A Reduced Clock Delay Approach for High Performance Mesochronous Pipeline

Suryanarayana B. Tatapudi and José G. Delgado-Frias
School of Electrical Engineering and Computer Science
Washington State University
Pullman, WA 99164-2752
Email: {statapud, jdelgado}@eecs.wsu.edu

Abstract— A mesochronous pipeline scheme is described in this paper. In a conventional pipeline scheme each pipeline stage operates on only one data set at a time. In the mesochronous scheme, pipeline stages operate on multiple data sets simultaneously. The clock period in conventional pipeline scheme is proportional to the maximum pipeline stage delay while in mesochronous pipelining, it is proportional to the maximum pipeline stage delay difference, which means higher clock speeds are possible and number of pipeline stages is significantly less. In mesochronous approach the clock distribution network is simple and load on it is less resulting in significant power savings. Also, the variations in supply current drawn by clock network is significantly less in mesochronous scheme, thus power supply noise (*IR* drop and *Ldi/dt* noise) is less. An 8×8-bit multiplier using carry-save adder technique has been implemented in conventional and mesochronous pipeline approach using TSMC 180nm (drawn length 200nm). The over all power dissipation in mesochronous approach is less than 50% of the power dissipation in conventional approach. In conventional approach, the power dissipation in clock network and pipeline registers is close to 80% of total power dissipation, while in mesochronous approach logic dissipates more power.

***Index Terms*—pipelined systems, high performance, mesochronous pipelining, multipliers, low power.**

I. INTRODUCTION

Clocking is an essential component in digital system design. Switching events in a pipelined system occur in well defined order and at precise moments with reference to a globally distributed clock signal. Today's high frequency clocks have to be generated on chip and distributed throughout the chip. In any digital system, of all data and control signals, clock signal is the one with the largest fan-out, and fastest switching rate. To achieve higher operational (clock) frequencies digital designers are using ultra-thin super-pipelines, as a result of which the load on the clock distribution is increasing and it is becoming extremely difficult to distribute a clean giga-hertz frequency clock signal [1], [2]. Increasing portion of clock period is also being spent in countering clock uncertainties like uncontrolled transmission line effects, clock skew and clock jitter [3], [4]. Thus the useful portion of clock period available for computation is decreasing. With shrinking feature sizes, interconnects are becoming thin, long, and their resistance is increasing. With high speed signals (with fast rise and fall times) running on thin long wires, the inductive component of wire parasitic is gaining significance [5]. The increase in clock frequency, system size, and wire parasitic values are introducing power supply noise [6], [7]. In conventional pipeline schemes, huge currents drawn by the clock network and large number of pipeline registers is increasing the power consumption. The clock network's power consumption has increased to 50% of the total chip power consumption [8]. These huge currents are also causing higher *IR* drops in the power supply network, which is essentially a huge RLC network. Also, the large current slew rates (*di/dt*) coupled with on-chip inductance are generating significant amount of *Ldi/dt* noise on power supply. These power supply noise affect the power supply integrity and this is worsened due to decreasing supply voltage levels.

Architecture modifications can eliminate complex clock distribution to reduce power consumption, power supply noise and improve system performance. Alternate architectures like wave-pipelining, asynchronous pipelining and package wiring [8] have been proposed. While asynchronous pipelining may be appealing since it completely eliminates the need to distribute clock, it is complex compared to synchronous schemes [8], [9]. Our Mesochronous pipeline (MPP) scheme [10], [11], [12] modifies the pipeline architecture to address the power issues and also achieve higher performance.

In this paper we present the power performance gains from our MPP scheme and compare it with the conventional pipeline scheme. The organization of this paper is as follows. In Section II, the MPP concept is discussed. In Section III, we discuss the implementation of an 8-bit multiplier in conventional and mesochronous pipeline architecture. Performance analysis of the multiplier is presented in Section IV. Finally in Section V, some concluding remarks are presented.

This work was sponsored in part by the Boeing Centennial Endowed Chair, School of Electrical Engineering and Computer Science, Washington State University.

