Optimal Useful Clock Skew Scheduling In the Presence of Variations Using Robust ILP Formulations*

Vaibhav Nawale
Intel Corporation, Santa Clara

Thomas W. Chen
Colorado State University, Ft. Collins, CO 80521

ABSTRACT

This paper exploits useful skew to improve system performance and robustness. We formulate a robust integer linear programming problem considering the interactions between data and clock paths on a microprocessor chip to improve clock frequency. The timing slack is optimized for each path to determine a clock schedule. The percentage of timing violations, obtained from a 1000 point Monte Carlo simulation, is higlighted as yield predictions and conveys the robustness of the clock schedule. The results show performance improvement of up to 9.747% with 20% yield and up to 6.682% with 100% yield. The novelty of the proposed method is its ability to tradeoff between performance improvement in frequency and robustness, via a single variable in the formulation.

1. INTRODUCTION

The ITRS future frequency trend for high performance microprocessors is predicted upwards to 20Ghz. One of the direct impacts of aggressive scaling is the increase in clock skew as a percentage of clock cycle time reducing the clock budget for useful computation. Early approaches were focused on designing clock distribution topologies to keep clock skew at a minimum. These efforts managed results close to zero skew but with huge power and area sacrifices. Also, with zero skew, the maximum achievable operating frequency is limited to the maximum datapath delay in the circuit. In the quest to find alternatives, there has been significant effort [4], [12], [6], [3], [10], [2] [5] to explore useful skew to improve performance. Although useful skew enables systems to operate at higher clock frequencies, more and more signal paths get pushed towards the edge of satisfying timing requirements. As the amount of uncertainty increases with scaling [15], [13], [9], the probability of failure for a design with useful skew increases. The random process and environmental variations that dominate the behavior of devices are hard to predict, let alone eliminate, making the clock skew a highly difficult source to manage. There is thus the need for a methodology to model the uncertainties to improve robustness of the design. Robustness, in the context of useful skew, is the percentage of chips that can meet timing requirements for the optimized clock schedule, intended to improve performance.

This paper presents a LP-based ILP formulations to optimize useful skew with considerations for uncertainties in clockpath and datapath delays. It attempts to predict the amount of risk involved in pushing the operating frequency using useful skew. Solutions to robust formulations are typically not optimum. Throughout this paper optimal is refered to the best trade off among competing constraints, i.e., performance and robustness. Our approach has the following salient features: 1) it considers variations in combinational block and clock delays; 2) it incorporates considerations for the register locations and physical clocking domains that have the same amount of clock skew adjustment; and 3) previous approaches [10], [5], [4] have considered robustness, however our approach has the ability to control the trade-off between performance and robustness of the clock schedule. When applied to a 64-bit microprocessor the percentage improvement in clock frequency ranges from 9.747% with 20% yield (no considerations for uncertainties), to a percentage improvement of 6.682% with a yield of 100% (data and clock uncertainties) The yield (i.e. the robustness) data was obtained using a 1000 point Monte Carlo simulation.

The remainder of the paper is organized as follows. Section 2 discusses existing research and Section 3 presents the background of our work. Section 5 presents the formulations used to optimize the clock frequency. Clock Distribution details of the microprocessor used in our experiment are given in Section 4. Section 6 and Section 7 discuss the experimental results and conclusions respectively. .

2. EXISTING RESEARCH

Retiming [8] was one of the earlier solutions to the problem of clock skew optimization. *Retiming* cannot be applied to certain areas and could increase the number of flops. This motivated Fishburn [4] to explore LP approaches to optimize different circuit parameters, performance and robustness, by introducing useful skew. The inability in Fishburn's approach to handle level triggered flops was highlighted and a formulation for *for level triggered memory elements* was presented by Sakallah et. al. [12]. Chao and Sha used the concept of scheduling to form a common background and proposed a combined *retiming and clock skew introduc-*

*

ICCAD '06, November 5-9, San Jose, CA

tion approach in [2] to study the interplay between retiming and clock skew introduction. Simultaneous, as opposed to sequential, clock skew scheduling and retiming could potentially provide better results was discussed in [6]. A two-phase graph approach was proposed in [3] to improve Fishburn's approach in terms of maintaining an upper bound on the skew. The lack of considerations for parallel and feedback paths was addressed in [10] by suggesting an approach to optimally assign each path with a skew value after determining a common permissible range for all the paths, between any two nodes. This approach provided a more robust design compared to that in [3]. However, all variations were collected into a single parameter making it impractical to handle large complex chips. [5] proposed a Quadratic programming problem to determine a clock skew schedule with improved tolerance to process variations by minimizing the least square distance between the desired and actual values of the clock skew schedule over the entire circuit. A different approach was presented in [11]. The objective was to obtain a clock schedule that achieves a shorter clock period and can be realized by a light clock tree. The algorithm takes into account the register locations while optimizing the clock schedule so that the wire lengths for clock distribution are within acceptable limits.

