Clocking in Multi-GHz Environment

VOJIN G. OKLOBDZIJA, FELLOW IEEE

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 should different clocked storage elements be compared against each other. The issues related to power consumption and low-power design are presented.

I. INTRODUCTION

Deciding on the clocking strategy in digital system is one of the single most important decisions. If taken lightly it always turns to be very costly afterwards.

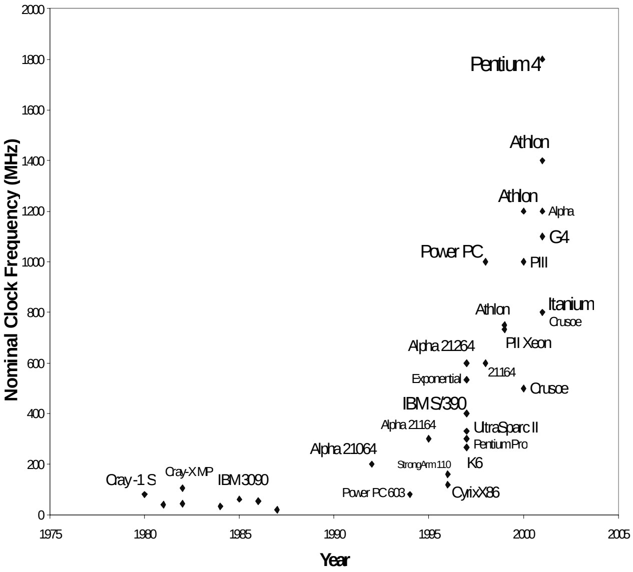


Fig. 1. Clock frequency versus year for various representative machines

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 or to make the Clocked Storage Element (CSE) faster 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.

With the increase in the level of integration, followed by a 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 is 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.

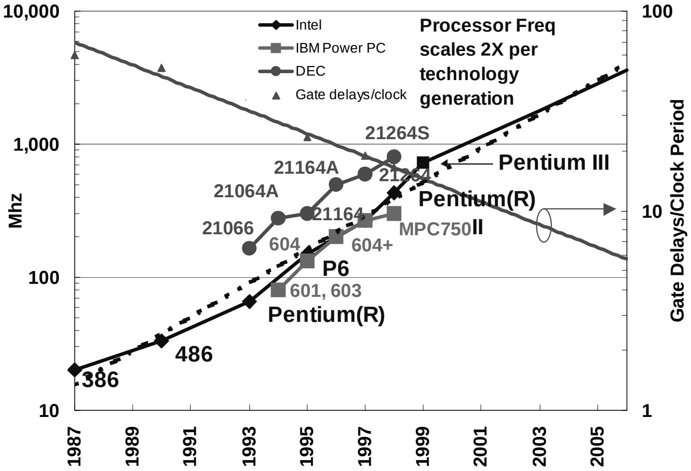


Fig. 2. Increase in the clock frequency and decrease in the number of logic levels in the pipeline (courtesy of Shekhar Borkar, Intel Corp.)

A. 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.

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

V. G. Oklobdzija is with Integration Corp. Berkeley, California, USA and Department of Electrical Engineering, University of California, Davis, E-mail: vojin@ece.ucdavis.edu

0-7803-7235-2/02/$10.00 © 2002 IEEE

is this phenomena that affects the time available to the logic.

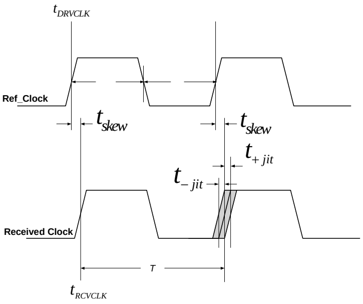


Fig. 3. Clock Parameters: Period, Width, Clock Skew and Clock Jitter

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). Therefore, the clock signal has the largest fan-out of any node in the design, which requires several levels of amplification (buffering). As a consequence, the clock system by itself can 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.

