

# Efficient Clock Distribution Scheme for VLSI RNS-Enabled Controllers

Daniel González, Luis Parrilla, Antonio García,  
Encarnación Castillo, and Antonio Lloris

Department of Electronics and Computers Technology,  
University of Granada, 18071- Granada, Spain  
{dgonzal, lparrilla, grios, encas, lloris}@ditec.ugr.es

**Abstract.** Clock distribution has become an increasingly challenging problem for VLSI designs because of the increase in die size and integration levels, along with stronger requirements for integrated circuit speed and reliability. Additionally, the great amount of synchronous hardware in integrated circuits makes current requirements to be very large at very precise instants. This paper presents a new approach for clock distribution in PID controllers based on RNS, where channel independence removes clock timing restrictions. This approach generates several clock signals with non-overlapping edges from a global clock. The resulting VLSI RNS-enabled PID controller, shows a significant decrease in current requirements (the maximum current spike is reduced to a 14% of single clock distribution one at 125 Mhz) and a homogeneous time distribution of current supply to the chip, while keeping extra hardware and power to a minimum.

## 1 Introduction

The Proportional-Integral-Derivate (PID) controller is used to solve about 90-95% of control applications, including special applications requiring ultra-high precision control [1]. The controllers for the above applications are usually implemented as discrete controllers. The Residue Number System (RNS) [2] has shown itself as a valid alternative for supporting high-performance arithmetic with limited resources. In this way, RNS enables high-precision arithmetic-intensive applications using limited and modular building blocks. Concretely, when applied to PID controllers over Field Programmable Logic Devices (FPLDs), RNS has shown an improvement of 331% in speed over the binary solution [3]. On the other hand, implementation of this controllers on VLSI leads to difficulties in the proper synchronization of such systems. As clock distribution represents a challenge problem for VLSI designs, consuming an increasing fraction of resources such as wiring, power, and design time; an efficient strategy for clock distribution is needed.

For True Single Phase Clock (TSPC), which is a special dynamic logic clocking technique and should not be used as a general example, clock lines must be distributed all over the chip, as well as being distributed within each operating block. More complex clocking schemes may require the distribution of two or four non-overlapping clock signals [4], thus increasing the resources required for circuit synchronization. Moreover, for clock frequencies over 500 MHz, phase differences between the clock signal at different locations of the chip (skew) are presenting serious

problems [5]. An added problem with increasing chip complexity and density is that the length of clock distribution lines increases along with the number of devices the clock signal has to supply, thus leading to substantial delays that limit system speed. A number of techniques exists for overriding clock skew, with the most common being RC tree analysis. This method represents the circuit as a tree, modeling every line through a resistor and a capacitor, and modeling every block as a terminal capacitance [6]. Thus, delay associated with distribution lines can be evaluated and elements to compensate clock skew can be subsequently added. Minimizing skew has negative sides, especially as simultaneous triggering of so many devices leads to short but large current demands. Because of this, a meticulous design of power supply lines and device sizes is required, with this large current demand resulting in area penalties. If this is not the case, parts of the chip may not receive as much energy as required for working properly. This approximation to the problems related to fully synchronous circuits and clock skew has been previously discussed [7]; this paper will present an alternative for efficiently synchronizing RNS-based circuits while keeping current demand to a minimum. The underlying idea is to generate out-of-phase clock signals, each controlling an RNS channel, thus taking advantage of the non-communicating channel structure that characterizes RNS architectures in order to reduce the clock synchronization requirements for high-performance digital signal processing systems. In the present work, this synchronization strategy is applied to RNS-PID controllers [3] implemented on VLSI, achieving important reductions in current demand and current change rate.

## 2 Digital PID Controllers

The PID controller is described by the following equation [1]:

$$y(t) = K_p \left( x(t) + \frac{1}{T_i} \int_0^t x(\tau) d\tau + T_d \frac{dx(t)}{dt} \right) \quad (1)$$

where  $K_p$  is the proportional constant,  $T_i$  is the integral time and  $T_d$  is the derivative time. The discrete implementation of the controller is usually derived from the following approximations:

$$\int_0^t x(\tau) d\tau \approx \sum_{i=0}^{n-1} x[i]h \quad (2)$$

$$\frac{dx(t)}{dt} \approx \frac{x[n] - x[n-1]}{h}$$

where  $x[j]$  is the  $j$ th sample and  $h$  is the sampling period. Using (2) in equation (1), the discrete version of (1) is:

$$y[n] = y[n-1] + K_p (x[n] - x[n-1]) + \frac{K_p \cdot h}{T_i} x[n-1] + \frac{K_p \cdot T_d}{h} (x[n] - 2x[n-1] + x[n-2]) \quad (3)$$