In light of the advances in technology and increased variations the existing approaches have the following shortcomings: 1) most of the approaches failed to consider the impact of variations in the combinational delays; 2) most of the approaches did not consider clock domains while trying to optimize the clock period to minimize wire lengths and power consumption. 3) there is no mechanism in the existing algorithms to trade performance against robustness.

3. BACKGROUND

The use of *slack* that exists between the required arrival time for the data ($t_{reqd.}$) and the actual arrival time of the data (t_{delay}) has the following advantages: 1)the slack information helps eliminate the difference between level triggered latches and edge triggered flops. It provides an effective model for both level triggered and edge triggered memory elements; and 2) the convenience of slack from typical static timing analysis takes into account the clock delays at the latch points, relative to data delay. There are two different types of paths in a typical logic circuit, *state paths and phase paths*. If adjacent memory elements are fed with clock signals having opposite phases, the path between them is characterized by half a clock cycle and is called a phase path. The state path exists between a pair of memory elements fed with the same phase clock and triggering at the same clock event. An increase in the slack for setup implies a decrease in the slack for hold. Thus, for an independent path, the maximum amount of improvement can be determined by formulating a problem that maximizes M subject to the constraints expressed below

$$\begin{aligned} x_i - x_j &\leq slack_{max} - M \\ x_j - x_i &\leq slack_{min} \end{aligned} \tag{1}$$

where $slack_{max}$ and $slack_{min}$ are the minimum slacks for the maxtime and mintime path between register i and j, M is the possible reduction in clock period and x_i *and* x_j the clock arrival time at the respective latches.

4. CLOCK DISTRIBUTION DETAILS

In modern 64-bit microprocessors the clock distribution network is often highly hierarchical and complex in order to achieve the maximum operating frequency possible. The use of useful skew fits well with its requirement for high performance. We apply our proposed useful skew optimization algorithm to a latest 64-bit microprocessor [7] to examine its effectiveness. Figure 1 shows that the PLL is the center of the clock distribution topology. It is responsible for generating the actual operating clock signal. Each CPU core is divided into three frequency regions, with a digital frequency divider (DFD) in each region, for independent control.

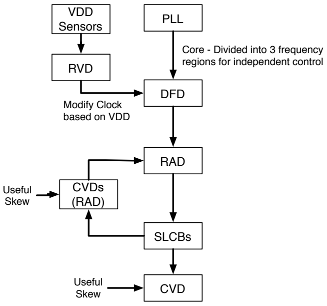
VDD Sensors
PLL
Core - Divided into 3 frequency regions for independent control
RVD
DFD
Modify Clock based on VDD
RAD
Useful Skew
CVDs (RAD)
SLCBs
Useful Skew
CVD

Figure 1: Electrical Aspects; The Clock Distribution

The voltage sensors convey to the regional voltage detector (RVD) changes in the regional voltage and help maintain the clock to data ratio thus avoiding possible timing violations. The second level clock buffers (SLCBs) comprise of self bias amplifiers that re-enforce the clock signal from the digital frequency dividers. The regional active deskew (RAD) internally comprise of a set of CVD's and a phase comparator that are responsible for region based active deskewing to reduce the skew resulting from process, voltage and temperature variations. The SLCBs feed the clock signal to the local buffers through a different set of CVDs and then down to the gaters. The CVDs are programmable devices, with a 3-bit(000-111), 8 quantized level delay programmability. These CVDs can be programmed via scancand firmware to debug post-silicon defects and to remove any skew that may be present. CVDs can also be used to introduce useful skew. In theory, the CVDs present within the RAD can be programmed to increase the range available for those CVDs that follow the SLCBs. Our experiments were carried out with adjustments enabled at the SLCB level only and at both RAD and SLCB levels. Table 4 summarizes the clock distribution system on the 64-bit microprocessor and gives an idea of the size of test circuit. Each CVD at the SLCB level controls a set of latches that will take the same skew adjustment from the controlling CVD and these latches are classified into a single domain.