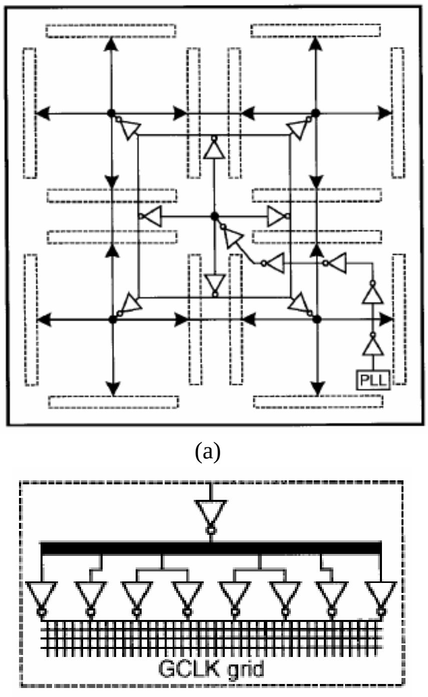
There are several methods for the on-chip clock signal distribution attempting to minimize the clock skew and contain the power dissipated by the clock system [18]. The clock can be distributed in several ways of which the two typical cases are: (a) an RC matched tree and (b) a grid shown in Fig. 4.

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. 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.

B. Controlling the Clock Signal Arrival Uncertainties

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. The digital deskewing circuit for clock distribution compensates out the *static components of skew* (load, interconnect, and device mismatches)



(a)

(b)

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

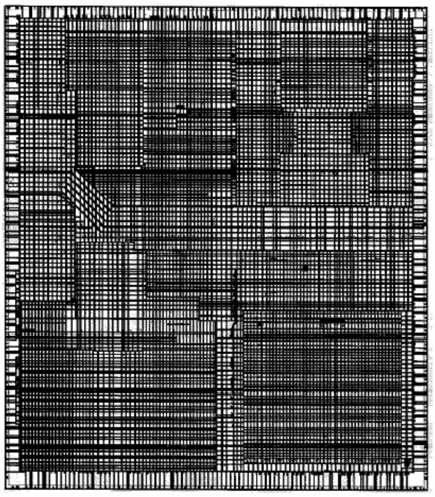


Fig. 5. Clock distribution grid used in DEC Alpha 600MHz processor [2], courtesy of IEEE.

It also compensates for the *dynamic* variations of temperature and voltage gradients between the two spines during all phases of microprocessor active operation.

Conceptual diagram of active digital deskewing [3] is shown in Fig. 6. The deskewing circuit equalizes insertion delay in the two clock distribution spines by compensating for delay mismatch in left and right spines of the microprocessor clock network. The circuit is composed of delay lines in both spines, a phase detection circuit, and a controller as illustrated in Fig. 6.

The skew for the clock distribution network in a 7.5M 0.25μm technology IA-32 P6 family microprocessor design [3] has >60ps of skew from left to right with the deskewing circuit inactive. With the deskewing circuit active the skew is reduced to 15ps.

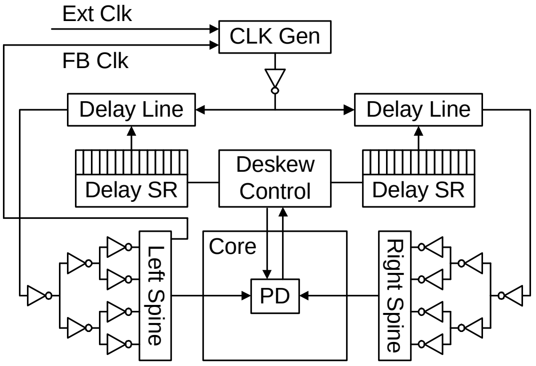


Fig. 6. Clock distribution network with deskewing circuit [3].

II. 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*.

A. Master-Slave Latch Arrangement

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 arrangement or L1 – L2 latch (in IBM) as shown in Fig. 7. This is not to be confused with the "*Flip-Flop*", though it seems that many practitioners today do erroneously call the arrangement shown in Fig. 7. (b), a Flip-Flop (F-F). We insist on the terminology that distinguishes Flip-Flop from M-S Latch and we will explain the fundamental differences between the F-F and M-S Latch in this paper.

In a Master-Slave arrangement 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-input for testing purposes, and clocked with a separate clock. One such arrangement, which utilizes two Masters, is a well-known IBM Level-Sensitive-Scan-Design [4].