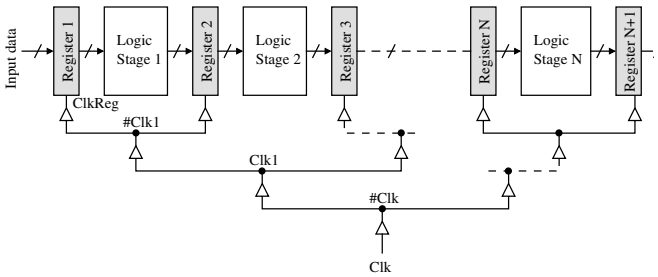

Fig. 1. Conventional pipeline scheme.

II. Mesochronous Pipeline Architecture

In a conventional pipeline (CPP) scheme, a digital system is divided into small sub-systems called pipeline stages separated by pipeline registers. The schematic of an *N*-stage pipelined system is shown in Fig. 1. In a pipelined system, at any given time each stage operates on only one data set. When the computation is complete in a stage, data is passed onto the next stage in the pipeline. Pipeline registers synchronize this data movement from one stage to the next with the help of globally distributed clock signal. This clock signal must trigger all the pipeline registers in the system simultaneously. New data is admitted into a stage only after data in that stage has been cleared and latched by the register following it. In a pipelined system, pipeline stage with the longest computation time dictates clock-cycle time for the entire system. The delay incurred in the registers also influences the clock period. Equation (1) defines the clock cycle time for a CPP scheme.

$$T_{clk\_cpp} \geq D_{\max} + D_R + t_s + \Delta_{clk} \tag{1}$$

where D_{max} is the computation time of stage with the longest propagation delay, D_R is the pipeline register delay, t_s is the pipeline register setup time and Δ_{clk} is clock uncertainty. From (1) it is clear that small clock periods are possible by reducing the delays: D_{max}, D_R, t_s and/or Δ_{clk}. Scaling can help decrease these delays and achieve smaller clock periods i.e. higher clock frequencies. However, in a given technology, to shrink the clock period further, the only delay which can be reduced is D_{max}. By partitioning each pipeline stage into more stages, stage delays can be reduced, in turn reducing D_{max} and T_{clk_cpp}. The result of such a partition is super-pipelines where there are more pipeline stages and registers. Increase in the number of registers complicates the clock distribution and also significantly increases the power consumption.

The mesochronous pipeline scheme (MPP) [10], [11], [12] modifies CPP scheme to gain higher performance, simplify the clock distribution and reduce power consumption. In the MPP scheme, like in the CPP scheme, a digital system is partitioned into pipeline stages. However it is clocked such that a pipeline stage is operating on more than one data set simultaneously. At any given time, multiple data sets can be present in a stage and these data sets are separated based on physical properties of internal nodes. This eliminates the need for some pipeline registers. The number of registers that can be eliminated depends on how many simultaneous data sets can be sustained in a stage without synchronization. This

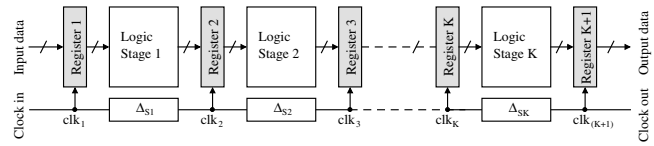

Fig. 2. Mesochronous pipeline scheme.

concept has some similarities to the wave-pipeline scheme [13], [14]. Unlike the CPP scheme, clock signal in MPP scheme travels along with the data and it is possible that different pipeline registers are triggered at different times. The schematic of this scheme is shown in Fig. 2. Clock signal path includes delay elements (Δ_{Si}) which emulate the delay experienced by data in pipeline stages. The clock period of the MPP scheme is defined by (2). A complete derivation of (2) is presented in [11], [12].

$$T_{clk\_mpp} \geq d_{\max(j)} - d_{\min(j)} + t_s + t_h + 2\Delta_{clk} \tag{2}$$

