Clocking and Clocked Storage Elements in Multi-GHz Environment

Vojin G. Oklobdzija[1], Fellow IEEE

Integration Corp. Berkeley, California, USA
http://www.integration-corp.com
[1] Department of Electrical Engineering, University of California, Davis
vojin@ece.ucdavis.edu
http://www.ece.ucdavis.edu/acsel

Abstract. An overview of clocking and design of clocked storage elements is presented. Systematic design of Flip-Flop is explained as well as "time borrowing" and absorption of clock uncertainties. We show how different clocked storage elements should be compared against each other. The issues related to power consumption and low-power designs are presented.

1 Introduction

Deciding on the clocking strategy in digital system is one of the single most important decisions. If not considered properly at the beginning of a design it can be very costly

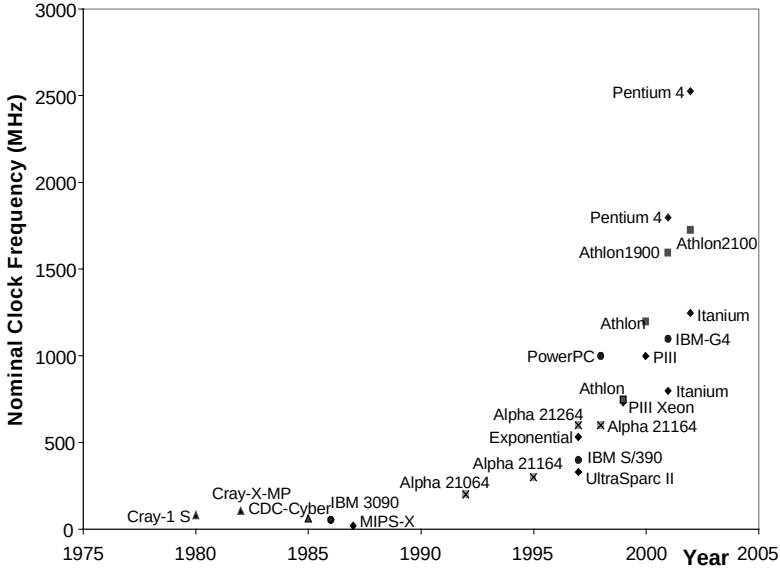

Fig. 1. Clock frequency over the years for various representative machines

B. Hochet et al. (Eds.): PATMOS 2002, LNCS 2451, pp. 128–145, 2002.

afterwards. The importance of clocking is gaining momentum as the clock speed rises rapidly; doubling every three years as shown in Fig.1. At today's frequencies ability to absorb clock skew and to use faster Clocked Storage Element (CSE), results in direct performance improvement. Those improvements are very difficult to obtain through architectural techniques or micro-architecture level. As the clock frequency reaches 5-10GHz traditional clocking techniques will be reaching their limit. New ideas and new ways of designing digital systems are required.

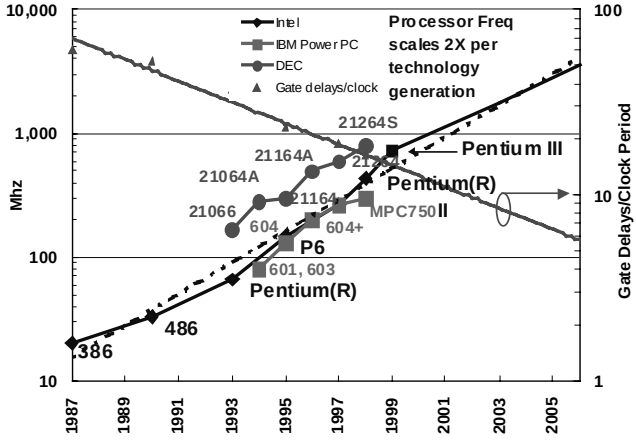

Fig. 2. Increase in the clock frequency and decrease in the number of logic levels in the pipeline (courtesy of Intel Corp.)

Following the speed increase, the number of logic levels in the *critical path* diminishes. In today's high-speed processors, instructions are executed in one-cycle, which is driven by a single-phase clock. In addition the pipeline depth is increasing to 15 or 20 in order to accommodate the speed increase. Today 10 levels of logic in the critical path are more common and this number is expected to be decreasing further as illustrated in Fig. 2. Thus any overhead associated with the clock system and clocking mechanism that is directly and adversely affecting the machine performance is critically important.

1.1 Clock Distribution

The two most important timing parameters affecting the clock signal are: *Clock Skew* and *Clock Jitter*:

Clock Skew is a spatial variation of the clock signal as distributed through the system. It is caused by the various RC characteristics of the clock paths to the various points in the system, as well as different loading of the clock signal at different points on the chip. Further we can distinguish *global clock skew* and *local clock skew*. Both of them are equally important in high-performance system design.

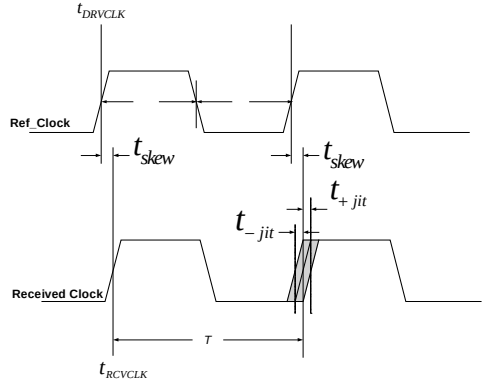

Fig. 3. Clock Parameters: Period, Width, Clock Skew and Clock Jitter

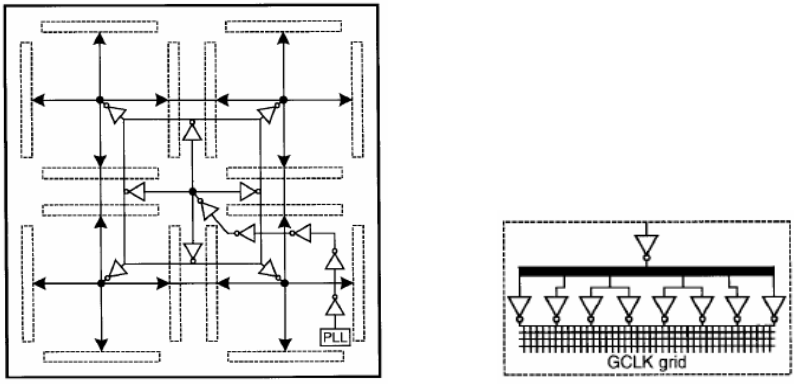

Fig. 4. Clock distribution methods: (a) an RC matched tree and (b) a grid [2]

Clock Jitter is a temporal variation of the clock signal with regard to the reference transition (reference edge) of the clock signal as illustrated in Fig. 3. Clock jitter represents edge-to-edge variation of the clock signal in time. As such clock jitter can also be classified as: long-term jitter and edge-to-edge clock jitter, which defines clock signal variation between two consecutive clock edges. In the course of high-speed logic design we are more concerned about edge-to-edge clock jitter because it is this phenomena that affects the time available to the logic.