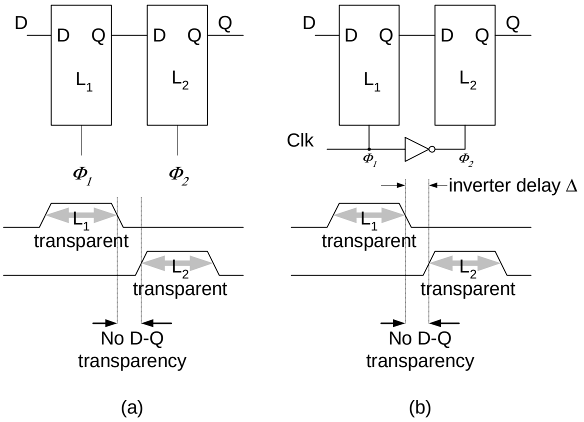


Fig. 7. Master-Slave Latch arrangement with: (a) non-overlapping clocks (b) single external clock.

B. 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 can have a small period of transparency associated with the narrow window during which the clock changes.

A general structure of the Flip-Flop is shown in Fig. 8. The difference between a Flip-Flop structure (Fig. 8) and that of the M-S Latch arrangement (Fig. 7) 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:

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

Those two equations (1) form a basis for derivation of a Flip-Flop structure.

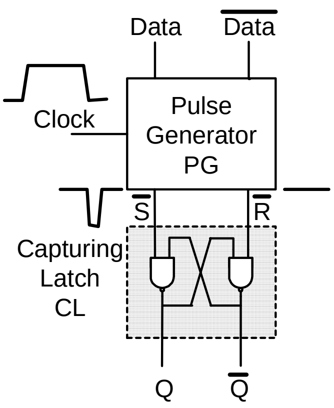


Fig. 8. General Flip-Flop structure

Simply stated, the equation for S_n tells us that: *The next state of this Flip-Flop will be set to "one" only at the time the clock becomes "one" (raising edge of the clock), the data at the input is "one", the flip flop is in the "steady state" (both S and R are "zero"). 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). Development of the Pulse Generator block of this Flip-Flop is illustrated in Fig. 9. A substantial improvement in speed is achieved by modification of the second stage by Stojanovic (US Patent No. 6,232,810) [6].

C. Time Window based Flip-Flops

Digital circuits are based on discrete 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 transition, which was demonstrated later.

A Flip-Flop based on the same described principle was introduced by Klass [8], Fig. 11.