5. ROBUST FORMULATIONS

A simple ILP formulation to maximize clock frequency by

Element	Number/per core
Digital frequency divider (DFD)	3
Regional Voltage Detector (RVD)	11
Secondary Level Clock Buffers (SLCB)	27
Regional Active Deskewing (RAD)	35
Clock Vernier Devices (CVD)/ Number of domains	4624
Latch-to-Latch paths maxtime	1,937,995
Latch-to-Latch paths mintime	1,299,283
Number of Latches	279,431

Table 1: Statistics of Clock Distribution

introducing useful skew is shown in Equation 2.

$$\begin{aligned} &\textit{Maximize M subject to:}\\ &S : x_i - x_j + coeff \cdot M \leq slack_{max}\\ &H : x_j - x_i \leq slack_{min}\\ &B1 : \sum_{m=1}^{n} w_{lm} = 1 \textbf{ and } B2 : \sum_{m=1}^{n} Q_m \cdot w_{lm} - x_l = 0\\ &I1 : w_{lm} \in \{0, 1\} \end{aligned} \tag{2}$$

where l is the number of flip flops and n is the number of quantization levels within a clock tuning element. w_{lm} is the m^{th} quantization variable associated with the l^{th} memory element and Q_m is the m^{th} quantization level. Constraints B1,B2 and I1 together make sure that x_i *and* x_j can take only one of the n quantized values. The S and H constraints are the setup and hold constraints expressed in terms of slack. $slack_{max}$ and $slack_{min}$ are the minimum slacks for the maxtime and mintime paths between register i and j. The setup constraint tries to maximize the negative skew that can be introduced while the hold constraint places an upper bound on the maximum negative skew that can be introduced to avoid mintime violations. *coeff* indicates how many clock cycles were allotted for a timing arc. It can take values like 0.5, 1, 1.5, which implies either half a clock cycle, one clock cycle, or one and a half clock cycle. M indicates the amount of reduction achievable in clock cycle time. The solution to the ILP discussed in Equation 2 pushes all the timing constraints to the edge of satisfying timing constraints. The nominal values for clock arrival times and data are not always precisely known and small variations in the input data can completely invalidate this solution.

There have been two approaches to address data uncertainty over the years (a)Stochastic programming, and (b) Robust Optimization. However, the ease with which robust optimization can be applied to real world problems has made it a popular approach to address data uncertainty. Soyster [14] proposed a robust approach which handled data uncertainty in a highly conservative manner. The feasible region in this formulation was specified via set containment instead of traditional set of convex inequalities.Soyster's formulation can expressed as in Equation 3

$$\begin{aligned} &maximize : c'x \textit{ subject to :}\\ &\sum_j a_{ij}x_j + \sum_{j\epsilon J_i} \hat{a}_{ij}y_j \leq b_i \quad \forall i\\ &-y_j \leq x_j \leq y_j \textbf{ and } l \leq x \leq u \ \forall y \geq 0 \end{aligned} \tag{3}$$

where data uncertainty is captured using bounded random variable $\hat{a}_{ij}$ for the coefficient matrix. The coefficient takes values in $[a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$ where a_{ij} is the nominal value of the coefficient. J_i is the set of a_{ij}, $j \in J_{ij}$, that are subject to uncertainty. y_j is the additional variable, for each x_j, defining new bounds for x_j. It can be shown[14] that for every possible value of the coefficient, the solution remains feasible. The purpose of the term $\sum_{j\epsilon J_i} \hat{a}_{ij} \mid x_j \mid$ is to create a "gap" between the optimal solution of the nominal problem (i.e. $\sum_j a_{ij}x_j^*$) and b_i to provide robustness. The clock arrival time, x_i, at a given latch i is subject to uncertainty. Thus, the clock arrival time is bounded by $[x_i - \delta, x_i + \delta]$. The robust ILP formulation for clock skew optimization based on Soyster's approach is given in Equation 4.

$$\begin{aligned} &\textit{Maximize M subject to:}\\ &S : x_i - x_j + coeff \cdot M + \delta \cdot y_i + \delta \cdot y_j \leq slack_{max} - (\Delta \cdot delay)\\ &H : x_j - x_i + \delta \cdot y_i + \delta \cdot y_j \leq slack_{min} - (\Delta \cdot delay)\\ &B1 : \sum_{m=1}^{n} w_{lm} = 1 \textbf{ and } B2 : \sum_{m=1}^{n} Q_m \cdot w_{lm} - x_l = 0\\ &B3 : -y_l \leq x_l \leq y_l \textbf{ and } I1 : w_{lm} \in \{0, 1\} \end{aligned} \tag{4}$$