Equation (3) may be rewritten more conveniently just defining the constants  $C_0$ ,  $C_1$  and  $C_2$ :

$$\begin{aligned} C_0 &= K_p \left( 1 + \frac{T_d}{h} \right) \\ C_1 &= K_p \left( \frac{h}{T_i} - 1 - 2 \frac{T_d}{h} \right) \\ C_2 &= \frac{K_p T_d}{h} \end{aligned} \quad (4)$$

Thus, the discrete version of the PID controller is:

$$y[n] = y[n-1] + C_0 x[n] + C_1 x[n-1] + C_2 x[n-2] \quad (5)$$

For a typical high-precision application, a 10-bit input may be considered, as well as 12-bit representations of the coefficients  $K_p$ ,  $K_i$ ,  $K_d$  and a 26-bit output.

### 3 Clock Skew

Clock skew occurs when the clock signal has different values at different nodes within the chip at the same time. It is caused by differences in the length of clock paths, as well as by active elements that are present in these paths, such as buffers. Clock skew lowers system throughput compared to that obtainable from individual blocks of the system, since it is necessary to guarantee the proper function of the chip with a reduced speed clock. Skew will cause clock distribution problems if the following inequality holds [8]:

$$\frac{D}{v} > \frac{k}{f_{app}} \quad (6)$$

where  $k < 0.20$  (typical value) is a constant,  $D$  is the typical size of the system,  $v$  is the propagation speed for the clock signal and  $f_{app}$  is the applied clock frequency. Existing solutions for clock skew provide two different approaches to the problem:

1. Equalize the length of clock paths to processing elements using buffer and delay elements or through H-tree, mesh or X-tree topologies [6, 9-12].
2. Eliminate or minimize the skew caused by variations during chip fabrication [13-14].

Typically, synchronous systems consist of a chain of registers separated by combinational logic that performs data processing. The maximum clock frequency is derived from:

$$\frac{1}{f_{\max}} = T_{\min} \geq T_{PD} + T_{skew} \quad (7)$$

where  $T_{PD}$  is the time between the arrival of the clock signal at the  $i$ -th register and stable processed data at the output of the  $(i+1)$ -th register.  $T_{skew}$  is the time between the arrival of the clock signal at the  $i$ -th register and the arrival of the same signal at the  $(i+1)$ -th register.

Clock skew can be considered as either positive or negative, although the sign criteria is not standardized. Hatamian [11] considers the skew to be positive when the clock signal arrives at the  $i$ -th register before that to the  $(i+1)$ -th register, as illustrated by Fig. 1. If positive, then from equation (7), the minimum system clock period is increased, while if negative,  $T_{\min}$  decreases. An excessive positive skew results in a decrease in system performance, but if the skew is negative race-related problems may arise if data processing time is lower than the skew.