Typically the clock signal has to be distributed to several hundreds of thousands of the clocked storage elements (also known as flip-flops and latches). levels of amplification (buffering). As a consequence, the clock system by itself can Therefore, the clock signal has the largest fan-out of any node in the design, which requires several use up to 40-50% of the power of the entire VLSI chip [1]. We also must assure that every clocked storage element receives the clock signal precisely at the same moment in time.

There are several methods for the on-chip clock signal distribution attempting to minimize the clock skew and contain the power dissipated by the clock system [18]. The clock can be distributed in several ways of which the two typical cases are: (a) an RC matched tree and (b) a grid shown in Fig. 4.

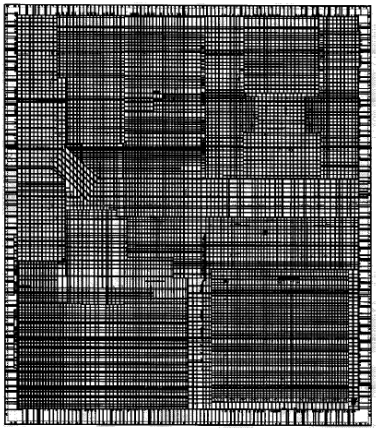

Fig. 5. Clock distribution grid used in DEC Alpha 600MHz processor [2], courtesy of IEEE.

If we had superior Computer Aided Design (CAD) tools, a perfect and uniform process and ability to route wires and balance loads with a high degree of flexibility, a matched RC delay clock distribution (a) would be preferable to grid (b). However,

neither of that is true. Therefore grid is used when clock distribution on the chip has to be very precisely controlled. This is the case in high performance systems.
An example of the clock distribution grid is shown in Fig. 5 [2]. The power consumed by the clock is also the highest in cases using grid arrangement. This is not difficult to understand given that in a grid arrangement a high-capacitance plate has been driven by buffers connected at various points. Local variations in device geometry and supply voltage are important component of the clock skew. More sophisticated clock distribution than simple RC matched or grid-based schemes are thus necessary. The active schemes with adaptive digital deskewing typically reduce clock skew of the simple passive clock networks by an order of magnitude, allowing tighter control of the clock period and higher clock rates [3].

2 Clocked Storage Elements

The function of a *clocked storage element*: flip-flop or latch, is to capture the information at a particular moment in time and preserve it as long as it is needed by the digital system. It is not possible to define a storage element without defining its relationship to the *clock*.

2.1 Master-Slave Latch

In order to avoid the *transparency* feature associated with a single latch, an arrangement is made in which two latches are clocked back to back with two non-overlapping phases of the clock. In such arrangement the first latch serves as a "*Master*" by receiving the values from the Data input and passing them to the "*Slave*" latch, which simply follows the "*Master*". This is known as a Master-Slave (M-S) Latch or L1 – L2 latch (in IBM) as shown in Fig. 6. This is not to be confused with the "*Flip-Flop*", though many practitioners today do call the configuration shown in Fig. 6. (b), a Flip-Flop (F-F). We distinguish Flip-Flop from M-S Latch. We will explain the fundamental differences between the F-F and M-S Latch in this paper.
In a Master-Slave Latch the "Slave" latch can have two or more masters acting as an internal multiplexer with storage capabilities. The first "*Master*" is used for capturing of data input while the second Master can be used for other purposes such as scan-

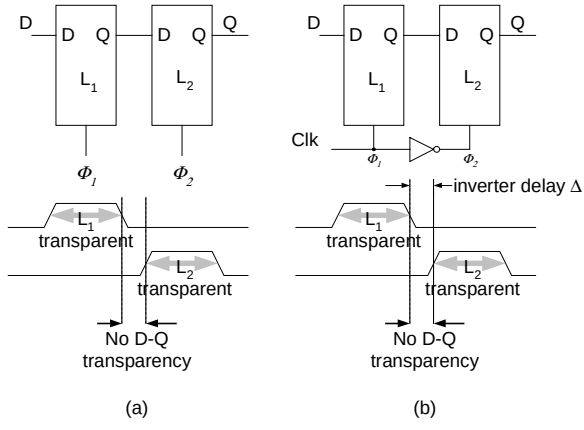

Fig. 6. Master-Slave Latch arrangement with: (a) non-overlapping clocks (b) single external clock.

input for testing purposes, and clocked with a separate clock. One such topology, utilizing two Masters, is a well-known IBM Level-Sensitive-Scan-Design [4].

2.2 Flip-Flop

Flip-Flop and Latch operate on different principles. While Latch is *"level-sensitive"* which means it is reacting on the *level* (logical value) of the clock signal, Flip-Flop is *"edge sensitive"* which means that the mechanism of capturing the data value on its input is related to the changes of the clock. Thus, the two are designed for a different set of requirements and thus consist of inherently different circuit topology. Level sensitivity implies that the latch is capturing data value during the entire period of time when clock is active (logic one) while the latch is *transparent*. The capturing process in the Flip-Flop occurs only during the transition of the clock, thus the Flip-Flop is *non-transparent*. However, even the Flip-Flop could have a small period of transparency associated with the narrow window during which the clock changes.

