IMTC 2004 - Instrumentation and Measurement
Technology Conference
Como, Italy, May 18-20, 2004

Timing Requirement for Reliable Latch-Based Circuit Design

Young Jun Lee[1], Yong-Bin Kim[1], F. Lombardi[1], Nohpill Park[2]

[1] ECE Department,
Northeastern University,
Boston MA 02115, USA
E-mail: {yjlee, ybk, lombardi}@ece.neu.edu

[2]Oklahoma State University,
Stillwater OK 74078, USA
Email: npark@cs.okstate.edu

Abstract – *This paper presents a framework of simulation and verification methodology for latch-based VLSI design. The proposed methodology includes optimal latch insertion point identification, how to consider clock skew for timing, and how to simulate circuits to verify the timing and functionality considering the clock skew in high speed VLSI systems for latch-based design. An existing flip-flop based FFT block is converted to latch-based design using the proposed methodology, and the performance of the block is increased by 10%.*

Keywords – *Timing requirement, Sync sim, Flow-thru sim, Minimum delay, Maximum delay*

I. INTRODUCTION

As performance target gets faster and interconnect delays increase as fabrication technology develops, the penalty that clock skew imposes on cycle time becomes more costly. The cycle time overhead of clock uncertainty can be eliminated in the latch-based design if the circuit delays between latches are balanced. Furthermore, latches provide more flexibility for tuning the stages of a pipeline or manipulating timing budget between the blocks of combinatorial logic. The techniques for the flexibility such as time borrowing, or cycle stealing are based on the fact that the extra time needed could be traded with the time allowed for the next cycle. However, latch-based designs may cause controversy in a few areas. Latch timing is ambiguous, and latch-based design makes timing analysis complicate. There have been publications focusing on timing analysis methodology for latch-based design[2][5][4]. However, it is very important to set up a well defined methodology for the design and simulation of the latch-based design to guarantee the latch-based timing of the high performance synchronous design and to use the cycle time effectively.

In this paper, we present a framework of the design and simulation methodology for latch-based VLSI system design that can be integrated in the existing design flow. The proposed methodology for design and simulation of the latch-based system include optimal latch insertion points in the logic paths, how to consider clock skew for timing, and how to simulate circuits to verify the timing and functionality considering the clock skew in high speed synchronous digital systems.

II. CONSIDERATIONS FOR LATCH-BASED SIMULATIONS WITH SKEW

Clock skew is a clock arrival time difference between any two points in a chip caused by process variation, power supply variation, clock loading difference, etc. The skew must be kept a minimum to ensure that setup and hold times are not violated at latches. And timing simulations must check the validity of a design with the estimated worst skew budget. To develop a timing simulation methodology for latch-based design, solid understanding about synchronous design is required.

A. *Simulation points and regions*

The term "sync-point" (synchronization point) refers to a point where the clock starts consequential data transitions. A latch is a sync-point where data is setup to the evaluating clock edge, including skew and setup requirements. Any data transitions before a sync-point cannot affect the logic path after the sync-point. For the latch B of Fig.1, there are three regions 1, 2, and 3 where data can arrive at the latch through the delay of the logic path between the latch A and the latch B. When the delay of the logic path is shorter than [HP (normal half clock period) - skew], the data arrives in region 1 and the latch B would be a new sync-point. When the delay of the logic path is longer than (HP + skew), the data arrives in region 3 and the latch is flow-thru. Note that the delay of the logic path must be shorter than [FP (normal full clock period) - skew]. Region 2 is a nondeterminant region, depending on the arrival of the clock signal within the skew band, the path could be clock or data limited. This may require moving logic or balancing wire delay between phases. Time-borrowing does not occur across a sync-point and always occurs across a flow-thru point. For example, in Fig.1, the data could arrive