where the S and H constraints differ from those in Equation 2 in terms of the protection they provide. y_l is the additional variable for each x_l defining new bounds. The constraint B3 ensures that optimality $y_l = |x_l^*|$ and thus each x_l is represented as $x_l^*(1+\delta)$ where δ was the intended percentage protection for the clock arrival times. Δ represents intended percentage protection for the delay through the combinational block. The value $(\Delta \cdot delay)$ is subtracted from the slack to represent delay variations. For a particular timing arc the uncertainties in the combinational block along the path alter the actual arrival time of the delay and thus the available slack. The uncertainties in the combinational block delay can thus be modeled as uncertainties in the slack. The motivation to use Soyster's formulation was to provide maximum protection against uncertainty. However, the formulation is extremely conservative. It is highly desirable to provide a mechanism to allow tradeoff between robustness and performance. A robust LP formulation proposed by Bertsimas and Sim [1] can be applied to handle parameter uncertainty and provide a lever to control the tradeoff between performance and robustness. Bertsimas' formulation is shown in Equation 5.

$$\begin{aligned} &\textit{maximize } c'x \textit{ subject to:}\\ &\sum_j a_{ij}x_j + \max_{\{S_i\cup\{t\}|S_i\subseteq J_i,|S_i|=\llcorner\Gamma_i\lrcorner,t\epsilon J_i \ \ S_i\}} \{\sum_{j\epsilon J_i} \hat{a}_{ij}y_j +\\ &(\Gamma_i - \llcorner\Gamma_i\lrcorner)\hat{a_{it}}y_t\} \leq b_i \quad \forall i\\ &-y_j \leq x_j \leq y_j \textbf{ and } l \leq x \leq u \textit{ and } y_j \geq 0 \quad \forall j \end{aligned} \tag{5}$$

where Γ_i is the protection factor. $a_{ij}, \hat{a}_{ij}, y_j$, and J_i are defined similar to those in Soyster's formulation. Protection is provided by the term $\beta_i(x) = \max_{\{S_i\cup\{t\}|S_i\subseteq J_i,|S_i|=\llcorner\Gamma_i\lrcorner\}}\{\sum_{j\epsilon J_i} \hat{a}_{ij} \mid x_j \mid\}$. If $\Gamma_i =\mid J_i \mid$, Bertsimas' formulation reduces to the Soyster's formulation with maximum protection. The tradeoff between the robustness and the conservativeness of the solution is provided by varying $\Gamma \in [0, \mid J_i \mid]$. However, this formulation is non-linear in nature. $\beta_i(x)$ can be converted to a linear formulation as shown in Equation 6.

$$\begin{aligned} &\beta_i(x^*) = \text{maximize} \sum_{j\epsilon J_i} \hat{a}_{ij} \mid x_j^* \mid z_{ij} \textit{ subject to} :\\ &\sum_{j\epsilon J_i} z_{ij} \leq \Gamma_i \textbf{ and } 0 \leq z_{ij} \leq 1 \qquad \forall j\epsilon J_i \end{aligned} \tag{6}$$

where x_j^* is the value of x_j at optimality. $z_{ij} = 1+(\Gamma_i - \lfloor \Gamma_i \rfloor)$. The optimization problem in Equation 6 ensures that the optimal solution has $\lfloor \Gamma_i \rfloor$ variables at 1 variable at $(\Gamma_i - \lfloor \Gamma_i \rfloor)$. The dual of Equation 6 is expressed in Equation 7.

$$\begin{aligned} &\textit{minimize} \sum_{j \epsilon J_i} p_{ij} + \Gamma_i z_i \text{ subject to:} \\ &z_i + p_{ij} \geq \hat{a_{ij}} \mid x^* \mid \qquad \forall i, j \epsilon J_i \\ &p_{ij} \geq 0 \ \forall j \epsilon J_i \textbf{ and } z_i \geq 0 \ \forall i \end{aligned} \tag{7}$$

p_{ij} is a dual variable added to form the dual of the Equation 6. The formulation in Equation 6 is feasible and bounded. Strong duality ensures that the dual formulation in Equation 7 is also feasible and bounded and that the objectives of Equations 6 and 7 coincide. Thus, the equivalent linear formulation for the problem can be obtained by substituting Equation 7 into Equation 5 and is shown in Equation 8.

$$\begin{aligned} &\textit{maximize } c^{'}x \textit{ subject to :} \\ &\sum_j a_{ij} x_j + z_i \Gamma_i + \sum_{j \epsilon J_i} p_{ij} \leq b_i \quad \forall i \\ &z_i + p_{ij} \geq \hat{a_{ij}} y_j \textbf{ and } p_{ij} \geq 0 \qquad \forall i, j \epsilon J_i \\ &-y_j \leq x_j \leq y_j \textbf{ and } l_j \leq x_j \leq u_j \qquad \forall j \\ &y_j \geq 0 \qquad \forall j \textbf{ and } z_i \geq 0 \qquad \forall i \end{aligned} \tag{8}$$