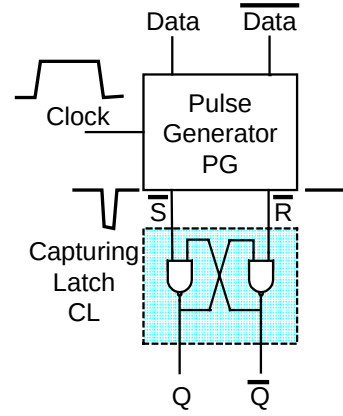

Fig. 7. General Flip-Flop structure

A general structure of the Flip-Flop is shown in Fig. 8. The difference between a Flip-

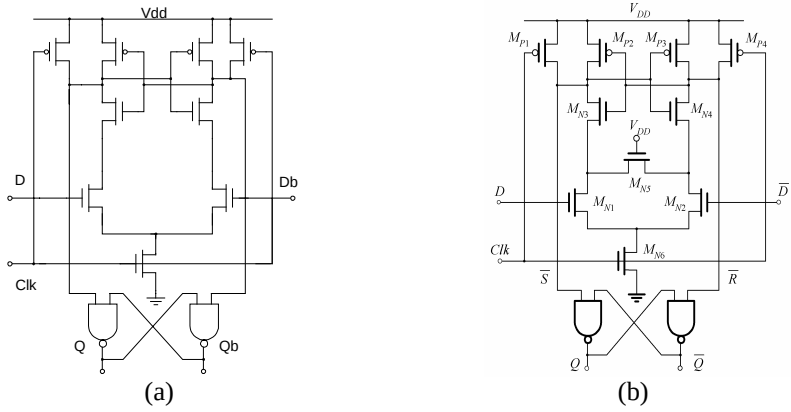

Fig. 8. Pulse Generator stage of the Sense Amplifier Flip-Flop: (a) Madden and Bowhill [5], (b) Improvement for floating nodes, Doberpuhl [9].

Flop structure (Fig. 8) and that of the M-S Latch (Fig. 6) should be noticed. A Flip-Flop consists of two stages: (a) Pulse Generator - PG and (b) Capturing Latch – CL. The pulse generator PG generates a negative pulse on either $\overline{S}$ or $\overline{R}$ lines, which are normally held at logic "one" level. This pulse is a function of Data and Clock signals

and should be of a sufficient duration to be captured in the capturing latch CL. The duration of that pulse can be as long as half of the clock period or it can be as short as one inverter delay. On the contrary M-S Latch generally consists of two identical clocked latches and its non-transparency feature is achieved by non-overlapping clocks ϕ_1 and ϕ_2, clocking master latch L_1 and slave latch L_2. The relationship of S and R signals with respect to Data (D) and Clock (Clk) signal can be expressed as:

$$S_n = Clk\overline{R}(D+S) \text{ and } \quad R_n = Clk\overline{S}(\overline{D}+R) \tag{1}$$

Those two equations (1) form a basis for derivation of a Flip-Flop structure.
Simply stated, the equation for S_n tells us that: *The next state of this Flip-Flop will be set to "1" only at the time the clock becomes "1" (raising edge of the clock), the data at the input is "1", the flip flop is in the "steady state" (both S and R are "0"). The moment Flip-Flop is set (S=1, R=0) no further change in data input can affect the Flip-Flop state: data input will be "locked" to set by (D+S)=1, and reset R_n would be disabled (by S=1).*

This assures the "*edge sensitivity*" – i.e. after the transition of the clock and setting of the S or R signal to its desired state, the Flip-Flop is "locked" for receiving a new data.

It is interesting that it took engineers several attempts to come to the right circuits topology of this Flip-Flop. The Flip-Flop used in the third generation of Digital Equipment Corp. 600MHz Alpha [1] processor used a version of the Flip-Flop introduced by Madden and Bowhill, which was based on the static memory cell design [5]. This particular Flip-Flop is known as Sense Amplifier Flip-Flop (SAFF), shown in Fig.8.a,b. Development of the Pulse Generator block of this Flip-Flop is illustrated in Fig. 8. A substantial improvement in speed is achieved by modification of the second stage by Stojanovic (US Patent No. 6,232,810) [6].

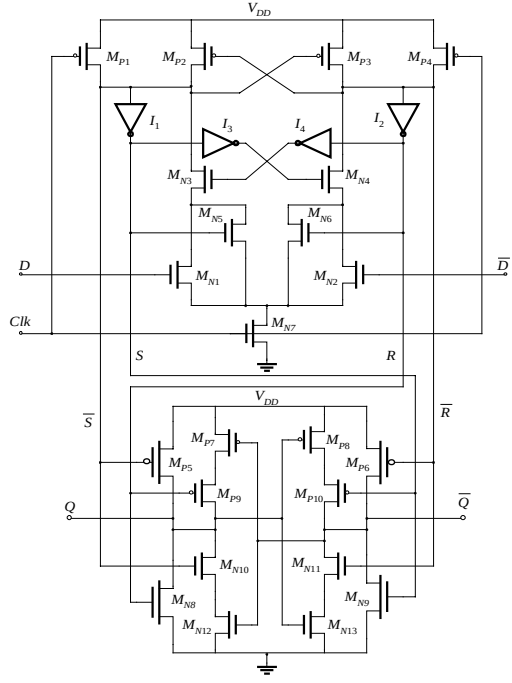

Fig. 9. Pulse Generator stage of the Sense Amplifier Flip-Flop: improvement by proper design: second stage (Stojanovic, US Patent: 6,232,810), first stage [10].

2.3 Time Window Based Flip-Flops

Digital circuits are based on discrete time events. The time reference is a clock signal and/or finite delay through one or more logic elements. To generate a needed time reference, a pulse created by the property of *re-convergent fan-outs, with non-equal parities of inversion* is commonly used. This method is illustrated in Fig. 10. on HLFF flip-flop introduced by Partovi [7]. The trailing edge of this short pulse is used as a time reference for shutting the Flip-Flop off. A short *"Time Window"* is created during which Flip-Flop is accepting data, which is the way of creating *"edge"* in digital world. Rigorous analysis of HLFF shows design incompleteness resulting in imperfections of the 1-1 output transition, which was demonstrated later.

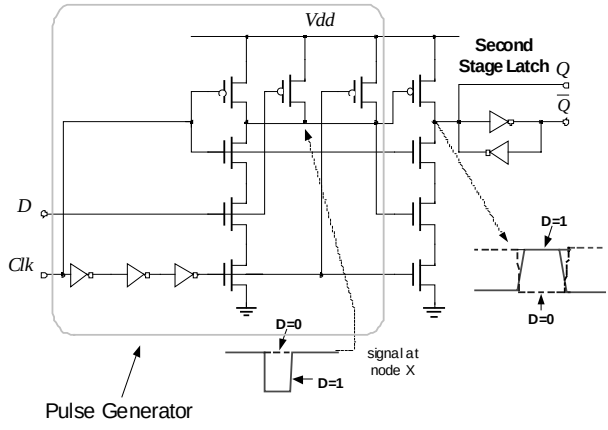

Fig. 10. Hybrid-Latch Flip-Flop introduced by Partovi [7]

A Flip-Flop based on the same principle was introduced by Klass [8], Fig. 11.
It uses a NAND gate to inhibit any further changes, and lock the existing ones after the time window has elapsed. It is characterized with one of the highest performance but suffers the same output glitch problem of HLFF. The problem is in the floating output node, which is susceptible to glitches and even slightest mismatch of clock signals.