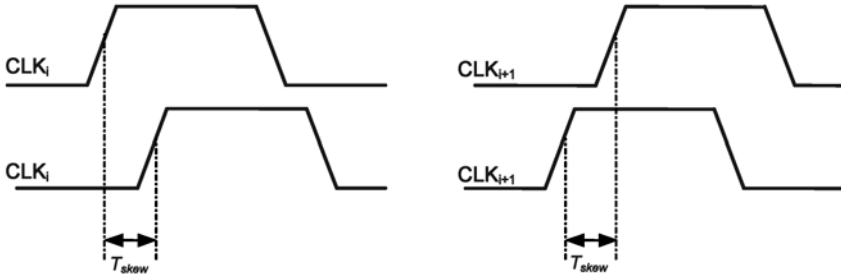


Fig. 1. Positive (left) and negative (right) skew

#### 4 Efficient Synchronization Scheme

The proposed synchronization scheme for RNS-based systems, introduces the generation of several signal clocks from the master clock. These clocks are slightly out-of-phase, thus with non-overlapping edges. Each one of these clock signals synchronizes one of the RNS channels, while global data synchronization (mainly at the global inputs and outputs of the system) is carried out by the global master clock. Thus, each channel computes at different time instants and, consequently, the current demand is distributed over the whole clock cycle. This has the effect of reducing current spikes on the power supply lines approximately by a factor that is the number of generated clock signals. The phase difference between the generated clocks has to satisfy some specifications. First of all, the number of clock signals with overlapping active cycles has to be minimized, as well as the time two or more active cycles overlap. Moreover, clock edges must not coincide. Finally, data coherency has to be respected at both the input and output of the system. Clear advantages are obtained when these requisites are satisfied, since current spikes are reduced and power dissipation is distributed over the master clock cycle, rather than concentrated around the master clock edges. Moreover, not only absolute current values are reduced, but also its temporal variation. Also, as a side effect, power supply lines may be scaled and clock distribution resources reduced, thus simplifying the chip design task.

At first sight, the synchronization scheme described above may seem to be impractical because of the presence of several clock signals within the chip, with associated synchronization problems. However, the nature of RNS [2,15], with non-communicating channels, perfectly suits this clocking scheme. Thus, a generated clock signal is applied to each independent channel, while the master clock signal used to generate these other clocks can also synchronize the global input and output. Moreover, it will be shown that the resources required for implementing this new strategy are minimum and a few transistors, basically three inverters, are required for

each channel. More specifically, the master clock signal is routed through an inverter chain, thus being delayed at every point of the chain. Meanwhile, the generated clock signals are extracted at adequate points of this chain and conditioned to be used as clock signal for a complete RNS channel. This scheme requires the inverter chain to alternate large and small input capacitances and low driving capabilities, so appropriate delays can be generated. This has the effect of generating low-quality clock signals within the inverter chain, so additional buffers are required in order to obtain adequate clock signals. Fig. 2 illustrates the hardware required to generate the proposed synchronization scheme, where  $dCLK$  stands for generated delayed clocks, while Fig. 3 shows the detailed scheme for the so-called  $dCLK\_cell$  cell, which consists of three inverters. It can be deduced from Fig. 3 that three design parameters,  $L_d$ ,  $W_b$  and  $L_b$ , are available in order to obtain the system specifications, while  $L_{min}$  represents the feature size of the fabrication process and  $W_{min}$  the minimum usual width for pMOS transistors. Connecting  $CH_{out}$  pads to  $CH_{in}$  pads, the inverter chain described above is built, while the master global clock is used as input to this chain. Fig. 3 illustrates how large capacitance inverters are alternated with minimum-size devices. Thus, the low driving capabilities of the latter allow modeling of the required delay using the  $L_d$  parameter. Meanwhile, the generated clock signal  $dCLK$  is regenerated by a third inverter that includes the parameters  $W_b$  and  $L_b$ . These allow matching of the timing specifications for a proper clock signal for a given system, also allowing the adaptation of the cell to the overall capacitance to be driven by the generated clock. However, these three design parameters  $L_d$ ,  $W_b$  and  $L_b$  are not fully independent, and their relation needs a careful study of the final system to be synchronized in order to select their optimum values. Fig. 4 shows the resulting generated clocks in a simple design example for a 300 MHz master global clock, with 0.1 pF loads for every  $dCLK$  signal. It can be noted that the requirements enumerated above about non-overlapping edges and active cycles are matched, with every  $dCLK$  signal being to be used as clock for a given RNS channel.

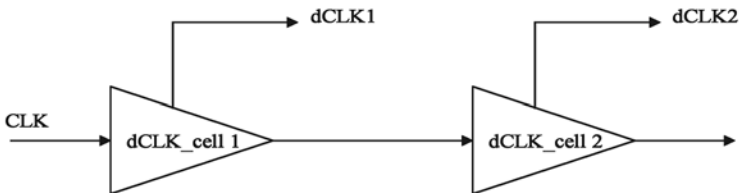


Fig. 2.  $dCLK\_cell$  chain for out-of-phase clock generation

## 5 Design Example

A real RNS-based processing application [2] was considered for the evaluation of the proposed synchronization technique. Specifically, a fast PID Controller with 26-bit dynamic range was designed at the transistor level simulated using PSpice for both a single global clock and the proposed technique. For these simulations, a public domain MOSIS CMOS 0.6  $\mu\text{m}$  process [16] was used. This is a three-metal, one-poly, 3.3V CMOS process that is available to MOSIS costumers through Agilent Technologies. RNS [15] have been shown to be a useful alternative for binary implemen-

tations for a variety of digital processing applications. Concretely, high performance PID controllers can be designed taken advantage of RNS properties [3].