Equation 8 can be adopted and applied to the problem of optimizing clock skew as shown in Equation 9. Each arrival time x_i at a given latch i is protected for uncertainty within the range $[x_i - \delta, x_i + \delta]$. However, as opposed to Soyster's formulation the additional constraints $z_i + p_{ij} \geq \hat{a_{ij}} y_j$ and $l_j \leq x_j \leq u_j$ in Equation 8 (represented by constraints C1,C2 and B3 in equation 9) help provide varying amount of protection to the arrival time within the range $[x_i - \delta, x_i + \delta]$ by changing the value of Γ_i for the i_{th} constraint.

$$\begin{aligned} &\textit{Maximize } M \textit{ subject to:} \\ &S : x_i - x_j + coeff \cdot M + p_{ki} + p_{kj} + \Gamma_k \cdot Z_k \leq \\ &\qquad slack_{max} - (\Delta \cdot delay) \\ &H : x_j - x_i + p_{hi} + p_{kj} + \Gamma \cdot Z_k \leq \\ &\qquad slack_{min} - (\Delta \cdot delay) \\ &C1 : Z_k + p_{ki} \geq \delta \cdot y_i \textbf{ and } C2 : Z_k + p_{kj} \geq \delta \cdot y_j \\ &B1 : \sum_{m=1}^{n} w_{lm} = 1 \textbf{ and } B2 : \sum_{m=1}^{n} Q_m \cdot w_{lm} - x_l = 0 \\ &B3 : -y_l \leq x_l \leq y_l \textbf{ and } I1 : w_{lm} \in \{0,1\} \end{aligned} \tag{9}$$

In Equation 9 k is the number of timing arcs in the formulation. The constraints C1 and C2 place a restriction on what values p_{ki} *and* p_{kj} can assume based on the intended percentage protection (δ) for the clock arrival times. This influences the value of Z_k and thus, by varying Γ_k a tradeoff between robustness and performance can be achieved. The constraints B1, B2, and I1 help to enforce quantization. Δ is the intended percentage protection for the delay through the combinational logic path.

6. EXPERIMENTAL RESULTS

All solutions were obtained using the CPLEX solver 8.1.0 running on a 3GHz. Pentium 4 microprocessor with 1GB RAM. A combination of C/C++ and PERL scripts were written to construct the LP and ILP formulations. The simulations were classified under two different categories, Experiment 1 was performed without the integer constraints, thus relaxing the quantization. Experiment 2 enforced strict quantization. In each experiment, two scenarios were considered. The first scenario did not enable any CVDs in the RAD. The second scenario enabled the CVDs in both RAD and SLCBs. The reason for considering these two scenarios was to investigate the effectiveness of incorporating clock tuning elements at different hierarchies in the clock tree. Each of these scenarios had provisions for 2 different cases. The first case was 5% clock uncertainty and 2% data uncertainty, the second case was 5% clock uncertainty and 0% data uncertainty. The choice of 2% (3-σ) variation for data arcs takes into account the statistical averaging of gate delays along a data timing arc for the 180nm technology.

6.1 Experiment 1

This was used to determine the trend of variations in the percentage improvement in clock period and robustness, for the different formulations. The solutions provide upper bounds for the clock frequency possible through useful skew but do not have quantized arrival times.

	Clock 5%,Data 2% protection	**Clock 5%,Data 0% protection**
formulation	improvement (%)	improvement (%)
LP	8.635	8.635
Soyster	6.870	8.205
$\Gamma = 0$	7.246	8.635
$\Gamma = 0.3$	7.173	8.506
$\Gamma = 0.6$	7.043	8.376
$\Gamma = 0.9$	6.914	8.247
$\Gamma = 1$	6.870	8.204
$\Gamma = 2$	6.870	8.204

Table 2: %age improvement in clock freq.; without RAD and adjustable delay quantization enabled

	Clock 5%,Data 2% protection			**Clock 5%,Data 2% protection**		
formul- ation	HV max	SV max	percentage violation	HV max	SV max	percentage violation
LP	-4.52	-14.97	68.90	-4.13	-15.46	65.9
Soyster	0.00	0.00	0.00	-0.87	-13.25	1.7
$\Gamma = 0$	-2.01	-2.16	13.50	-2.57	-17.10	50.8
$\Gamma = 0.3$	-1.28	-1.73	4.50	-3.12	-13.78	41.3
$\Gamma = 0.6$	-1.31	-2.17	0.30	-2.28	-12.98	25.1
$\Gamma = 0.9$	-0.11	-1.58	0.20	-1.86	-13.75	6.8
$\Gamma = 1$	0.00	-1.86	0.00	-1.73	-12.31	6.5
$\Gamma = 2$	0.00	-1.69	0.00	-1.72	-12.78	6.5