A systematic approach in deriving a single-ended Flip-Flop is shown in Fig. 12. This Flip-Flop has three time reference points: (a) Clock signal: Clk (b) Clock signal passed through three inverters: Clk_3, (c) Clock passed through two inverters: Clk_2. The equations describing the pulse generator stage of this Flip-Flop is given by:

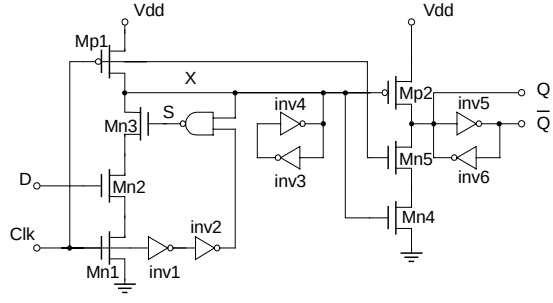

Fig. 11. Semi-Dynamic Flip-Flop: SDFF [8]

$$\overline{S} = X = \overline{(Clk + CLK_2) * (D * Clk_3 + \overline{X})} \tag{2}$$

The nMOS transistor section is a full realization of this equation. The pMOS section is somewhat abbreviated for performance reasons to:

$$X = \overline{(Clk + CLK_2) * (Clk_3 + \overline{\overline{X}})} \tag{3}$$

The second stage(capturing latch) is implemented as:

$$Q = \overline{X * (CLK_2 + \overline{\overline{Q}})} \tag{4}$$

This systematically derived Flip-Flop [11] does not have hazards in the output stage and is outperforming HLFF [7] and SDFF Flip-Flops [8].

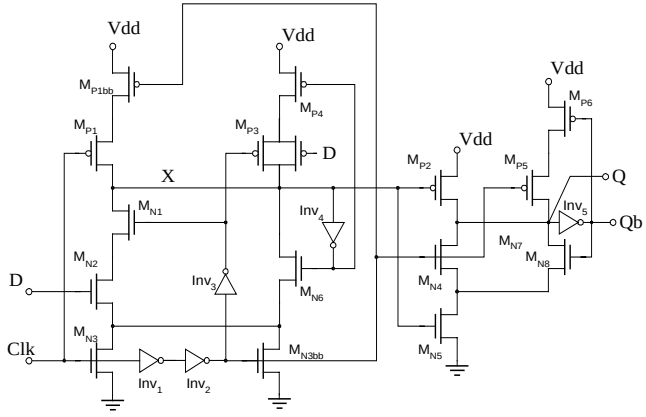

Fig. 12. Systematically derived single-ended Flip-Flop [11]

2.4 Pulsed Latches

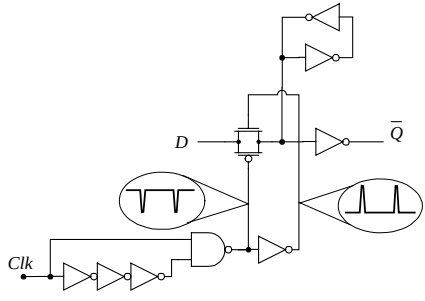

Fig. 13. Pulse Latch: Intel's Explicit Pulsed Latch [24]

In order to decrease the time overhead imposed by M-S Latch, or a Flip-Flop some designers resort to using Single Latch. To narrow the transparency window of the latch, they are clocked with short pulses generated locally from the global clock signal. Thus, the possibility of hold time violation and "races" (short paths) is not entirely eliminated, but it is traded for the convenience of a single latch and lower pipeline overhead. Given that the clock pulse is short, the hazard could be reduced by "padding" the logic, i.e. adding inverters in the fast paths so to eliminate the problem.

The clock produced by local clock generator must be wide enough to enable the Latch to capture its data. At the same time it must be sufficiently short to minimize the possibility of "critical race". Those conflicting requirements make use of such single-latch design hazardous by reducing the robustness and reliability of such design. Nevertheless, such design has been used due to the critical need to reduce cycle overhead imposed by the clocked storage elements. Intel's version of Pulsed Latch is

shown in Fig. 13. Additional benefit of this design is low power consumption due to the common clock signal generator and a simple structure of the latch. This power can be traded for speed. Pulse generator used in Intel's Pulsed Latch uses the principle of re-convergent fan-out with non-equal parity of inversion in order to obtain desired short clock pulse.

Analysis of the Pulsed Latch Timing Conditions

The conditions for reliable operation of a system using a single Latch are described in the paper by Unger and Tan [25] an given by Eqs. (5), (6) and (7):

$$P_m = P \geq D_{LM} + D_{CQM} + T_L + T_T + U - W \tag{5}$$

$$P \geq D_{LM} + D_{DQM} \tag{6}$$

$$D_{Lm} > D_{LmB} \geq W + T_T + T_L + H - D_{CQm} \tag{7}$$

One can notice from Eq. (5) that the increase of the clock width W is beneficial for speed, but it increases the minimal bound for the fast paths Eq. (7). Maximum useful value for W is obtained when the period P is minimal Eq. (6). Substituting P from Eq. (6) into Eq. (5) yields optimal value of W:

$$W^{opt} = T_L + T_T + U + D_{CQM} - D_{DQM} \tag{8}$$

If we substitute the value of the optimal clock width W^{opt} into (5), then we will obtain the values for the maximal speed (6) and minimal signal delay in the logic (4.26) which has to be maintained in order to satisfy the conditions for optimal single-latch system clocking:

$$D_{LmB} = 2(T_T + T_L) + H + U + D_{CQM} - D_{CQm} - D_{DQM} \tag{9}$$

Equation (6) tells us that in a single Latch system, it is possible to make the clock period *P* as small as the sum of the delays in the signal path: Latch and critical path delay in the logic block. This can be achieved by adjusting the clock width *W* and assuring that all the fast paths in the logic are larger in their duration than some minimal time D_{LmB}. In practice the optimal clock width W^{opt} is very small and does support the use of Pulsed-Latches.

$$W^{opt} \approx 2T_{SKW} \qquad D_{LmB} = 4T_{SKW} + H - D_{CQm} \tag{10}$$

Equation (9) tells us is that under ideal conditions, if there are no clock skews and no process variations, the fastest path through the logic has to be greater than the sampling window of the Latch (*H+U*) minus the time the signal spend traveling through the Latch. If the travel time through the Latch D_{DQM}, is equal to the sampling window, than we do not have to worry about fast paths.

