (0.25~0.33), C (0.25~0.33), and G (0.3~0.5). Furthermore, the optimal logic depth for switching power is very close to that for leakage as seen in Figure 2 and Figure 3. Hence, increasing G — the fraction of leakage power does not change the optimal logic depth of every M we investigated, but 1-wide parallel processing with 6~8×FO4 becomes total optimal if G is over 0.8.

5. Implications of Limited Parallelism

In Section 3 and Section 4, we assumed that the throughput of processors scales linearly with both the depth of pipelining and the width of parallel processing with very high parallelism (e.g., applications such as streaming, networking, and graphics processors). However, in fact, it does not scale very well for many other applications because of limited parallelism, data dependencies, and branch misprediction penalties in

general purpose processors. In this section, we investigate the impact of such limited parallelism on total power reduction. To reflect the throughput impact by increasing both the width of parallel processing and the depth of parallel processing, the equation (3) is revised to:

$$CPI(1,1)\cdot(T_{logic}(V_{DD})+T_{flop}(V_{DD})) = \frac{M\cdot CPI(M,N)}{M}\cdot\left(\frac{T_{logic}(V)}{N}+T_{flop}(V)\right) \quad (10)$$

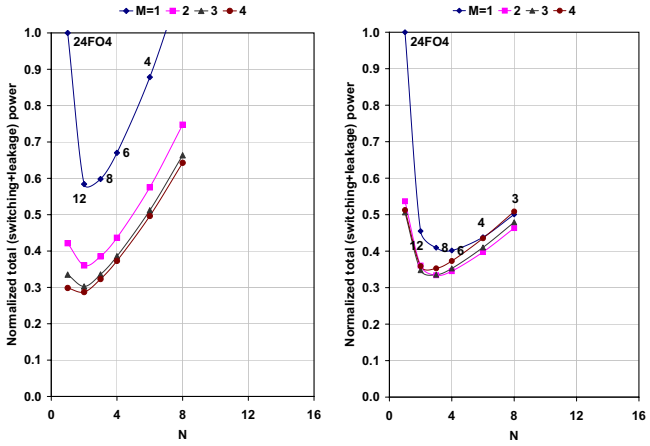
where *CPI(M, N)* is the greatest number of cycles among the *M* processors divided by the sum of total executed instructions by the *M*-processor system. Note that a larger CPI means worse performance. Equation (10) can be re-expressed as:

$$T_{logic}(V_{DD})+T_{flop}(V_{DD}) = \frac{l(M,N)}{M}\cdot\left(\frac{T_{logic}(V)}{N}+T_{flop}(V)\right) \quad (11)$$

where *l(M, N)* is *M*×*CPI(M,N) / CPI (1,1)*—a performance degrading factor that is greater than 1 for the limited parallelism case. For the ideal case, *CPI(M, N)* should remain constant regardless of *N* for a fixed *M* and decrease linearly with *M*. However, in reality, increasing either *M* or *N* increases CPI due to branch mispredictions, data dependencies, and so forth. According to [8], [9], and other earlier studies, the CPI increase almost linearly with the number of pipeline stages. Figure 5 shows the normalized total power of *M-N* processors at the same throughput. We assume that the CPI of *M-N* processors degrades by 25% when the number of pipeline stages are doubled and the CPI scales lineally with the number of parallel processors without any loss in Figure 5-(a). In Figure 5-(b), we assume that the CPI of *M-N* processors degrades by 25% when the number of parallel processors are doubled and the CPI scales linearly with the number of pipeline stages without any loss. Note that different degradation factors depending on the applications can be applied to (11) as analysis parameters.

According to the analyses, increasing CPI degradation with the pipelining depth shifts the total-power optimal point to a deeper logic depth—12FO4 for all the parallel processing widths in Figure 5-(a)—with increased total power consumption compared to the embarrassingly parallel case. Even when we increase the degradation factor up to 35%, the optimal logic depth is still 12FO4. Also, note that there is only negligible reduction by using the 4-wide parallel processing width

Figure 5: Normalized total power at the same throughput point with limited parallelism (B/C=0.25, and *G* = 0.4).



(a) 25% CPI degradation per 2× pipelining depth increase (b) 25% CPI degradation per 2× processing width increase.

although we assume a perfect performance scaling with increasing parallel processing width. When we consider the CPI degradation by increasing the parallel processing width, the 4-wide parallel processor starts to consume more energy than 3-wide at the 25% CPI degradation rate because of the marginal voltage reduction and the increased leakage overhead compared to the 2 or 3-wide processor case at the optimal points.