Table 3: MonteCarlo simulations; without RAD and adjustable delay quantization enabled

Table 2 shows the percentage improvement in clock frequency without RAD adjustments and without adjustable delay quantization. Table 2 shows that normal LP formulation without any considerations for uncertainty has the best improvement and Soyster's formulation shows the least improvement in clock frequency. The robust formulation shows varying levels of improvements for Γ between 0 and 2 and emphasizes the advantage of the proposed approach i.e. performance trade-off. Table 3 shows the level of robustness for each of the different formulations obtained from Monte Carlo simulations. For Monte Carlo simulations, data and clock delay uncertainties were assumed to have Gaussian distributions. The mean values of the Gaussian distributions for data were obtained from the static timing results. The 3-σ deviations for data and clock were chosen to be 2% and 5% of the mean values, respectively. The effects of delay uncertainties of both data and clock are mapped to timing slacks during Monte Carlo simulations. The column

SV max reports the maximum margin by which a maxtime constraint was violated over 1000 monte carlo simulations. The column *SH max* reports the maximum margin by which a mintime constraint was violated over a 1000 monte carlo simulations. The percentage violation is calculated as

$$\mathit{violation(\%)} = \frac{\mathit{no.\ of\ iterations\ with\ 1\ or\ more\ mintime\ paths\ failing}}{\mathit{total\ no.\ of\ iterationssims.}} * 100$$

The robust formulation shows a decrease in the percentage violation as we slide Γ from 0 to 2. The data from the *SV max* column in Table 3 can be used as the correction factor to slow down the clock to avoid maxtime violations, also called as *frequency binning*. Figure 2 shows the percentage improvement in clock period after frequency binning. The solutions marked as 'Normal LP' did not consider for any protection in either data or clock for both 0% and 5% data cases and as can be seen are the worst in performance after frquency binning. It is worth pointing out that the solutions based on 5% clock and 0% data uncertainties actually ended up with a lower performance gain, after frequency binning, compared to to those based on 5% clock and 2% data uncertainties thus highlighting the need for considering uncertainties in the design.

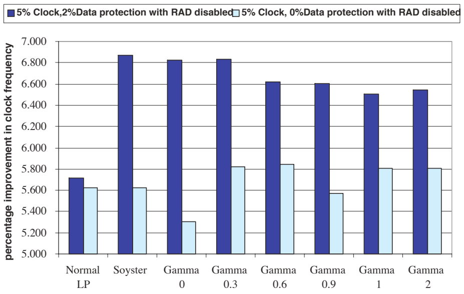

Figure 2: %age improvement in clock frequency after correction from Monte Carlo simulations without RAD enabled

Enabling RAD in the clock tree provides additional range for clock tuning since we move a level higher in the hierarchy. Comaprisons between Table 2 and Table 4 highlights the improvement in performance by enabling RAD.

	Clock 5%,Data 2% protection	**Clock 5%,Data 0% protection**
formulation	improvement (%)	improvement (%)
LP	9.747	9.747
Soyster	7.441	9.546
$\Gamma = 0$	7.550	9.747
$\Gamma = 0.3$	7.524	9.703
$\Gamma = 0.6$	7.498	9.660
$\Gamma = 0.9$	7.473	9.618
$\Gamma = 1$	7.465	9.604
$\Gamma = 2$	7.440	9.546

Table 4: %age improvement in clock freq.; with RAD enabled and without adjustable delay quantization

Table 5 shows the Monte Carlo simulations for the same scenario. The results followed a similar trend, in terms of varying performance improvement and robustness as for the case when CVD's at the RAD level were disabled.

	Clock 5%,Data 2% protection			**Clock 5%,Data 2% protection**		
formul-ation	HV max	SV max	percentage violation	HV max	SV max	percentage violation
LP	-3.21	-14.98	58.50	-2.98	-15.87	64.2
Soyster	0.00	-1.57	0.00	-0.85	-12.88	1.40
$\Gamma = 0$	-1.91	-2.57	12.6	-3.06	-20.52	49.30
$\Gamma = 0.3$	-3.02	-2.47	3.30	-3.12	-20.31	49.06
$\Gamma = 0.6$	-1.14	-2.44	0.50	-4.01	-14.88	43.20
$\Gamma = 0.9$	-0.02	-1.44	0.10	-4.60	-17.03	36.10
$\Gamma = 1$	-0.06	-1.80	0.10	-2.93	-17.09	27.10
$\Gamma = 2$	0.00	-1.93	0.00	-2.71	-13.67	6.70