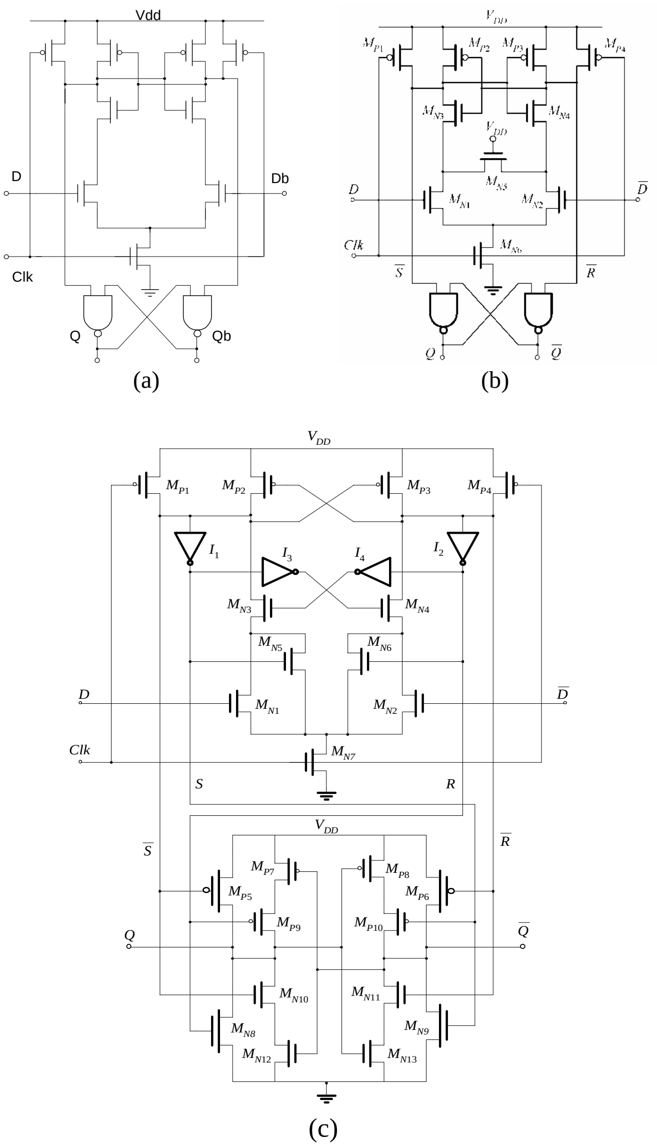


Fig. 9. Pulse Generator stage of the Sense Amplifier Flip-Flop: (a) Madden and Bowhill [5], (b) Improvement for floating nodes, Doberpuhl [9], (c) improvement by proper design: second stage (Stojanovic, US Patent: 6,232,810), first stage [10].

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 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:

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

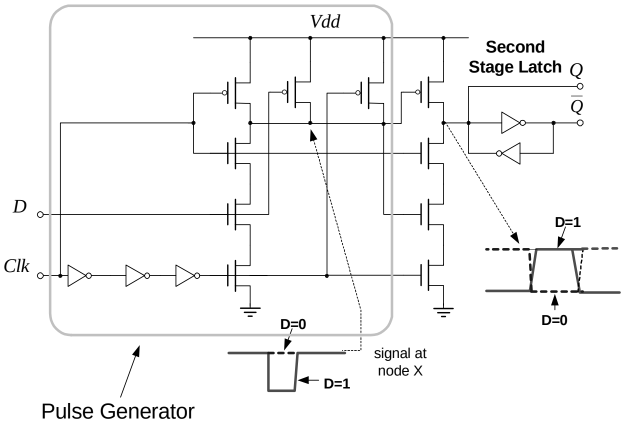


Fig. 10. Hybrid-Latch Flip-Flop introduced by Partovi [7]

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

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

The second stage(capturing latch) is implemented as:

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

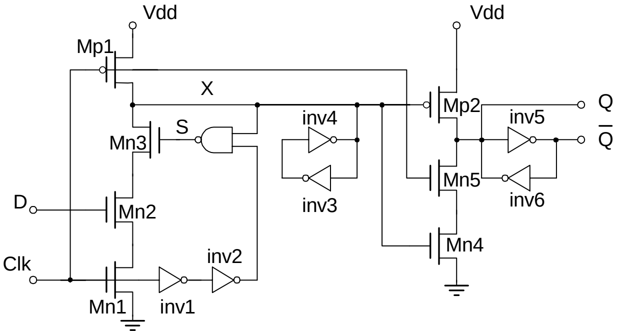


Fig. 11. Semi-Dynamic Flip-Flop: SDFF [8]

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

III. 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. 13. 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.

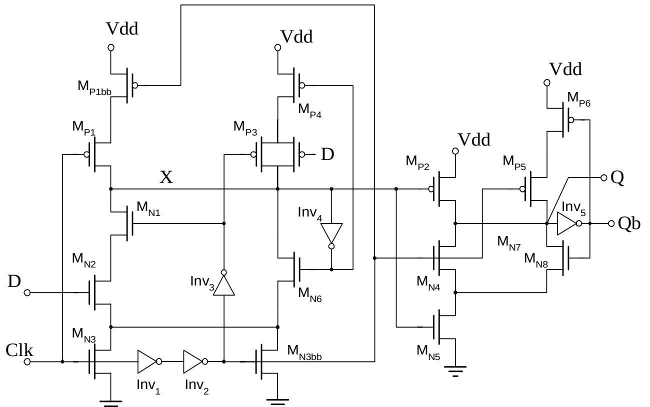


Fig. 12. Systematically derived single-ended Flip-Flop [11]

A. 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. 13. Considering how close should data be allowed to change with respect to the locking event, we encounter two opposing requirements:

(a) it should be kept further from the failing region for the purpose of design reliability.

(b) it should be as close to the clock in order to increase the time available for the logic operation.

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.