6. Impacts by Process Variations

As the feature size shrinks, process parameter variations increase for channel length, width, and threshold voltage, for example. These parameter variations change the delay and the leakage power of circuits significantly [2]. With a given nominal mean μ_{cp} and standard deviation σ_{cp}, the critical path delay density functions resulting from within-die (WID) or die-to-die (D2D) parameter variations are modeled as normal distributions. The probability of one critical path satisfying a specified maximum delay t_{max} is calculated as [2]:

$$P_{T_{cp}}(t \le t_{max}) = F_{T_{cp}}(t_{max}) = \int_0^{t_{max}} f_{T_{cp}}(t)\,dt \quad (12)$$

where t is the variable critical path delay and $F_{T_{cp}}$ is either the WID ($F_{T_{cp\_WID}}$) or D2D ($F_{T_{cp\_D2D}}$) cumulative distribution for one critical path. A chip, however, contains many critical paths, all of which must satisfy the worst-case delay constraint. Assuming N_{cp} is the number of independent critical paths for the chip, the WID cumulative distribution for the chip can be modeled as [2]:

$$P_{T_{cp\_WID\_MAX}}(t \le t_{max}) = (F_{T_{cp\_WID}}(t_{max}))^{N_{cp}} \quad (13)$$

When the process parameter variations are introduced, the maximum delay through the pipeline stages of parallelized processors is no longer a deterministic value based on the critical paths in the system. While some pipeline stages may be fast, other blocks may be slower. Likewise, some parallelized system may be fast, others may not. In both cases, the operating frequency is determined by the slowest one and the supply voltage should be scaled up for the slowest one to maintain the same throughput. This argument can be modeled statistically as follow. When $F_{1,1}$ is the baseline processor cumulative critical path delay distributions considering only the WID random *uncorrelated* variations that *may* become more dominant than the systematic (or random *correlated*) variations, the probability of satisfying t_{max} for the *M-N* processor is:

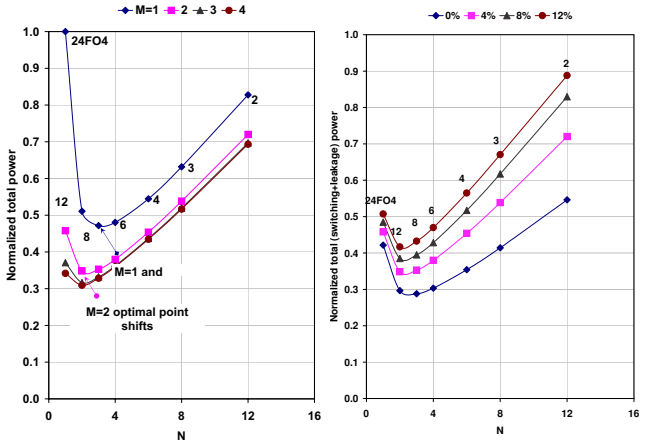
$$P_{M,N}(t \le t_{max}) = F_{M,N}(t_{max}) = (F_{1,1}(t_{max}))^{M\times N} \quad (14)$$

where $F_{M,N}$ is the *M-N* processor cumulative critical path delay distribution. Equation (14) implies that both deeper pipelining (shallower logic depth per pipeline stage) and wider parallel processing will suffer more severely from the variations. Furthermore, deeper pipelining makes the number of logic gates (n) in the critical paths smaller as well. Considering only the WID random uncorrelated variations, the relative variations in the critical path delay are expected to increase due to the reduced averaging effect over the number of logic gates in the critical path such that [16]:

$$\sigma_{cp}/\mu_{cp} \propto 1/\sqrt{n} \quad (15)$$

Therefore, we can expect that the optimal pipelining depth will be shifted to a deeper logic depth (shallower pipelining) than the nominal case. Note that systematic (or random correlated) variations shifts the process parameters for the transis-

Figure 6: Normalized total power to achieve 95% yield for different amounts of variations at the same throughput point (*B*/C = 0.25, G=0.4).



(a) σ_{cp}/μ_{cp} for 24×FO4 at 1.2V = 4%

(b) σ_{cp}/μ_{cp} for 24×FO4 at 1.2V = 0%, 4%, and 8% at M = 2

tors in a given region of a die in the same direction where σ_{cp}/μ_{cp} is independent of n. Also, note that the systematic WID variations impact the number of independent critical paths in (14). Finally, as supply voltage approaches V_{th}, σ_{cp}/μ_{cp} or the critical path delay variation increases significantly, because a reduced supply voltage leads to an increased sensitivity of logic circuit delays to parameter variations. This in turn influences V_{th} according to [16]:

$$S_T^{V_{th}} = \frac{V_{th}}{T(V)} \times \frac{\partial T(V)}{\partial V_{th}} = \frac{\alpha \times V_{th}}{(V - V_{th})} \quad (16)$$

This implies that a significant fraction of the voltage scaling gain will be lost to compensate for the critical delay variations. This can be a factor that makes deeper pipelining or wider parallel processing inefficient.