In (2), j is the stage with the maximum delay difference. Delay difference of a stage i is the difference between its maximum and minimum propagation delays ($d_{max(i)} - d_{min(i)}$). The delay difference of any stage, gives the amount of time the values generated at d_{min} have to be held until the computation is complete in that stage. The clock period in this pipeline scheme is determined by stage with the largest delay difference and safe time required before a new data set is admitted into this stage. From (2) it is easy to see that for any stage i, $d_{max(i)} \geq T_{clk\_mpp}$ is always true. This means that new data is admitted into a stage before computation on previously admitted data set is complete. Depending on the $d_{max(i)}$ value of a stage, at any given time two or more data sets can be present in a stage. From (2) it is clear that a smaller delay difference would result in higher clock frequency. It is not difficult to show that for any system the following expression is valid. $D_{\max} \geq \left(d_{\max(j)} - d_{\min(j)}\right)$, which implies that $T_{clk\_mpp} \leq T_{clk\_cpp}$. Thus MPP delivers higher performance compared to conventional scheme. Also, complexity of clock distribution is greatly reduced as shown in Fig. 2 and so significant power savings are possible. This scheme also eliminates long interconnects in clock network, so the line parasitic values are lower and simpler delay models can be used for delay estimation. The number of registers required in MPP is significantly less compared to CPP, reducing the load and power dissipation of the clock network.

In MPP scheme, the clock signal travels with data. Delays are included in the clock signal path so that clock experiences the delay similar to data waves in pipeline stages. Consider the example of a stage shown in Fig. 3. The shaded regions in Fig. 3 are called computation cones and represent when computation is being performed in this stage. The clock edge at ***A*** samples a data set from the previous stage. After traveling through the register and the stage i, the data set arrives at the next register before time ***E***. The next register must latch this data for the next stage ($i+1$) at time ***E***. The clock edge at ***A*** must be delayed for time period ***AE*** which can be represented as

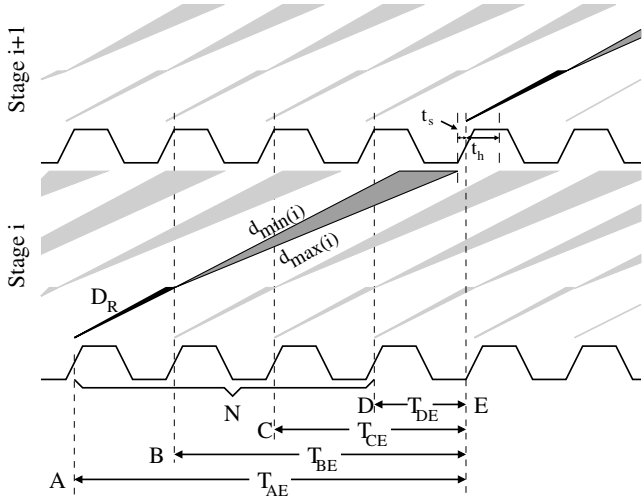

Fig. 3. Clock period and delay element.

$$T_{AE} = d_{\max(i)} + D_R + t_s + \Delta_{clk} \tag{3}$$

The delay value shown in (3) must be present in the clock signal path to ensure that delays experienced by logic and clock satisfy the relation: clock delay ≥ logic delay. This value of delay required in clock signal path is large. Instead of using such a delay element (Δ_{Si} in Fig. 2) we can take advantage of the periodic nature of the clock signal. As shown in Fig. 3, the delay AE can be expressed as a smaller delay ($\delta_{(i)}$) plus an integer multiple ($N_{(i)}$) of clock period.

$$T_{AE} = \Delta_{Si} = d_{\max(i)} + D_R + t_s + \Delta_{clk} = N_{(i)} T_{clk\_mpp} + \delta_{(i)} \tag{4}$$

From example in Fig. 3, possible combinations of $N_{(i)}$ and $\delta_{(i)}$ are shown in Table I. This technique helps further reduce the power consumption of clock network in the MPP scheme.

TABLE I
COMBINATIONS OF $N_{(i)}$ AND $\delta_{(i)}$

$N_{(i)}$	$\delta_{(i)}$
3	DE
2	CE
1	BE
0	AE

III. 8×8-BIT MULTIPLIER

In this section we present a multiplier simulated in CPP and MPP schemes, to illustrate how the MPP clocking technique impacts power and performance of a pipelined system.