3 Timing Parameters

Data and *Clock* inputs of a clocked storage element need to satisfy basic timing restrictions to ensure correct operation of the flip-flop. Fundamental timing constraints between data and clock inputs are quantified with *setup* and *hold* times, as illustrated in Fig. 14. Setup and hold times define time intervals during which input has to be stable to ensure correct flip-flop operation. The sum of setup and hold times define the "*sampling window*" of the clocked storage element.

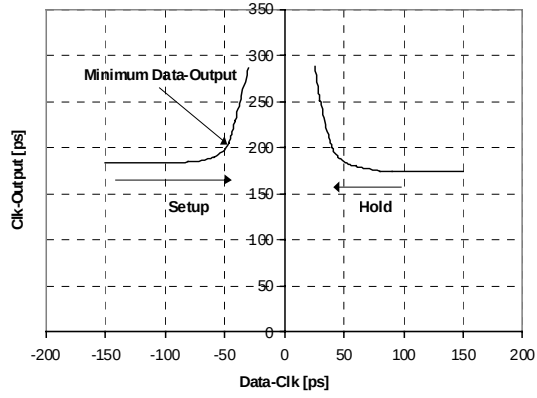

Fig. 14. Setup and Hold time behavior as a function of Clock-to-Output delay

3.1 Setup and Hold Time Properties

Failure of the clocked storage element due to the Setup and Hold time violations is not an abrupt process. This failing behavior is shown in Fig. 14. Considering how close should data be allowed to change with respect to the locking event, we encounter two opposing requirements:
– it should be kept further from the failing region for the purpose of design reliability.
– it should be as close to the clock in order to increase the time available for the logic operation.

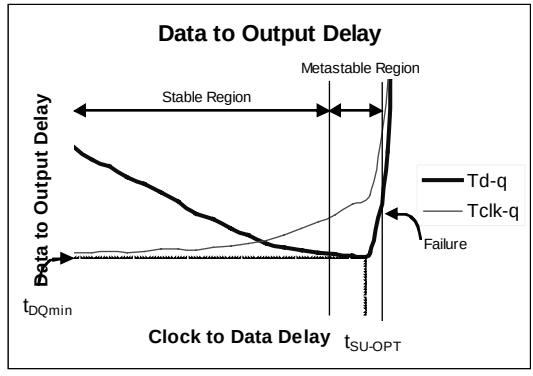

Fig. 15. Setup and Hold time behavior as a function of Data-to-Output delay

This is an obvious dilemma. In some designs an arbitrary number of 5-20% is used. Setup and Hold times are defined as points in time when the Clk-Q (t_{CQ}) delay raises for that amount. We do not find this reasoning to be valid.
A redrawn picture, Fig.15, where D-Q (t_{DQ}) delay is plotted (instead of Clk-Q), provides more information. From this graph we see that in spite of Clk-Q delay rising, we are still gaining because the time taken from the cycle is reduced.

3.2 Time Borrowing and Absorption of Clock Uncertainties

Even if data arrives close to the clock edge or pass the clock edge, the delay increase due to the storage element is still smaller than the amount of delay introduced into the next cycle. This allows for more time to be spent on useful logic operation in the previous cycle. This is known as: "*time borrowing*", "*cycle stealing*" or "*slack passing*". In order to understand the full effects of delayed data arrival we have to consider a pipelined design where the data captured in the first clock cycle is used as input in the next clock cycle as shown in Fig. 17.

As it can be seen in Fig. 17, the "*sampling window*" moves around the time axes. The "*sampling window*" is defined as the sum of the Setup and Hold times, i.e. the time period in which clocked storage element is "*sampling*" and data is not allowed to change. As the data arrive closer to the clock, the size of the "*sampling window*" shrinks (up to the optimal point). Even though, the sampling window is smaller, the data in the next cycle will still arrive later compared to the case where the data in the previous cycle was ahead of the setup-time. The amount of time for which the T_{CR1} was augmented did not come for free. It was simply taken away ("*stolen*" or "*borrowed*") from the next cycle T_{CR2}. As a result of late data arrival in the Cycle 1 there is less time available in the Cycle 2. Thus a boundary between pipeline stages is somewhat flexible. This feature not only helps accommodate a certain amount of imbalance between the critical paths in various pipeline stages, but it helps

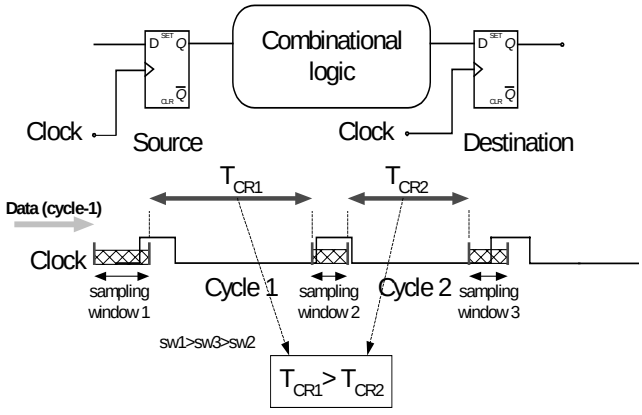

Fig. 16. "*Time Borrowing*" in a pipelined design

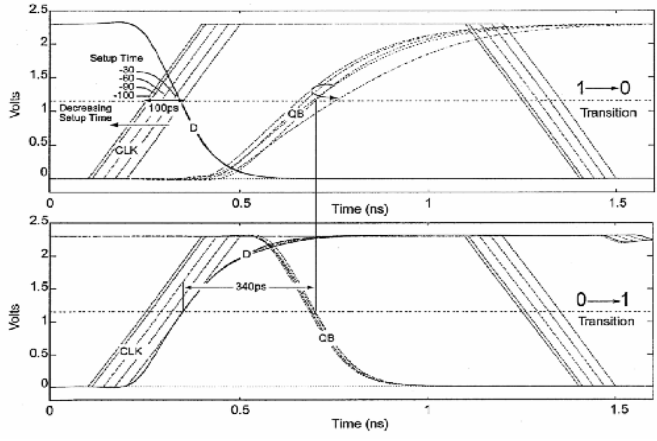

Fig. 17. Clock jitter-absorbing properties of HLFF [7].

in absorbing the clock uncertainties: *skew* and *jitter*. Thus, "*time borrowing*" is one of the most important characteristics of today's high-speed digital systems. Absorption of the clock jitter in HLFF is shown in Fig. 17 as observed by Partovi [7].
The maximal clock skew that a system can tolerate is determined by clock storage elements. If the clock-to-output delay of a clocked storage element is shorter than the hold time required and there is no logic in between two storage elements, a race condition can occur. A *minimum delay restriction* on the clock-to-output delay given by:

$$t_{CLK-Q} \geq t_{hold} + t_{skew} \tag{11}$$

If this relation is satisfied, the system is immune to hold time violations. Otherwise, it is necessary to check that all the timing paths have some minimal delay, which assures that there is no hold time violation.