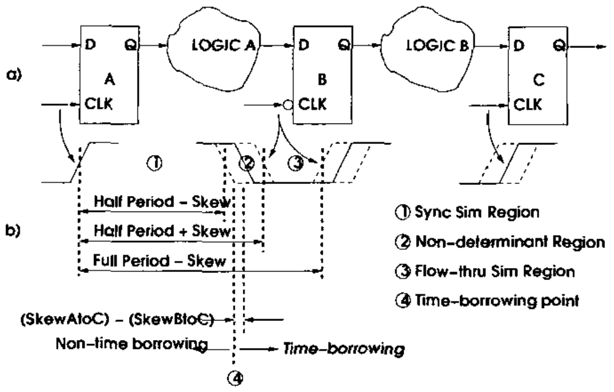

Fig. 1. Sync point and flow-thru point: a) schematic abstract, b) simulaton regions and time-borrowing point.

after the sync-point at (HP - skew_A_to_B) and before the time borrowing point at (HP - skew_A_to_C + skew_B_to_C), and the path would not require time borrowing. Therefore, the following rules are to be observed.

Rule 1: All simulations must initiate with a sync-point, finish with a sync-point, and include all flow-thru latches between sync points.

Rule2: Latches should be placed such that they fall either in region 1 (sync) or in region 3 (flow-thru).

B. Minimum delay and Maximum delay

The setup time of a circuit is the amount of time an input must arrive before the clock edge arrives. The time separation between the clock and data depends on the type of circuit as well as the direction of transition. For example in Fig.2, a voltage drop or glitch should not appear when the two input edges transition in opposite directions, i.e. clock rises before data falls. If the edges overlap too much, then both N devices will be ON and the output may transition low. To prevent such an erroneous transition, the data must transition a setup time before the clock edge. The cycle time must account for the skew caused by the early clock reaching latch B and the late clock reaching latch A as shown in Fig.3 (a), and (c). The cycle time must be greater than the maximum delay, the sum of the maximum propagation delays of latch A (1), the maximum logic delay (2), the minimum setup time of latch B (3), and the maximum skew between latch A and latch B (4).

Minimum delay simulations check the hold time constraint. The situation involves one latch that becomes transparent and a following latch that becomes opaque. If the data from the first latch moves too fast, it can pass through the following latch before it becomes opaque. The method of correcting this problem is to require a minimum delay between latches, which is called minimum delay simulation. These simulations must be run in process corner that gives the fastest delay. When designing a critical path, this requirement must be kept in mind, especially with respect to the last phase in a flow-thru path. Normally, this phase should be as short as possible so that the second to last latch is flow-thru while the last latch meets the sync requirement. Fig.3(b) includes latch B that feeds directly into latch A. The delay of the path from the clk to the input of latch B [(1) + (2)] must be greater than [the maximum skew (4) + the minimum hold time (5)] in Fig.3(d).

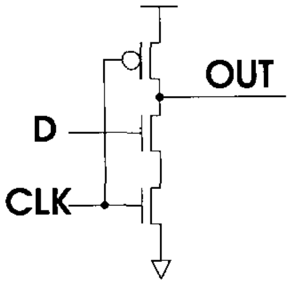

Fig. 2. A clocked dynamic gate

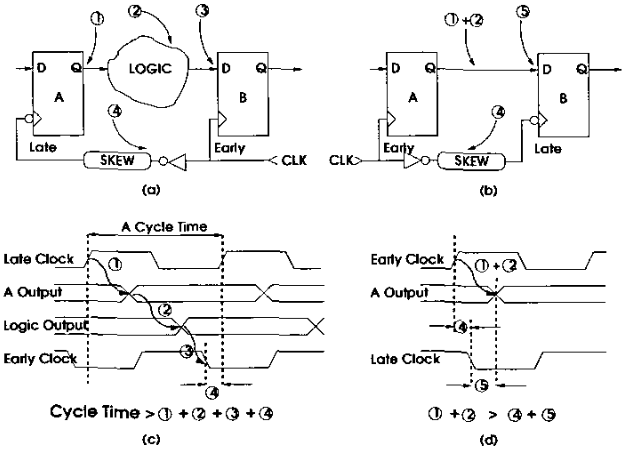

Fig. 3. Maximum and Minimum delays: a) maximum delay schematic abstract, b) minimum delay schematic abstract, c) maximum delay diagram, and d) minimum delay diagram.