Carry-Save Adder (CSA) technique [15] is a well known technique often used to realize fast multipliers. Using this technique, in an *M*-bit multiplier, *M* layers with 1-bit Full Adders (FA) reduce *M*-partial products to two partial products. Until this point the data flow is from one layer of adders to the next. In the last layer of the multiplier, the two *M*-bit partial products have to be merged to form the final product. The adder used for the final merging involves propagation of the carry signal and this would make it the bottleneck stage. Fast adder implementations like carry-look-ahead or carry-select structure can be used to reduce delay in

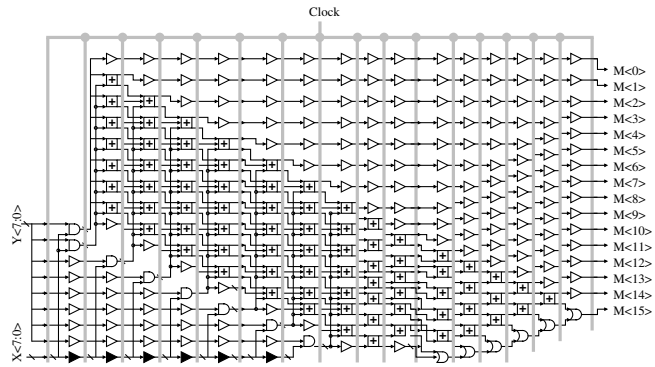

Fig. 4. 8×8-bit conventional pipelined multiplier.

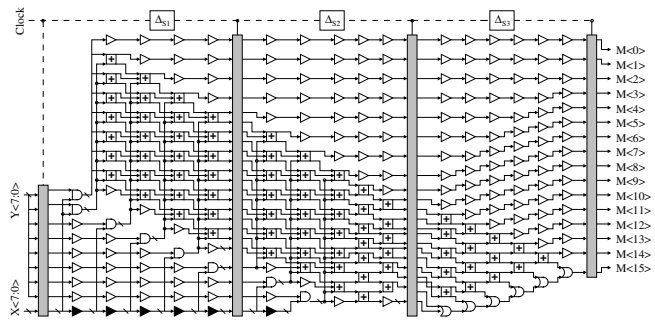

Fig. 5. 8×8-bit mesochronous pipelined multiplier.

the last layer; however these structures increase in complexity for large word lengths and produce diminishing returns. Instead of this, we added *M*-layers of 1-bit Half Adders (HA) to merge the final two partial products. This improves throughput, however there is increase in latency.

To achieve a fast multiplier the CSA architecture must be pipelined. In CPP scheme according to (1) minimum clock period is achieve by making each of the *2M* layers into stages of a pipeline, separated by pipeline registers. Effectively, the CPP multiplier would have *2M* stages with *2M+1* pipeline registers. An 8×8-bit pipelined multiplier implemented has 16 pipeline stages and 17 sets of inter-stage registers. The schematic of this multiplier is shown in Fig. 4. Figure 5 shows the schematic of the same 8×8-bit multiplier implemented in MPP scheme. The logic enveloped between any two adjacent register stages supports multiple data sets simultaneously. In this implementation there are only 3 pipeline stages and 4 register stages. The placement of the registers is based on the maximum delay difference that can be handled for a target clock frequency. This will be elaborated in Section IV.

A fast multiplier can be implemented if its basic cells have small propagation delay. The basic cells in the multiplier schematic shown in Fig. 4 are FA, HA, flip-flop, two input AND gate, two input OR gate, and buffers. The critical cells in the multiplier are the FA and HA. A differential transmission-gate implementation [12] has been used to realize the FA, HA, and all other basic cells. The registers in the multiplier have been realized using a dynamic two-phase D flip-flop [15].

In the CPP implementation of multiplier, a tree network has been used to distribute the clock signal to all the register stages of the pipeline. This is identical to the distribution shown in Fig. 1. Inverters have been used in place of buffers,

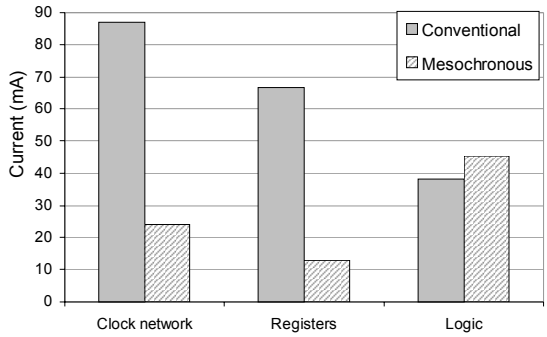

Fig. 6. Clock network, registers, and logic current at 2GHz.

and a fan-out of four has been used. The inverters in the tree network have sizes 50, 40, 25, 10 times the minimum sized inverter. Each register stage has another small tree network to deliver the clock to all the flip-flops in that stage without any vertical skew. The path taken by clock signal in MPP implementation is clear from Fig. 5. The clock travels close to the data path and includes delay elements realized using simple inverters. The periodic nature of clock signal has been used to achieve most of the delay, while small delay elements have been used to align the clock edge at the register stages [12].