4 Characterization

4.1 Power and Energy

It is important to emphasize the sources of power consumed in the Clocked Storage Element (CSE) and the correct set-up for the characterization and comparison. Power consumed by a CSE comes from various sources of which power-supply (V_{DD}) is only one of several. Using V_{DD} as a point for measuring power consumption can be misleading. Some CSE, characterized with low internal power consumption, represent a considerable load on the clock distribution network, thus taking considerable amount of power from the clock. Power can be drown from the Data input as well. Therefore the total power P_{tot} should account for all the possible power sources supplying the CSE [12].

$$\mathrm{P_{tot}} = \mathrm{P_{internal}} + \sum_{\text{inputs(D,CLK)}} \mathrm{P_{driver}} \tag{12}$$

4.2 Delay

In characterizing delay it is only appropriate to take into account the amount of time taken from the cycle T due to the insertion of the CSE. This represents $D-Q$ delay (t_{DQ}) as it was discussed in III. The question is whether this delay should be $D-Q$, $D-\overline{Q}$ or the worse of the two? We strongly argue that it is the most appropriate to characterize the CSE with the worse of the two delays since the critical path in a design may impose that scenario. Another question is that of the output load: how large should the outputs load be?. It is only reasonable that the load on the output: $(Q, \overline{Q})$ be representative of the conditions existing in a real design. In our measurements we use 14 minimal size inverters (in the same technology) as a representative load. Finally the remaining question is: should we load only the output producing the longer delay or both? We performed our measurements by loading only

the worse of the two $(Q, \overline{Q})$. This is justified by the fact that the critical path can always be improved by duplicating the CSE, and thus reducing the load to zero on the output that is not in the critical path. This is the approach that is taken by a reasonable designer and a synthesis tool as well.

4.3 Figure of Merit

It is well known that power can always be traded for speed and that superior speed can always be obtained by allowing for higher power consumption. Thus, it is hard to tell which one of the two CSE compared against each other is better. Various figures of merit have been used in the past. One commonly used and grossly misleading factor is Power-Delay-Product (PDP). It is not difficult to prove that PDP would always favor slower design, given that the energy consumed depends on the clock speed as well. It has been shown that more appropriate figure of merit is Energy-Delay-Product (EDP), [16]. However, some recent results argue that ED^2P is even more appropriate, at least in high-performance systems [19]. In our measurements we use PDP at a fixed frequency, which represents EDP.

5 Design for Low Power

The energy consumed in a clocked storage element is approximated by:

$$E_{switching} = \sum_{i=1}^{N} \alpha_{0-1}(i) \cdot C_i \cdot V_{swing}(i) \cdot V_{DD} \qquad (13)$$

where N is the number of nodes in a clocked storage element, C_i is the node capacitance, $\alpha_{0-1}(i)$ is the probability that transition occurs at node i, and V_{swing} is the voltage swing of node i. Starting from (7), several commonly used techniques applied to minimize energy consumption can be derived:

Reducing the number of active nodes and assuring that when they are switching the capacitance is minimized.
Reducing the voltage swing of the switching node
Reducing the voltage (technology scaling)
Reducing the activity of the node

The approaches listed in (a)-(d) result in several known techniques used in low-power applications. One of the most common is “clock gating” which assures that the storage elements in an inactive part of the processor are not switching. A thorough review of the common techniques for low-power can be found in [13]. In this paper we describe some recent techniques applicable to low-power design of clocked storage elements.

5.1 Conditional Capture Flip-Flop

Motivation behind Conditional Capture technique is the observation that considerable portion of power is consumed for driving internal nodes even when the value of the output is not changed (low input activity). It is possible to disable internal transitions when it is detected that they will have no effect on output. Conditional capture technique attempts to minimize unnecessary switching of the CSE. By disabling redundant internal transitions, this technique achieves power reduction at little or no delay penalties. Due to this property, it is particularly attractive from the point of view of high-performance VLSI implementations. One such structure is CCFF [14], which operates on the principle of J-K Flip-Flop: data can affect the Flip-Flop only if it will result in the change of the output. An improved version of CCFF is presented in [15] which reduces the overall Energy-Delay Product by up to 14% in for 50% data activity, while total power saving is more than 50% with quiet inputs (Fig. 19.). CSE equipped with conditional features have advantageous properties in low data activity conditions. In the implementation shown in Fig. 19, conditional capture is achieved by direct sampling of (inverted) input during the transparency window in single-ended CCFF. However, this approach is associated with severe drawbacks, most important of which is related to increased set-up time for sampling logic *"0"*.

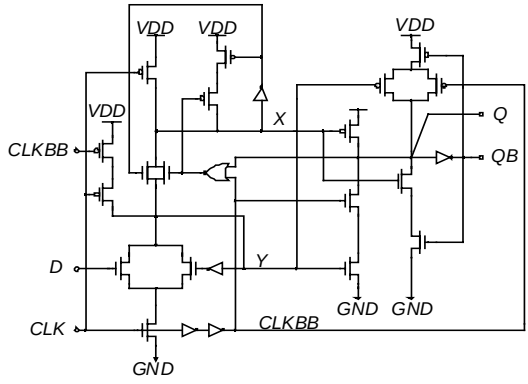

Fig. 18. Conditional Capture Flip-Flop [15]

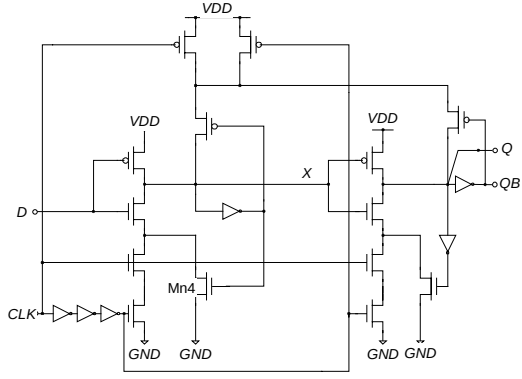

Fig. 19. Conditional Precharge Flip-Flop [15]