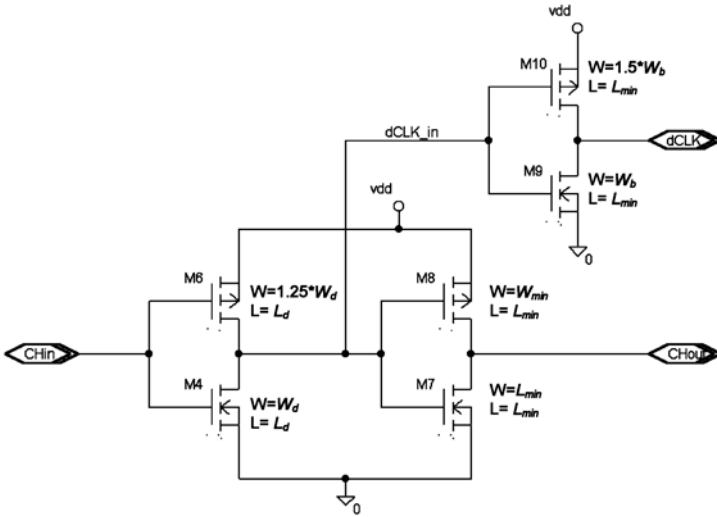


Fig. 3. dCLK\_cell schematic

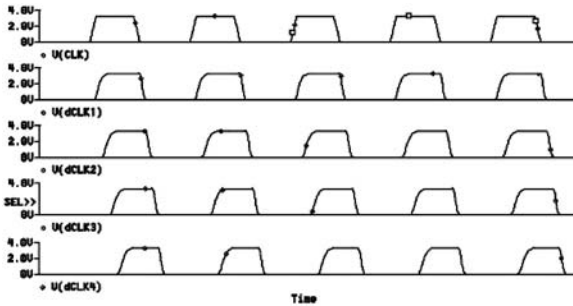


Fig. 4. Resulting generated clock signals for a design example (125 MHz)

For a typical high-precision application, a 12-bit input may be considered, as well as 16-bit representations of the coefficients  $C_0, C_1, C_2$  (5). Thus, it is possible to obtain a 10-bit output without round errors. This RNS-enabled system requires four channels with moduli  $\{256, 63, 61, 59\}$ . Each one of this channels includes LUTs for fixed coefficient multiplications, adders, and a compensated modulo accumulator for synchronization of datasets. Fig. 5 shows the structure of this output accumulator. Since the system is composed just of adders, tables and registers, the well-known two-stage modulo adder [3] was used, while registers were implemented using negative edge triggered D flip-flops (nETDFF) based on TSPC logic [1]. TSPC was selected because it only requires a single-phase clock, thus minimizing synchronization resources and simplifying the implementation of the proposed alternative. Because of the great connection locality for the example system, load driving is kept to a minimum and device sizes can be fixed to the process minimum for most of the transistor

involved. Only transistors involved in clock management will have larger sizes since they have to drive large loads. The systems under simulation include around 50.000 transistors.

In order to get illustrative comparison results, the RNS-enabled PID controller was simulated under two different clocking strategies: first of all, a single global clock used to synchronize the whole circuit, using a train pulse voltage source; and second, the proposed strategy.