IV. Performance

Simulations have been performed on multiplier layout in TSMC 180nm (drawn length 200nm, 1.8V supply voltage) CMOS technology. A number of simulations have been performed on the full adder to precisely characterize performance of this cell. The propagation delay for the full adder varied from 210ps (d_{min}) to 280ps (d_{max}), resulting in a maximum delay variation of 70ps [10], [11]. From simulations of the D flip-flop, the clock-to-output delay is approximately 130ps, set-up time is 65ps, and hold time is 5ps. From (1), the minimum achievable clock period in conventional scheme is 475ps. A fair compare between the two schemes in terms of power consumption is when they are operating at the same clock period. For this purpose a clock period of 500ps (2GHz) has been chosen. From (2) it is clear that for a clock period of 500ps, the maximum delay variation of any stage can be 400ps. The placement of registers as shown in Fig. 5 is based on this calculated limit on delay difference. The delay variation of the FA is 70ps, and maximum calculated delay variation is 400ps, and so maximum number of FA layers in a stage is five. This placement also accommodates additional variations that can occur in a stage. From Fig. 5 it can be seen that stage 2 is the critical stage as it has five FA/HA layers combined into a single stage. The logic enclosed between any two adjacent register stages supports two or more data sets simultaneously and the stage delay difference is less than 400ps.

Simulations have been performed to calculate the average current drawn by the clock network, registers, and logic in both the pipeline schemes. The current consumption in the clock network and flip-flops should be relatively constant. In logic stages, current drawn is data dependent. The average current drawn by the clock network, registers, and logic

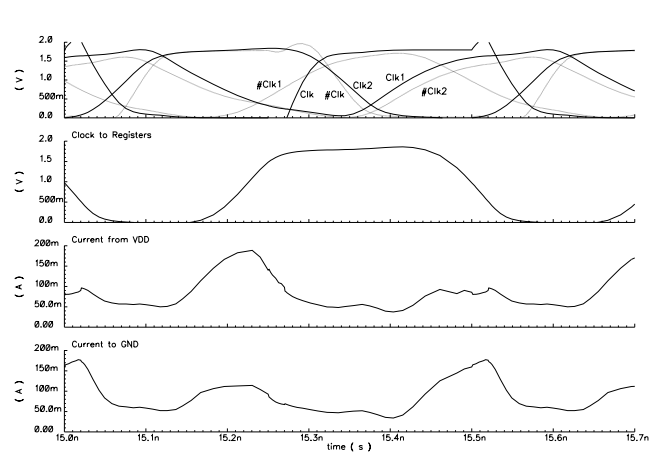

Fig. 7. Clock network current in CPP scheme at 2GHz.

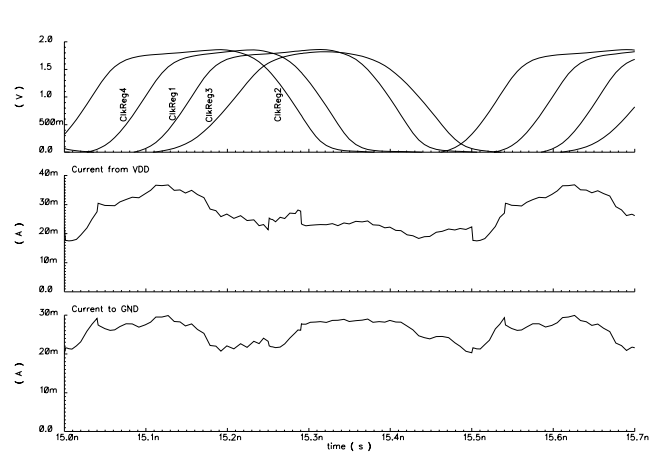

Fig. 8. Clock network current in MPP scheme at 2GHz.

stages, is shown in Table II [16]. The current drawn by the logic stages has been calculated for a significant activity in these stages. From simulations it has been observed that the CPP scheme consumes less current (power) than MPP scheme only if operated at one-third the speed of MPP [16].