5.2 Conditional Precharge Flip-Flop

Conditional Precharge Flip-Flop (CPFF) [15] is shown in Fig. 19. Conditional Precharge technique is a way to save the unnecessary portion of the power in the Flip-

Flop. It eliminates power consuming precharge operation in dynamic Flip-Flops when it is not required.
Instead of gating data (in the evaluation phase), it is the precharge of the internal node that is conditioned by the state of the output. With the assumption that the internal node X is precharged (to logic "1") when the clock is in the "0" state, the evaluation of the node X happens during the Flip-Flop "transparency window". If the input D is "1", X is discharged to "0", which is used to set the output Q to "1". Node X remains at logic "0" as long as both input D and output Q are at the logic "1" level. This allows savings in the power consumed on unnecessary consecutive evaluations and precharges for D=1. Logic "1"-to-"0" transition of the output is achieved by sampling high level on X in the transparency window. Also, conditional keeping function is applied at the output to avoid contention with the output keeper - the output is kept at logic "0" as long as X is "1" and, similarly, it is kept high outside of the transparency window.
Like CCFF, this flip-flop has the problem of effectively higher set-up time for 1-to-0 transition due to the requirement to discharge the output before the transparency window is closed.

5.3 Dual Edge Triggering

One of the approaches amenable to high-performance as well as low-power application is the use of Dual-Edge Triggered (DET) clocked storage elements. Substantial power savings in the clock distribution network can be achieved by reducing the clock frequency by one half. This can be done if every clock transition is used as a time reference point, instead of using only one (*leading edge* or *trailing edge*) transition of the clock. Main advantage of this approach is that the system operates at half of the frequency of the conventional single-edge clocking design

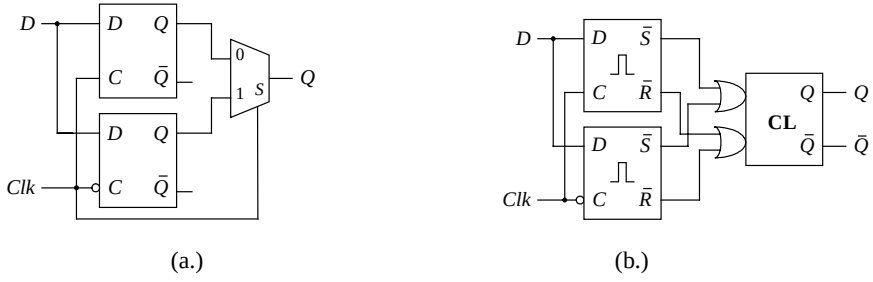

Fig. 20. (a.) Dual-Edge Triggered Latch-Mux (b.) Flip-Flop topology

style, while obtaining the same data throughput. Consequently, power consumption of the clock generation and distribution system is roughly halved for the same clock load. In addition, less aggressive clock subsystems can be built, which further reduces power consumption and clock uncertainties.

Dual-edge clocking requires Dual Edge-Triggered Storage Elements (DETSE), capable of capturing data on both rising and falling edge of the clock. The most critical obstacle for extensive use of dual-edge clocking strategy is the difficulty to precisely control the arrival of both clock edges. This control is essential in order to avoid large timing penalty incurred by the clock uncertainties. Even though this requirement imposes additional complexity, it can be satisfied with reasonably low hardware overhead. In addition, the clock uncertainty due to the variation of the duty cycle can be partially absorbed by the storage element [20].

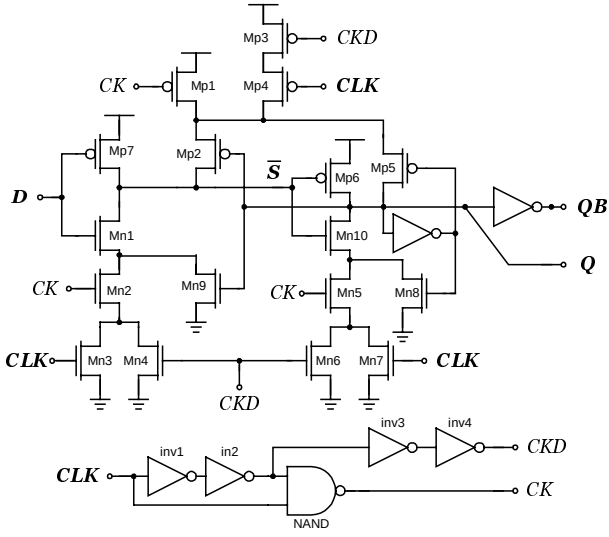

Fig. 21. Dual-Edge Conditional Pre-charge Flip-Flop, DE-CPFF [22]

There are two fundamental ways of building dual-edge clocked storage elements: Latch-Mux and Flip-Flop as shown in Fig.21.

An example of a dual edge-triggered Flip-Flop (Dual-edge Conditional Pre-charge Flip-Flop, DE-CPFF) is shown in Fig.21 [22]. Its operation is based on creating two narrow *transparency windows* during which the logic level of the input *D* can be transferred to the output. This Flip-Flop is a dual-edge version of Conditional Pre-charge Flip-Flop (CPFF, [21]).

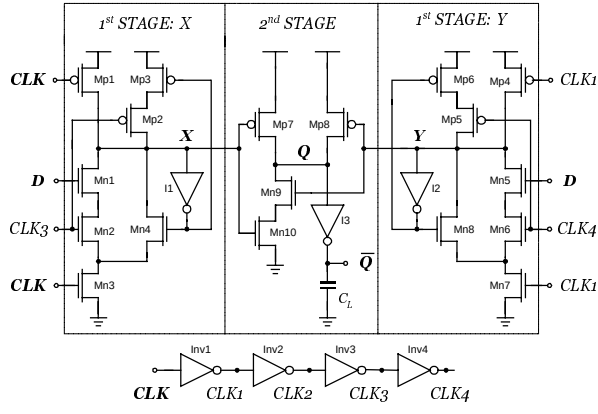

Fig. 22. Dual-Edge Triggered Flip-Flop [23]

Dual Edge Triggered Flip-Flop

An example of DET Flip-Flop design [22] is shown in Fig.22 The circuit has a narrow data transparency window and clock-less output multiplexing scheme. The first stage

is symmetric consisting of two Pulse Generating (PG) Latches. It creates the data-conditioned clock pulse on each edge of the clock. The clock pulse is created at node S_X on the leading and node S_Y on the trailing edge of the clock. The second stage is a 2-input NAND gate. It effectively serves as a multiplexer, implicitly relying on the fact that nodes S_X and S_Y alternate in being pre-charged "high", while the clock is "low" and "high", respectively. This type of output multiplexing is very convenient because it does not require clock control. The clock energy is mainly dissipated for pulse generation in the first stage. The clock load of the proposed flip-flop is similar to the clock load of SETSEs used in the high-performance processor designs, allowing power savings of about 50%. This makes this DETSE a viable option to be used in both high-performance and low-power systems.