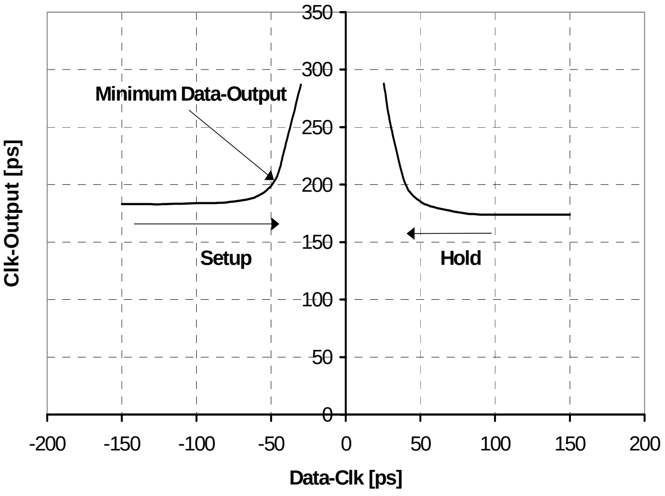


Fig. 13. Setup and Hold time behavior as a function of Clock-to-Output delay

A redrawn picture, Fig.14, where D-Q (t_{DQ}) delay is plotted (instead of Clk-Q), provides more information. From this graph we see that in spite of Clock-Q delay rising, we are still gaining because the time taken from the cycle is reduced.

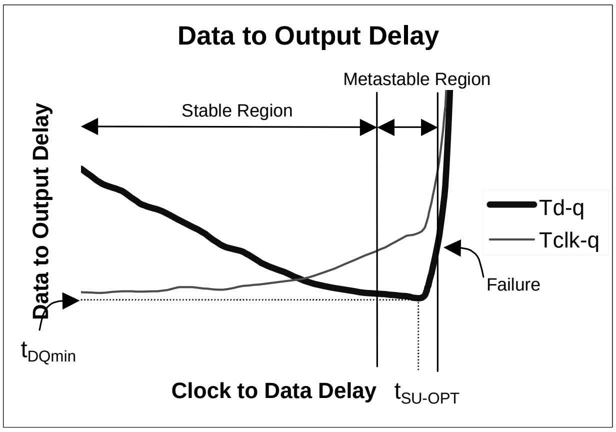


Fig. 14. Setup and Hold time behavior as a function of Data-to-Output delay

B. Time Borrowing and Absorption of Clock Uncertainties

The increase in delay from the storage element is still smaller than the amount of delay introduced to the cycle, thus allowing more time for the useful logic operation. 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. 15.

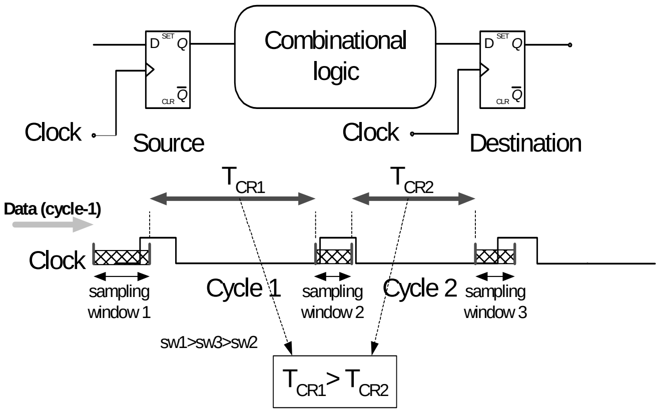


Fig. 15. "*Time Borrowing*" in a pipelined design

As it can be seen in Fig. 15, the *"sampling window"* moves around the time axes. 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 in absorbing the clock 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. 16.

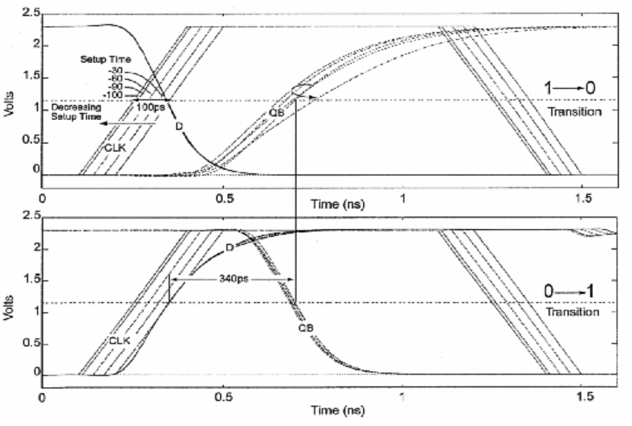


Fig. 16. Clock jitter absorbing properties of HLFF [7].