TABLE II. Clock network, Registers, and Logic Current at 2GHz

Scheme	Current (mA)			
	Clock network	*Registers*	*Logic*	*Total*
CPP	86.9	66.6	38.2	191.7
MPP	24.2	12.8	45.3	82.3
CPP/MPP	3.6	5.2	0.84	2.3

A graphical comparison of the current results is shown in Fig. 6 [16]. In CPP scheme, the amount of current drawn by the clock network and registers is greater than in MPP implementation. This is due to the complex clock distribution and higher number of register stages in CPP. The overall current drawn is higher in CPP scheme.

Now let us take a closer look at the current drawn by the clock distribution network in both CPP and MPP schemes. The current drawn by the clock network in CPP scheme is shown in Fig. 7. In Fig. 7 the signals (Clk, #Clk, Clk1, #Clk1, Clk2, #Clk2) show the clock at various stages of the tree distribution network. The peak current drawn from power supply line and peak discharge current to the ground line clearly coincide with the switching event of the clock applied

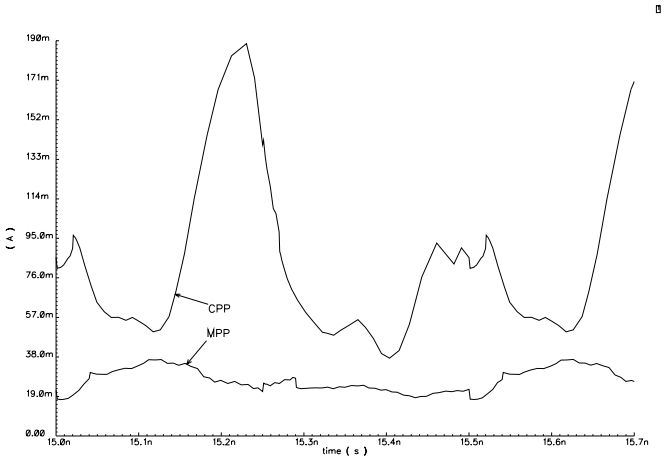

Fig. 9. Clock network current (from Vdd) at 2GHz.

to the pipeline registers. This is due to the large number of pipeline registers that have to be driven simultaneously in CPP scheme. The slew rate (*di/dt*) of the current from Vdd is approximately 1230V/µs. Similarly the slew rate of current discharged into the ground rail is approximately 1665V/µs. The large current drawn can induce a large *IR* drop on the supply network, while the large current slew rates (as shown in Fig. 7) can generate significant *Ldi/dt* drops. These drops are aggravated by technology scaling, decreasing supply voltages and increasing clock frequencies. These voltage fluctuations can be suppressed by increasing the on-chip decoupling capacitance, however this results in increased die size and cost. The current drawn by the clock network in MPP scheme is shown in Fig. 8. In Fig. 8 ClkReg1, ClkReg2, ClkReg3, ClkReg4 signals are the clock signals applied to the first, second, third and fourth register stages respectively. Due to the clock distribution approach taken in MPP, the registers are not triggered at the same time, which is clear from Fig. 8. Also, the current drawn by the clock distribution network is relatively small and has less variation compared to current in CPP scheme as shown in Fig. 9. This means significant power savings and less power supply noise.

V. Concluding Remarks

In this paper, the power performance gains from mesochronous pipeline (MPP) scheme over conventional pipeline (CPP) scheme have been presented. Following are features of the MPP scheme.

- *Smaller number of pipeline registers*. The performance achieved in CPP scheme can be easily achieved using MPP scheme with fewer pipeline stages and registers.
- *Simpler clock distribution*. The clock signal in the proposed scheme travels along with data, greatly reducing the complexity of clock distribution.
- *Low power dissipation*. A MPP implementation of an 8×8-bit carry-save adder multiplier using modest TSMC 180nm CMOS technology is dissipating less power compared to a CPP implementation. The average power dissipation in MPP implementation is 148.05mW, while in CPP implementation is 345.6mW.
- *Lower power supply noise*. In MPP scheme the variation in current drawn by clock network is significantly less, which means less power supply noise.
- *Shorter clock period* (T_{clk}). The clock period in MPP architecture is determined by the pipeline stage with the largest difference between its minimum and maximum propagation delay. So, smaller clock periods are possible.