6 Conclusion

A review of some (but not all) of the techniques for high performance and low-power CSE design is presented. For complete analysis of representative CSE please see [27] or visit: www.ece.ucdavis.edu/acsel where extensive database of comparative results exist. In the future we expect that pipeline boundaries will start to blur and synchronous design will be possible only in limited domains on the chip.

Acknowledgement. I gratefully acknowledge contributions from my current and former students: Vladimir Stojanovic, Dejan Markovic, Nikola Nedovic, Bora Nikolic and Bart Zeydel.

References

1. Gronowski P.E, et al, “High-performance microprocessor design” Solid-State Circuits, IEEE Journal of , Volume: 33 Issue: 5 , May 1998.
2. Bailey D.W, Benschneider B.J. “Clocking design and analysis for a 600-MHz Alpha microprocessor”, Solid-State Circuits, IEEE Journal of , Vol.33, No.11 , November 1998.
3. Schutz J, Wallace R. “A 450MHz IA32 P6 Family Microprocessor,” ISSCC Dig. Tech. Papers, pp. 236–237, Feb. 1998.
4. LSSD Rules and Applications, Manual 3531, Release 59.0, IBM Corporation, March 29, 1985.
5. Madden, W.C, Bowhill, W.J, "High input impedance, strobed sense-amplifier," United States Patent 4,910,713, March 1990.
6. V. Stojanovic, V. G. Oklobdzija, FLIP-FLOP, US Patent No. 6,232,810, May 15, 2001.
7. Partovi, H. et al, “Flow-through latch and edge-triggered flip-flop hybrid elements”, 1996 IEEE International Solid-State Circuits Conference. Digest of Technical Papers, ISSCC, San Francisco, February 8–10.
8. Klass F, “Semi-Dynamic and Dynamic Flip-Flops with Embedded Logic,” Symposium on VLSI Circuits, Digest of Technical Papers, pp. 108–109, June 1998.
9. Dobberpuhl, D.W. **“**Circuits and technology for Digital’s StrongARM and ALPHA microprocessors”, Proceedings of the Seventeenth Conference on Advanced Research in VLSI, Ann Arbor, Michigan, September 15–16. p. 2–11.

10. Nikolic, B, Oklobdzija, V.G, "Design and optimization of sense-amplifier-based flip-flops", Proceedings of the 25th European Solid-State Circuits Conference, ESSCIRC'99., Duisburg, Germany, 21-23 September 21–23. p.410–13.
11. Nedovic N, Oklobdzija V.G, Dynamic flip-flop with improved power. Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors. ICCD 2000, Austin, Texas, September 17-20. p.323-6.
12. V.Stojanovic and V.G. Oklobdzija, "Comaparative Analysis of Master-Slave Latches and Flip-Flops for High-Performance and Low-Power VLSI Systems," IEEE Journal of Solid-State Circuits, Vol.34, No.4, April 1999.
13. T. Kuroda and T. Sakurai *"Overview of Low-Power ULSI Circuit Techniques"*, IEICE Trans. Electronics, E78-C, No 4, April 1995, pp. 334–344, INVITED PAPER, Special Issue on Low-Voltage Low-Power Integrated Circuits.
14. B. S. Kong, S. S. Kim, Y. H. Jun, "Conditional Capture Flip-Flop Technique for Statistical Power Reduction", Digest of Technical Papers, p. 290–291, February 2000.
15. Nikola Nedovic, Marko Aleksic and Vojin G. Oklobdzija, "Conditional Techniques for Small Power Consumption Flip-Flops", Proceedings of the 8th IEEE International Conference on Electronics, Circuits and Systems, Malta, September 2–5, 2001.
16. M. Horowitz, et al, *"Low-Power Digital Design"*, Proceedings of the 1994 IEEE Symposium on Low-Power Electronics, 1994.
17. F. Klass et al, "A New Family of Semidynamic and Dynamic Flip-Flops with Embedded Logic for High-Performance Processors", IEEE Journal of Solid-State Circuits, vol. 34, no. 5, pp. 712–716, May 1999.
18. Friedman EG (ed.) "Clock Distribution Networks in VLSI Circuits and Systems", IEEE Press.
19. K. Nowka, P. Hofstee, private communication, IBM Research, Austin, Texas, 2000.
20. M. Saint-Laurent et al, "Optimal Sequencing Energy Allocation for CMOS Integrated Systems", Proceedings of International Symposium on Quality Electronic Design, p.94–99, March 2002.
21. N. Nedovic, V. G. Oklobdzija, "Hybrid Latch Flip-Flop with Improved Power Efficiency", Proceedings of the Symposium on Integrated Circuits and Systems Design, SBCCI2000, Manaus, Brazil, p. 211–215, September 18–22, 2000.
22. N. Nedovic, M. Aleksic, V. G. Oklobdzija, "Conditional Pre-Charge Techniques for Power-Efficient Dual-Edge Clocking", Proceedings of the International Symposium on Low Power Electronics and Design, Monterey, California, August 12–14, 2002.
23. N. Nedovic, W. W. Walker, V. G. Oklobdzija, M. Aleksic, "A Low Power Symmetrically Pulsed Dual Edge-Triggered Flip-Flop", Proceedings of the European Solid-State Circuits Conference, ESSCIRC'02., Florence, Italy, September 24–26, 2002.
24. Tschanz James, Siva Narendra, Zhanping Chen, Shekhar Borkar, Manoj Sachdev, Vivek De, "Comparative Delay and Energy of Single Edge-Triggered & Dual Edge-Triggered Pulsed Flip-Flops for High-Performance Microprocessors, Proceedings of the 2001 International Symposium on Low Power Electronics and Design, Huntington Beach, California, August 6–7, 2001.
25. Unger S.H, Tan CJ, (1986) *"Clocking Schemes for High-Speed Digital Systems"*, IEEE Transactions on Computers, Vol. C-35, No 10, October 1986.
26. Oklobdzija V.G, (ed.) "*High-Performance System Design: Circuits and Logic*", Book, IEEE Press, July 1999.
27. Oklobdzija V.G, Stojanovic V, Markovic D, Nedovic N, "*Digital System Clocking: High-Performance and Low-Power Aspects*, J. Wiley, in press expected December 2002.