C. Simulation environment

When simulating a critical path, special care should be taken to simulate its worst possible delay path for a gate as well as the critical logic path. If one or few gates are connected in a favorable way for the propagation of either rising or falling edges, the connection of the gates can produce an erroneous result when the path delay is simulated. Fig.4 describes the input setup that provides the fastest and slowest rise and fall times for NAND and NOR configurations.

And if the longest path is found without considering logical function, the long path is called a false critical path. The worst

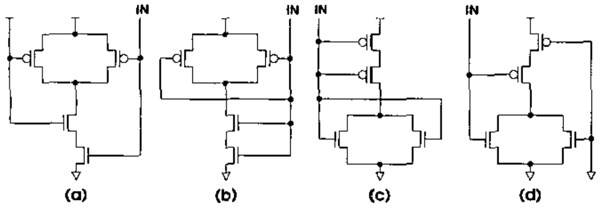

Fig. 4. Worst timing setup for NAND and NOR: a) slowest rise, b) slowest fall, c)fastest fall, and d) fastest rise.

path must not be false path. There have been proposed several methods for false path detection that use a static sensitization condition[6], a dynamic sensitization condition[?], or viability sensitization condition[3]. False path detection is important for checking if a circuit can operate at a given speed or not. Detection of false paths is a complex problem, and usually requires an extensive computational efforts.

The followings are simulation setup requirement for exact timing evaluations. First of all, always simulate a critical path when the gates are setup in the least favorable way for both rising and falling output. Secondly, input into the SPICE circuit must be sourced at least two nodes away from the node under test. Third, always let a circuit under test run one clock cycle before starting its real activity. This ensures that false measurement due to initial conditions will be avoided. Then, assign physical dimensions to all the parasitic elements in simulation schematics. Next, define the maximum slew rate for signals according to signal types such as static, dynamic or special signals. Note that the data and clock need to be measured at the actual clocking points, from the gate input of the pass gate to the data into the drain of the pass gate.

III. CIRCUIT SIMULATION METHODOLOGY CONSIDERING SKEW

A. Sync simulation and Flow-thru simulation

At every latch, there are only two styles of critical path simulations: sync simulation from the clock edge forward of current latch and flow-thru simulation from the previous logic path forward through the current latch. For latch B in Fig.5, the sync simulation examines the path from the clock edge of the latch B to the ending sync-point. The sync simulation means that the clock edge of current latch sets the subsequent data transition. In other words, during the sync simulation, the earliest clock edge has to appear after the latest data edge, i.e. the data should be stemmed from the clock edge. This leaves the sync simulation to dominate for the latches in the region 1, and leads to a sync simulation constraint of (HP - skew_B_to_C) for the delay of the logic path between latch B and C. The sync simulation passes if the delay of the logic path is shorter than (HP - skew_B_to_C). And the flow-thru simulation can be ignored in all sync latches.

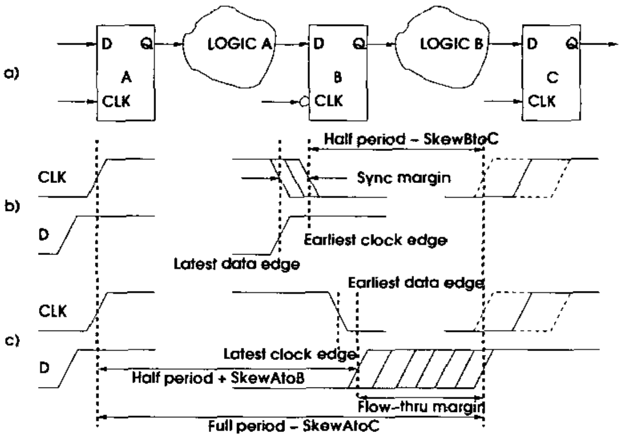

Fig. 5. Two simulations for Latch B: a) schematic abstract, b) sync simulation for clk after data, and c) flow-thru simulation for clk before data.