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:

$$t_{CLK-Q} \geq t_{hold} + t_{skew} \tag{5}$$

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.

IV. CHARACTERIZATION

A. 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. 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].

$$\mathrm{P_{tot}} = \mathrm{P_{internal}} + \sum_{\mathrm{inputs(D,CLK)}} \mathrm{P_{driver}} \tag{6}$$

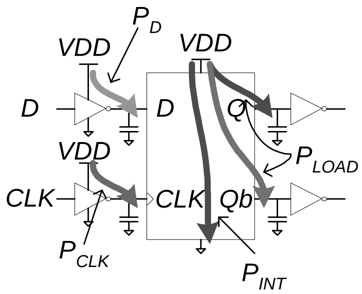


Fig. 17. Sources of power consumption in a CSE

B. 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. 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: $(Q,\overline{Q})$? We performed our measurements by loading only the worse of the two. This is justified by the fact that the critical path can always be improved by duplicating the CSE, and 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.

C. 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^{2}P is more appropriate [19]. In our measurements we use PDP at a fixed frequency, which represents EDP.

V. Design for Low Power

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

$$E_{switching} = \sum_{i=1}^{N} \alpha_{0-1}(i)\cdot C_i \cdot V_{swing}(i)\cdot V_{DD} \qquad (7)$$

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:

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

A. Conditional Capture Flip-Flop

Conditional capture technique attempts to minimize unnecessary switching of the CSE. 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

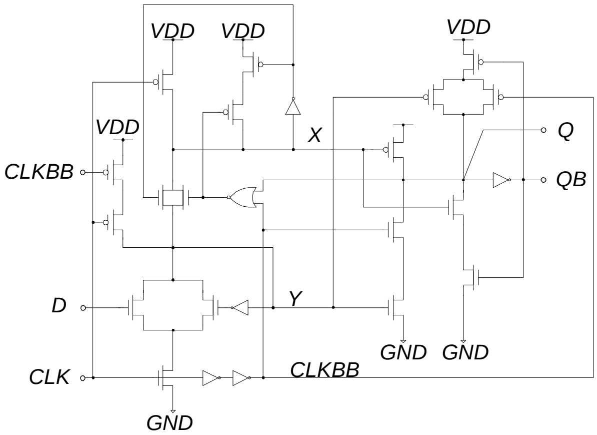


Fig. 18. Conditional Capture Flip-Flop [15]

total power saving is more than 50% with quiet inputs (Fig. 18.). CSE equipped with conditional features have advantageous properties in low data activity conditions. However, conditional techniques are suitable for applications in the high-performance circuits as well.

B. Conditional Precharge Flip-Flop

Conditional Precharge Flip-Flop (CPFF) [15] is shown in Fig. 19. It eliminates power consuming precharge operation in dynamic Flip-Flops when it is not required.

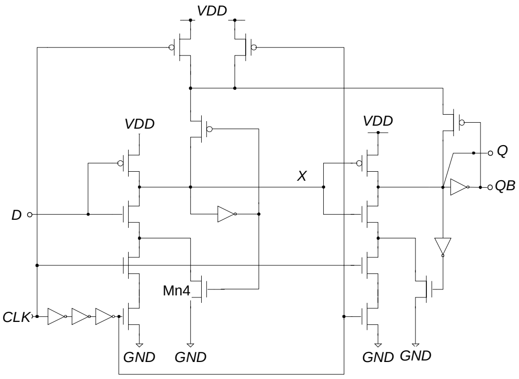


Fig. 19. Conditional Precharge Flip-Flop [15]

VI. 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 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 contribution from my students: Nikola Nedovic, Marko Aleksic, Bart Zeydel, Hoang Dao and Xiao-Yan Yu.

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, p290-291, 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] Oklobdzija V.G et al, “*Digital System Clocking: Clocked Storage Elements for High-Performance and Low-Power Design*, Book, in press, July 2002.

[21] Oklobdzija V.G, (ed.) “*High-Performance System Design: Circuits and Logic*”, Book, IEEE Press, July 1999.