References

[1] V. G. Oklobdzija *et. al.*, Digital System Clocking, Wiley-Interscience, 2002.

[2] F. Klass, *et. al.*, "A New Family of Semidynamic and Dynamic Flip-Flops with Embedded Logic for High-Performance Processors," *IEEE J. Solid-State Circuits*, vol. 34, no. 5, pp. 712 – 716, May 1999.

[3] P. J. Restle, and A. Deutsch, "Designing the best clock distribution network," *Symp. VLSI Circuits*, pp. 2 – 5, June 1998.

[4] S. Tam, R. D. Limaye, U. N. Desai, "Clock Generation and Distribution for the 130-nm Itanium 2 Processor with 6-MB On-Die L3 Cache," *IEEE J. Solid-State Circuits*, vol. 39, no. 4, April 2004 pp. 636 – 642.

[5] Y. I. Ismail, and E. G. Friedman, "Effects of Inductance on Propagation Delay and Repeater Insertion in VLSI Circuits," *IEEE Trans. VLSI Syst.*, vol. 8, no. 2, pp. 195 – 206, April 2000.

[6] S. Zhao, K. Roy, and C-K Koh, "Estimation of Inductive and Resistive Switching Noise on Power Supply Network in Deep Sub-micron CMOS circuits," *Proc. Intl. Conf. Comp. Design*, pp. 65 – 74, Sept. 2000.

[7] W. H. Lee, S. Pant, and D. Blaauw, "Analysis and Reduction of On-Chip Inductance Effects in Power Supply Grid," *Proc. 5th Intl. Symp. Quality Electronic Design*, pp. 131-136, 2004.

[8] D. E. Duarte, N. Vijaykrishnan, and M. J. Irwin, "A Clock Power Models to Evaluate Impact of Architectural and Technology Optimizations," *IEEE Trans. on VLSI Syst.*, vol. 10, no. 6, pp. 844 – 855, Dec. 2002.

[9] E. G. Friedman, "Clock Distribution Networks in Synchronous Digital Integrated Circuits," *Proc. IEEE*, vol. 89, no. 5, pp. 665 – 692, May 2001.

[10] S. B. Tatapudi and J. G. Delgado-Frias, "A pipelined multiplier using a hybrid-wave pipelining scheme," *Proceedings IEEE Computer Society Annual Symp. VLSI*, pp. 282 – 283, May 2005. Available: http://www.eecs.wsu.edu/~jdelgado/Hipercops_pubs.htm

[11] S. B. Tatapudi and J. G. Delgado-Frias, "Designing pipelined systems with a clock period approaching pipeline register delay," *48th IEEE Intl. Midwest Symp. Circuits Syst.*, Aug. 2005. Available: http://www.eecs.wsu.edu/~jdelgado/Hipercops_pubs.htm

[12] S. B. Tatapudi and J. G. Delgado-Frias, "A Mesochronous Pipelining Scheme for High-Performance Digital Systems," *IEEE Trans. Circuits Syst. I*, to be published.

[13] C. T. Gray, W. Liu, and R. K. Cavin, "Timing Constraints for Wave-pipelined Systems," *IEEE Trans. Computer-Aided Design*, vol. 13, no. 8, pp. 987 – 1004, Aug. 1994.

[14] W. P. Burleson, M. Ciesielski, F. Klass, and W. Liu, "Wave-Pipelining: A Tutorial and Research Survey," *IEEE Trans. VLSI Syst.*, vol. 6, no. 3, pp. 464 – 474, Sep. 1998.

[15] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits*, 2nd ed., Upper Saddle River: NJ, Prentice Hall, 2002.

[16] S. B. Tatapudi and J. G. Delgado-Frias, "A Mesochronous Pipeline Scheme for High Performance Low Power Digital Systems," *IEEE Intl. Symp. Circuits Syst.*, May 2006. Available: http://www.eecs.wsu.edu/~jdelgado/Hipercops_pubs.htm.