On the other hand, the flow-thru simulation covers the entire logic path from beginning sync point to ending sync-point. The flow-thru simulation denotes that the data from the previous logic path sets the subsequent data transition. During a flow-thru latch simulation, the earliest data edge should arrive after the latest possible clock edge, i.e. the clock edge at the latch B has to be derived from the data transition. This leaves the flow-thru simulation to dominate for the latches in the region 3, and draws to a flow-thru simulation requirement of (FP - skew_A_to_C) for the delay of the total logic path between latch A and C. The flow-thru simulation passes if the delay of the logic path is shorter than (FP - skew_A_to_C) and longer than (HP + skew_A_to_B). And the sync simulation can be ignored in all flow-thru laches.

B. Simulations with latches in the region 2

When data appears in the nondeterminant region as shown in Fig.6, both sync and flow-thru simulations must be run. Latches A and C are sync-points. In these simulations, the data must arrive before the skew budget of latch C. The difference between the data arrival and the nominal clock edge at latch B has been labeled Δt. This number may be positive or negative with respect to nominal clock. If the circuit under test meets both the sync simulation and the flow-thru simulation constraints, the circuit will work.

In Fig.6, data appears on the right side of the nondeterminant region (positive Δt). The sync simulation evaluates that the clock arrives after the data, and arranges the subsequent data transition. According to the sync simulation constraint, the delay of the logic path between latch B and C must be shorter than (HP - skew_B_to_C). The worst case occurs when the clock attains a negligibly short period after the data. In this case, the data edge should be stemmed from the clock edge using a perfect buffer.

On the other hand, the flow-thru simulation examines that the clock edge arrives before the data, leaving the data to set the subsequent data transition. According to the flow-thru simulation requirement, the delay from beginning sync to end sync must be shorter than (FP - skew_A_to_C). Since the data appears at latch B at (HP + Δt), (HP - skew_A_to_C - Δt) remains for the next phase. The worst case occurs when the clock arrives an infinitesimally short period before the data. Thus, the clock edge should be derived from the data edge using a perfect buffer.

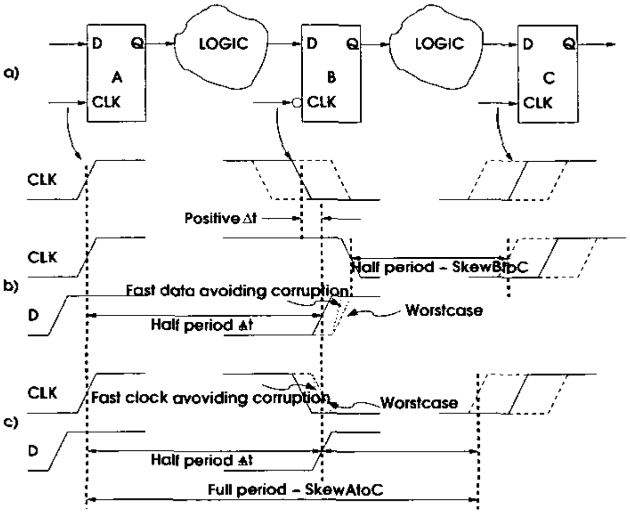

Fig. 6. Two simulations for the nondeterminant region: a) schematic abstract, b) sync simulation for clk after data, and c) flow-thru simulation for clk before data.

When the data arrives on the left side of the nondeterminant region (negative Δt), both simulations are the exact same as the above. One difference is that skew_A_to_C has a larger value than skew_B_to_C. As with the previous case, the delay of the logic path between latch B and C for the sync simulation must be shorter than (HP - skew_B_to_C). According to the flow-thru simulation, the delay from beginning sync to end sync must be shorter than (FP - skew_A_to_C). Since the data arrives at latch B at (HP - Δt), (HP - skew_A_to_C - Δt) is valid for the second phase.

Rule 3: Two simulations must be checked for every latch whose latest arriving data edge with skew arrives in the region 2 (Nondeterminant region).