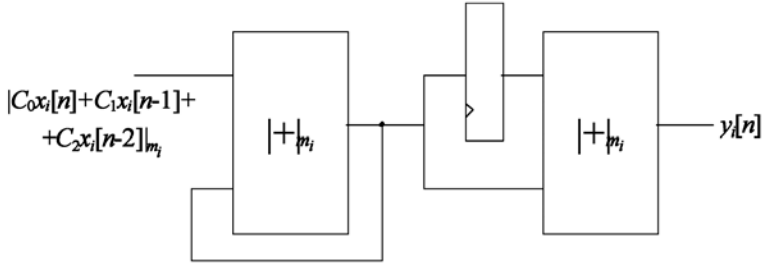


Fig. 5. Corrected modulo  $m_i$  accumulator for PID applications

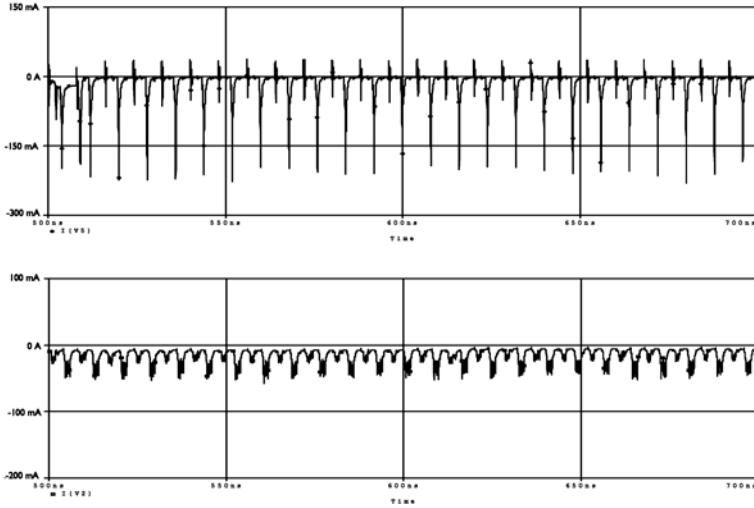
Since the RNS-enabled PID controller requires four channels, as mentioned above, the proposed design example was synchronized using four `dCLK_cell` cells and four generated `dCLK` signals, each one synchronizing an RNS channel. The design parameters for the `dCLK_cell` cells, after careful selection, were fixed at  $L_d=1 \mu\text{m}$ ,  $W_d=2 \mu\text{m}$  and  $W_b=9 \mu\text{m}$ . These two alternatives have been simulated for two different clock frequencies, 125 MHz and 300 MHz. A comparison between the proposed strategy and the buffered clock simulation will illustrate the affordable power penalty introduced by this new clocking strategy.

Fig. 6 shows the current on power supply lines for the PID full system working with a 125 MHz clock for both a single clock and the proposed synchronization scheme. Fig. 7 shows the corresponding currents when a 300 MHz clock is applied. Clearly evident is the considerable decrease in the magnitude of the current spikes. In this way, current supply to the chip is distributed over time when the new strategy is considered, while for a global clock current spikes are around four times larger. This indicates that the expected benefits derived from the proposed synchronization scheme are confirmed through simulation. Table 1 summarizes the results obtained for the different simulations and both clock frequencies. We note that the maximum current spike is clearly reduced when this new clocking strategy is considered, as well as the maximum value of the current change rate ( $di/dt$ ). Finally, if power dissipation is considered, the comparison between the single clock and the proposed strategy shows that the latter introduce an affordable increase in power.

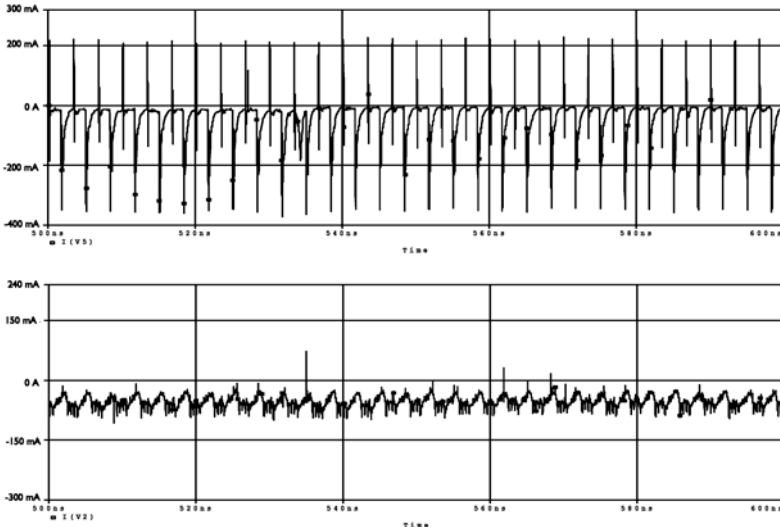
## 6 Conclusions

This paper has presented a new alternative for synchronizing RNS-based systems and reducing current demand. The proposed strategy was tested using an RNS-Enabled PID controller consisting of around 50.000 transistors. Simulation results demonstrate the effectiveness of this new clocking strategy in reducing the maximum current

spike as well as reducing the maximum time derivative of the current spike. Concretely, the maximum current spike is reduced to 14-23% of the single clock strategy one at 125-300MHz, and the time derivate to 3-1%. On the other hand, the power penalty introduced by the new scheme is clearly affordable. Thus, the use of this synchronization scheme may lead to reduced skew-related problems as well as to reducing chip area through the reduction of the size of power supply lines, caused by the reduction in current and current change rate requirements.