Table 5: MonteCarlo simulations; with RAD enabled and without adjustable delay quantization

6.2 Experiment 2

The experiments described in this section provide solutions with quantized arrival times and thus are practically feasible. The LP formulations with quantization transform the problem into a mixed integer programming (MIP) problem. Due to increased number of variables and constraints in the LP formulation to accommodate quantization, the convergence became an issue during the experiments. The results presented in this Section are a mix of the results from the conventional MIP approaches and our modified randomized rounding approach. Whenever the conventional MIP solver (CPLEX) could not converge to a solution, modified randomized rounding was used to find the solution. In the modified randomized approach, the corresponding relaxed integer solutions were used as a starting solution. A subset of the CVD arrival times from the non-integer solution that matched the quantization were hard coded as absolute values and the integer constraints for them were eliminated. The decision regarding which quantized arrival times be hard coded was based on a random number generator. This was done to provide more flexibility of choice for the arrival times. The best solution was tracked and updated after each iteration. Tables 6 and 7 show a decrease in percentage improvement in clock frequency and an increase in robustness going from $\Gamma = 0$ to 2.

	Clock 5%,Data 2% protection	**Clock 5%,Data 0% protection**
formulation	improvement (%)	improvement (%)
ILP	8.635	8.635
Soyster	6.682	8.204
$\Gamma = 0$	6.924	8.635
$\Gamma = 0.3$	6.702	8.506
$\Gamma = 0.6$	6.690	8.376
$\Gamma = 0.9$	6.682	8.247
$\Gamma = 1$	6.682	8.204
$\Gamma = 2$	6.554	8.204

Table 6: %age improvement in clock freq.; without RAD and with adjustable delay quantization

	Clock 5%,Data 2% protection			**Clock 5%,Data 2% protection**		
formul-ation	HV max	SV max	percentage violation	HV max	SV max	percentage violation
ILP	-4.16	-15.23	61.50	-4.17	-18.21	65.2
Soyster	0.00	0.00	0.00	0.00	0.00	0.0
$\Gamma = 0$	-1.06	-1.63	11.5	-3.695	-21.250	63.0
$\Gamma = 0.3$	-0.73	-1.02	3.20	-3.359	-14.350	47.9
$\Gamma = 0.6$	-0.23	-0.83	0.60	-2.244	-13.968	20.4
$\Gamma = 0.9$	0.00	-0.91	0.00	-1.483	-17.649	2.0
$\Gamma = 1$	0.00	0.00	0.00	-1.643	-8.603	2.3
$\Gamma = 2$	0.00	0.00	0.00	-2.563	-9.129	2.9

Table 7: MonteCarlo simulations; without RAD and with adjustable delay quantization

Tables 8 and 9 show the percentage improvement and the Monte Carlo simulations, respectively, for the scenario with RAD enabled for Experiment 2. The improvement is limited to atmost 1% due to the fact that enabling RAD increases

	Clock 5%,Data 2% protection	Clock 5%,Data 0% protection
formulation	improvement (%)	improvement (%)
ILP	9.630	9.630
Soyster	6.682	9.025
$\Gamma = 0$	6.924	9.630
$\Gamma = 0.3$	6.807	9.589
$\Gamma = 0.6$	6.690	9.548
$\Gamma = 0.9$	6.682	9.507
$\Gamma = 1$	6.682	9.493
$\Gamma = 2$	6.682	9.384

Table 8: The %age improvement in clock freq.; with RAD and adjustable delay quantization enabled

	Clock 5%,Data 2% protection			Clock 5%,Data 2% protection		
formul-ation	HV max	SV max	percentage violation	HV max	SV max	percentage violation
ILP	-6.91	-17.8	77.8	-5.12	-16.37	80.0
Soyster	0.00	0.00	0.00	-2.85	-15.61	1.56
$\Gamma = 0$	-4.23	-3.17	10.2	-4.91	-12.52	75.8
$\Gamma = 0.3$	-3.80	-2.47	5.20	-3.62	-12.97	53.7
$\Gamma = 0.6$	-3.23	-1.84	4.60	-3.49	-11.88	28.8
$\Gamma = 0.9$	-2.32	-0.90	2.20	-3.10	-16.23	7.0
$\Gamma = 1$	0.00	-0.36	0.00	-1.66	-12.18	2.0
$\Gamma = 2$	0.00	-0.46	0.00	-0.61	-14.93	0.9

Table 9: MonteCarlo simulations; with RAD and adjustable delay quantization

the number of variables and constraints thus forcing us to use the modified randomized rounding for more cases and sacrificing optimization for run time. The run time for all formulations varied from a couple of minutes to atmost a couple of hours.