C. Multiple nondeterminant region latches

Fig.7 shows the basic structure of simulations required for the case of multiple latches in nondeterminant region. The simulation in Fig.7(b) examines the case where the clock edges at both latch B and latch C arrive before the data. Fig.7(c) evaluates the case where the clock arrives after the data at latch B, and before the data at latch C. Fig.7(d) covers that the clock at latch C arrives after the data. If the arrival of data relative to clock is constrained at each latch, the number of simulations can be reduced. For example, if the data arrived at latch B in the flow-thru region, then the simulation (c) could be dropped leaving the flow-thru simulation (b) and the sync simulation from latch C (d). Another case would be if data were constrained to arrive at latch B in the sync region, then the general flow-thru simulation at the top (b) could be dropped.

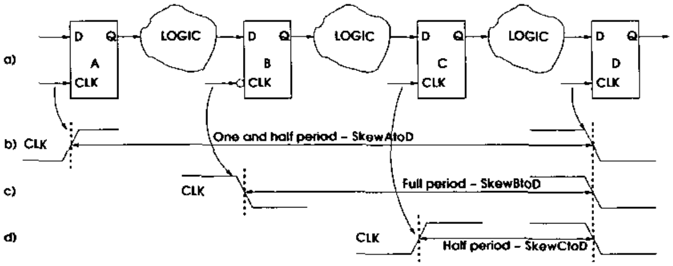

Fig. 7. Multiple Latches in Region 2: a) schematic abstract, b) flow-thru simulation at B and C, c) sync simulation at B and flow-thru simulation at C, and d) sync simulation at C.

D. Time borrowing in the region 2

Fig.8 (b) and (c) show the case where data arrive late in the nondeterminant region. If the data arrives after (HP + skew_B_to_C - skew_A_to_C), then the delay of the logic between latches B and C must introduce shorter than (HP - skew_B_to_C) in order to meet the flow-thru simulation constraint of (FP - skew_A_to_C). Note that this makes the flow-thru simulation constraint tighter than the sync simulation constraint which is only (HP - skew_B_to_C). Basically, enough logic has been shifted into the first phase so that the data arriving at latch B sets the critical path to the next sync-point. In the nondeterminant region, by using time borrowing, one can utilize all the available time without distorting the location of their latches.

Fig.8 (d) and (e) shows how to simulate time borrowing in the nondeterminant region of 500MHz clock frequency including the general constraints that need to be checked. The example specifies skews of 280ps between latches B and C, and of 400ps between latches A and C. The cycle path begins at a sync-point A, and ends at a sync-point C. The sync simulation checks the clock (B) to Q (C) path delay which must be less than (1000ps - 280ps). The flow-thru simulation evaluates the input (A) to Q (C) path delay through latch B which must be less than (2000ps - 400ps).

The example assumes that both clock to Q and D to Q paths have 0 delay, i.e. that data and clock are measured at the internal pass gate of latch B and the delays of the latch are in-

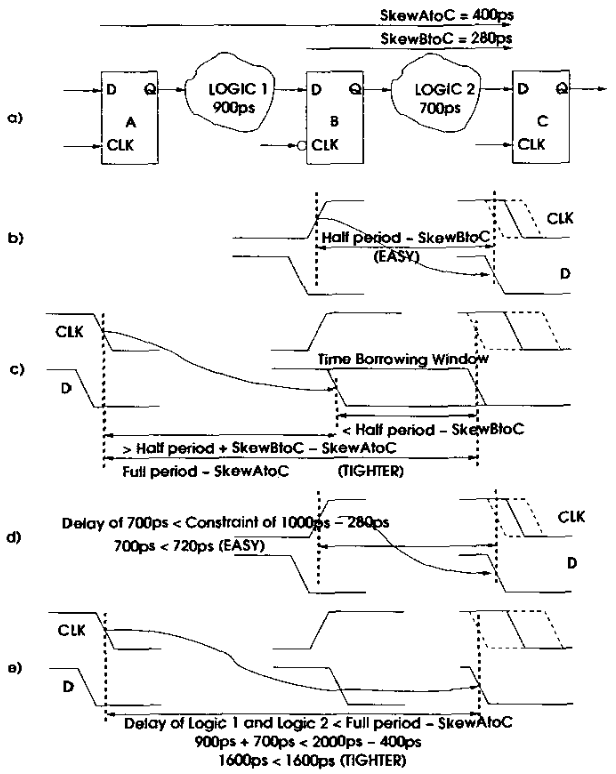

Fig. 8. Time Borrowing in the region 2: a) schematic abstract, b)-c) sync and flow-thru simulations, and d)-e) sync and flow-thru simulations for borrowing 5ps from 2nd phase