Figure 6-(a) shows the normalized total power to achieve 95% yield at the same throughput point. The WID variation, σ_{cp}/μ_{cp} of the baseline 24×FO4 logic depth at 1.2V was set to 4%. For each 0.1V supply voltage interval and each logic depth and parallel processing width under investigation, we estimated a new σ_{cp}/μ_{cp} based on (14), (15), and (16). With the estimated numbers, the logic delay T for each discrete voltage, logic depth, and parallel processing with at a specific yield point was characterized to achieve a 95% yield. Figure 6-(b) illustrates the total power trends as we increase the amount of variations (σ_{cp}/μ_{cp}) from 0% (no variation) to 8% at a fixed parallel processing depth (M=2). According to Figure 6-(a) and (b), the process parameter variations shift the optimal logic depths from 6 to 8×FO4 for 1-wide processors, and from 8 to 12×FO4 for 2-wide processors. This supports our earlier argument—the deeper logic depth (or shallower pipelining) is better due to the averaging effect of the delay variations. The other effect of process parameter variation is that the absolute amounts of total power reduction are reduced by 9~12% depending on the analysis parameters, because higher supply voltage is required to make the slowest pipeline stage and/or processor in the *M-N* processor system meet the timing or maximum frequency constraint.

7. Conclusions

In this study, we investigated the optimal combination of pipeline depth and parallel processing for total power reduction. Our analyses for embarrassingly parallel applications show that 8~12×FO4 logic depth pipelining and 2-wide parallel processing results in the optimal combination that reduces the total power by 70%~80%, depending on the fraction of leakage power and the timing and power overheads of timing elements. As we employ wider parallel processing further reduction is possible, but the difference is not very noticeable compared to the 2-wide case at the optimal pipelining depth. Compared to a system employing either a 1-wide optimal pipelining of 6~8×FO4 logic depth or a 2-wide pipelining of 24×FO4 baseline logic depth alone, the optimal combination reduces total power by 20~44%. The analyses concerning the performance loss from increasing either pipelining depth or parallel processing width show that the optimal logic depth should be shifted to a deeper logic depth—12×FO4 with the same 2-wide parallel processing for the high parallelism case. Furthermore, our results concerning the process parameter variations suggest that the optimal pipelining depth and parallel processing width should be shallower and narrower compared to the optimal combination when variations are ignored.

Acknowledgements

The authors would like to thank S. Borkar, M. Haycock, M. Khellah, Y. Ye, J. Tschanz, and D. Somasekhar for their valuable supports and discussions.

References

[1] V. Srinivasan, et al., "Optimizing pipelines for power and performance," *IEEE Microarchitecture*, pp. 333~344, 2002.

[2] K. Bowman, et al., "Impact of die-to-die and within die parameter fluctuation on the maximum clock frequency distribution for gigascale integration," *IEEE JSSC*, pp. 183~190, Feb. 2002.

[3] S. Borkar, et al., "Design and reliability challenges in nanometer technologies," *IEEE DAC*, pp. 75~75, 2004.

[4] A. Chandrakasan, et al., "Low-power CMOS digital design," *IEEE JSSC*, 27(4), pp. 473~484, Apr. 1992.

[5] R. Ho, et al., "The future of wires," *IEEE Proc.*, Vol. 89(4), pp. 490~503, Apr. 2001.

[6] E. Sprangle, et al., "Increasing processor performance by implementing deeper pipelines," *IEEE ISCA*, pp. 25~36, 2002.

[7] M. Hrishikesh, et al., "The optimal logic depth per pipeline stage is 6 to 8 FO4 inverter delays," *IEEE ISCA*, pp. 14~24, 2002.

[8] A. Hartstein et al., "The optimum pipeline depth for a microprocessor," *IEEE ISCA*, pp. 7~13, 2002.

[9] A. Hartstein et al., "Optimum power/performance pipeline depth," *IEEE Microarchitecture*, pp. 333~344, 2003.

[10] S. Heo et al., "Power-optimal pipelining in deep submicron technology," *IEEE ISLPED*, pp. 218~223, 2004.

[11] D. Markovic, et al., "Methods for true energy-performance optimization," *IEEE JSSC*, 39(8), pp. 1282~1293, Aug. 2004.

[12] D. Liu, et al., "Trading speed for low power by choice of supply and threshold voltage," *IEEE JSSC*, 28(1), pp. 10~17, Jan. 1993.

[13] G. Sery, S. Borkar, and V. De, "Life is CMOS: Why chase life after?," *IEEE DAC*, pp. 78~83, 2002.

[14] B. J. Sheu, et al., "BSIM: Berkeley short-channel IGFET model for MOS transistors," *IEEE JSSC*, 22(1), pp. 558–566, Aug. 1987.

[15] S. Mukhopadhyay, et al., "Gate leakage reduction for scaled devices using transistor stacking," *IEEE TVLSI*, 11(4), pp. 716~730, Aug. 2003.

[16] M. Eisele, et al., "The impact of intra-die device parameter variations on path delays and on the design for yield of low voltage digital circuits," *IEEE TVLSI*, 5(4), pp. 360~368, Dec. 1997.