In general the trend shows that the cases with 2% data protection show greater reduction in optimal solution after quantization as they have constraints for both robustness and quantization that make the problem tightly constrained reducing the range of values for the clock arrival times. The randomized rounding further sacrifices optimality as it hard codes certain values. The effects of the reduced performance gain can be seen by comparing Tables 4 and 8. The trends in performance gain obtained from our experimental results are in accordance with the theoretical predictions for cases where data uncertainty was assumed to be 0%, i.e. the performance gain for the normal ILP-formulation is the same as that for the robust ILP formulation with $\Gamma = 0$. However, for cases where data uncertainty was assumed to be non zero, the performace gains for the forementioned formulations are not the same. This is due to the fact that for the robust formulation with $\Gamma = 0$, as shown in Equation 9, there still is a non-zero data protection on the the right hand side of the equation. Such a protection does not exist for the normal LP formulation in Equation 2.

7. CONCLUSIONS

The uncertainty due to process and environmental variations can be of significant concern in chip yield. The need for providing risk assessment and manipulation is paramount for future designs beyond the 90nm generation. Clock skew has been the bottleneck for improving system performance and useful skew to improve clock frequency has been proposed in the past. We present a novel approach which accounts for variations in both the clock arrival times and the delay through the combinational logic. The proposed approach allows designers to formulate problems with varying levels of robustness. One important observation from this work is that formulations for clock skew optimization considering less amount of uncertainties in DSM technologies can result in lower timing robustness and thus lower performance gain, compared to formulations that account for all sources of variations. The proposed method provides a single lever (Γ) to allow easy tradeoff between robustness and performance.

8. ACKNOWLEDGMENT

The research presented in this paper was partially funded by INTEL through a INTEL Research Fellowship. The generous support of INTEL during this project is greatly appreciated.

9. REFERENCES

[1] M. Bertsimas, D.Sim. Robust Discrete Optimization and Network Flows. *Oper. Res. Center,MIT*, Jan.

[2] L.-F. Chao and H.-M. Sha. Retiming and clock skew for synchronous systems. *ISCAS; pp 283-286*, May 1994.

[3] R. Deokar and S. Sapatnekar. A graph-theoretic approach to clock skew optimization. *ISCAS; pp 407-410*, 30 May - 2 June 1994.

[4] J. P. Fishburn. Clock skew optimization. *IEEE Trans. Comput.; pp 945-951*, July 1990.

[5] I. Kourtev and F. E.G. Clock skew scheduling for improved reliability via quadratic programming. *ICCAD; pp 339-243*, Nov 1999.

[6] X. Liu, M. Papaefthymiou, and E. Friedman. Maximizing performance by retiming and clock skew scheduling . *DAC; pp 231-236*, 1999.

[7] P. Mahoney, E. Fetzer, B. Doyle, and S. Naffziger. Clock Distribution on a Dual-Core Multi-Threaded Itanium- Family Processor . *ISSCC Session 16*, 2005.

[8] S. Malik, E. Sentovich, R. Brayton, and A. Sangiovanni-Vincentelli. Retiming and resynthesis: optimizing sequential networks with combinational techniques. *1990., Proc. 23rd Annual Hawaii Int. Conf. ; pp 397-406*, 2-5 Jan 1990 1997.

[9] V. Mehrotra and D. Boning. Technology scaling impact of variation on clock skew and interconnect delay. *Interconnect Tech. Conf., Proc. of IEEE*, 4-6 June 2001.

[10] J. Neves and E. Friedman. Optimal clock skew scheduling tolerant to process variations. *DAC ; pp 623-628*, 3-7 June 1996.

[11] M. Saitoh, M. Azuma, and A. Takahashi. Clustering based fast clock scheduling for light clock-tree. *DATE, Conf. ; pp 240-244*, 13-16 March 2001.

[12] K. e. a. Sakallah. Analysis and design of latch-controlled synchronous digital circuits . *DAC; pp 322-333*, March 1992.

[13] R. Saleh, S. Z. Hussain, S. Rochel, and D. Overhauser. Clock skew in the presence of IR-drop in the power distribution network. *IEEE Tran. on CAD of ISCAS*, Jun 2000.

[14] A. L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Oper. Res.; pp 1154-1157*, 1973.

[15] Y. Taur and D. Buchanan. CMOS scaling in nanometer regime. *Proc. IEEE; pp 486-503*, Apr 1997.