**Fig. 6.** Current from power supply line for a single clock (above) and the proposed alternative (below) for a 125 MHz frequency



**Fig. 7.** Current from power supply line for a single clock (above) and the proposed alternative (below) for a 300 MHz frequency



**Table 1.** Simulation results for RNS-Enabled PID controller using different synchronization approaches

	Single clock		Proposed strategy	
	125 MHz	300 MHz	125 MHz	300 MHz
Max Spike	275.06 mA	348.4 mA	38.84 mA	82.143 mA
Max di/dt	45.11 A/ns	261.75 A/ns	1.67 A/ns	2.47 A/ns
Power	55.37 mW	119.67 mW	66.95 mW	146.11 mW

**Acknowledgements.** Authors wish to acknowledge financial support from Investigation General Direction (Spain) under project TIC2002-02227. The CAD tools were provided by Mentor Graphics Inc, through their university program.

## References

1. Aström, K.J., Hägglund, T.: PID Control. Theory, Design and Tuning, 2nd ed., Instrument Society of America. Research Triangle Park, NC (1995).
2. Szabo, N.S., Tanaka, R.I.: Residue Arithmetic and Its Applications to Computer Technology, McGraw-Hill, NY (1967).
3. Parrilla, L., García, A., Lloris, A.: Implementation of High Performance PID Controllers using RNS and Field-Programmable Devices, Proc. of 2000 IFAC Workshop on Digital Control PID'00 (Terrassa, Apr. 5-7 2000), (2000) 628-631.
4. Yuan, J., Svensson, C.: High-speed CMOS Circuit Technique, IEEE Journal of Solid State Circuits, Vol 24, No. 1, (1989) 62-70.
5. Bailey D.W., Benchnneider, B.J.: Clocking Design and Analysis for a 600-MHz Alpha Microprocessor, IEEE Journal of Solid State Circuits, Vol 33, (1998) 1627-1633.
6. Ramanathan, P., Dupont A.J., Shin, K.G.: Clock Distribution in General VLSI Circuits. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Vol 41, No. 5, (1994) 395-404.
7. Yoo, J., Gopalakrishnan G., Smith, K.F.: Timing Constraints for High-speed Counterflow-clocked Pipelining, IEEE Transactions on VLSI Systems, Vol. 7, No. 2, (1999) 167-173.
8. Grover, W.D.: A New Method for Clock Distribution. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Vol 41, No. 2, (1994) 149-160.
9. Jackson, M.A.B., Srinivasan, A., Kuh, E.S.: Clock Routing for High Performance IC's. 27th ACM/IEEE Design Automation Conference (1990).
10. Wann, D.F., Franklin, N.A.: Asynchronous and Clocked Control Structures for VLSI Based Interconnect Networks. IEEE Transactions on Computers, Vol 32, No.5, (1983) 284-293.
11. Hatamian, M.: Chapter 6, Understanding clock skew in synchronous systems. In Concurrent Computations (Algorithms, Architecture, and Technology). S.K. Tewksbury, B.W. Dickinson, and S.C. Schwartz (Eds.), Plenum Publishing, New York, (1988) 87-96.
12. Friedman, E.G.: Clock Distribution Networks in Synchronous Digital Integrated Circuits. Proceedings of the IEEE, Vol. 89, No. 5, (2001).
13. Friedman, E.G., Powell, S.: Design and Analysis of an Hierarchical Clock Distribution System for Synchronous Cell/macrocell VLSI. IEEE Journal of Solid State Circuits, Vol. 21, No. 2, (1986) 240-246
14. Shoji, M.: Elimination of Process-dependent Clock skew in CMOS VLSI. IEEE Journal of Solid State Circuits, vol. 21, (1986) 869-880.
15. Soderstrand, M.A., Jenkins, W.K., Jullien G.A., Taylor, F.J.: Residue Number System Arithmetic: Modern Applications in Digital Signal Processing. IEEE Press (1986).
16. MOSIS Process Information: Hewlett Packard AMOS14TB, <http://www.mosis.org/technical/processes/proc-hp-amos14tb.html>