corporated into logic 1 and logic 2. As long as the data arrives sometime (900ps) after 880ps, i.e. logic 1 has delay greater than 880ps, then data will time borrow across latch B to set the subsequent sync-point. Basically, the flow-thru path is still setting the overall amount of logic even through the data is arriving before the nominal clock edge. The sync simulation shows there is 20ps margin, while flow-thru simulation is the tighter constraint. On the other hand, if the data arrives before 880ps, then data will wait at latch B and no borrowing will occur while the sync simulation is the tighter constraint.

E. Breaking up multi-phase simulations

A simulation between two sync-points with multi-phase over 4 must be broken into more manageable size, i.e. partial simulations. By having one simulation with the entire critical path, SPICE naturally carries a lot of information between all the parts of the simulation. When breaking one simulation into partial simulations, this information must be carried between simulations by hand. Basically designers must build a model of the load for the first partial simulation and a model of the driver for the second partial simulation. And the designers must carry forward the knowledge about the original source of the signal so that the appropriate amount of skew can be applied. A distinct set of information must be carried for each edge in each simulation.

Rule 4: The number of phases between sync-points should be less than 4 for analysis simplicity.

The most critical elements in the driver model are the driver type, the driver size, the arrival time and the wire model. For example, in a critical path with one nondeterminant region latch, the designer might need up to 4 different driver models: one for the falling edge of the flow-thru simulation, one for the rising edge of the flow-thru simulation, one for the falling edge of the sync simulation, and one for the rising edge of the sync simulation. In reality, some of these will be redundant because usually one path dominates the critical path simulation. The load model requires the correct amount of load including wire model. In order to check that the load and driver models reflect each other, the designer should compare the output waveform from the first partial simulation against the input waveform of the second partial simulation. The most important characteristics include absolute arrival time at 50% of V_{DD} and transition time between 20% and 80% of V_{DD}. Some cases may require special attention. For example, when feeding a dynamic circuit, the N-device threshold of the waveform is much more critical to timing than the 50% point, especially for longer rise times.

IV. RESULTS

To evaluate this methodology, we chose Fast Fourier Transform (FFT) block for a speech recognition chip. This was implemented using a 4 stage pipelined hardware that consists of adders, subtractors, multipliers, and registers. It takes time about 4500 cycles to calculate 256 point complex FFT. Replacing flip-flops with latches and retiming logic blocks between sync points eliminated the redundancy time of the original FFT module, and the performance of FFT block was increased by 10%.

V. CONCLUSIONS

This paper presents a framework of the design and simulation methodology for latch-based VLSI design that can be integrated in the existing design flow. This approach provides the design and simulation guidelines to help designers design and verify the functionality and timing of the block systematically.

REFERENCES

[1] D. Barnd and V. Iyengar. "Timing analysis using a functional relationship", In *IEEE/ACM ICCAD*, pages 126–129, 1986.
[2] A. Champernowne, L. Bushard, J. Rusterholz, and J. Schomburg. "Latch-to-latch timing rules", *IEEE Trans. on Computers*, 39(6):798–808, June 1990.
[3] H.-C. Chen and D. H.-C. Du. "Path sensitization in critical path problem", *IEEE Trans. on CAD/ICS*, 12(2):196–207, February 1993.

[4] D. Harris, M. Horowitz, and D. Liu. "Timing analysis including clock skew", *IEEE Trans. on CAD/ICS*, 18(11):1608–1618, November 1999.

[5] K. Sakallah, T. Mudge, and O. Olukotun. "Analysis and design of latch-controlled synchronous digital circuits", *IEEE Trans. on CAD*, 11(3):322–333, March 1992.

[6] S.Devadas, K. Keutzer, and S. Malik. "Computation of floating mode delay in combinational circuits: Theory and algorithms", *IEEE Trans. on CAD*, 12(12):1913–1923, December 1993.