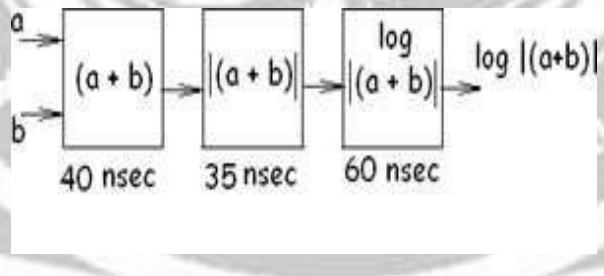
STATIC REGISTERS AND LATCHES

While studying sequential circuits, we studied about Latches and Flip Flops. While Latches formed the heart of a Flip Flop, we have explored the use of Flip Flops in applications like counters, shift registers, sequence detectors, sequence generators and design of Finite State machines. Another important application of latches and flip flops is in pipelining combinational/algebraic operation. To understand what is pipelining consider the following example. Let us take a simple calculation $C = \log(|a + b|)$ which has three operations to be performed viz. 1. add a and b to get (a+b), 2. get magnitude of (a+b) and 3. evaluate log |(a + b)|. Each operation would consume a finite period of time. Let us assume that each operation consumes 40 nsec., 35 nsec. and 60 nsec. respectively. The process can be represented pictorially



. Consider a situation when we need to carry this out for a set of 100 such pairs. In a normal course when we do it one by one it would take a total

of 100 * 135 = 13,500 nsec. We can however reduce this time by the realization that the whole process is a sequential process. Let the values to be evaluated be a_1 to a_{100} and the corresponding values to be added be b_1 to b_{100}.

Since the operations are sequential, we can first evaluate (a_1 + b_1) while the value

$|(a_1 + b_1)|$ is being evaluated the unit evaluating the sum is dormant and we can use it to evaluate $(a_2 + b_2)$ giving us both $|(a_1 + b_1)|$ and $(a_2 + b_2)$ at the end of another evaluation period.

Now at the end of the second evaluation period, in a similar fashion, we can initiate evaluation of log $|(a_1 + b_1)|$, $|(a_2 + b_2)|$ and $(a_3 + b_3)$ simulataneously and obtain the output log $|(a_1 + b_1)|$ at the end of the third evaluation period. To accommodate the operation with the longest delay of 60nsec., each evaluation period will be of 60nsec. This assumes that before initiating each operation all operations should have been over. After the arrival of the first output data log $|(a_1 + b_1)|$. The subsequent outputs log $|(a_2 + b_2)|$, log $|(a_3 + b_3)|$ will now start arrivaing at a gap of every 60 nsec. In other words, the inputs to the system can be changed every 60 nsec. i.e. all the 100 inputs can be applied in a time of 99*60 = 5940 nsec. And the total time taken tio evaluate 100 data will be 5940 + 180 = 6,120 nsec. -- less than half of 13,500 nsec needed earlier. This process of evaluation is called **registers**. There is however a catch here, we need to ensure that the input to each block is held constant till the calculations are over, an impossible task if we stick to the configuration in Fig. 1. A simple approach to ensure stability of input to each block is to insert a edge triggered flip flop register between each block with the registers clocked by a clock running

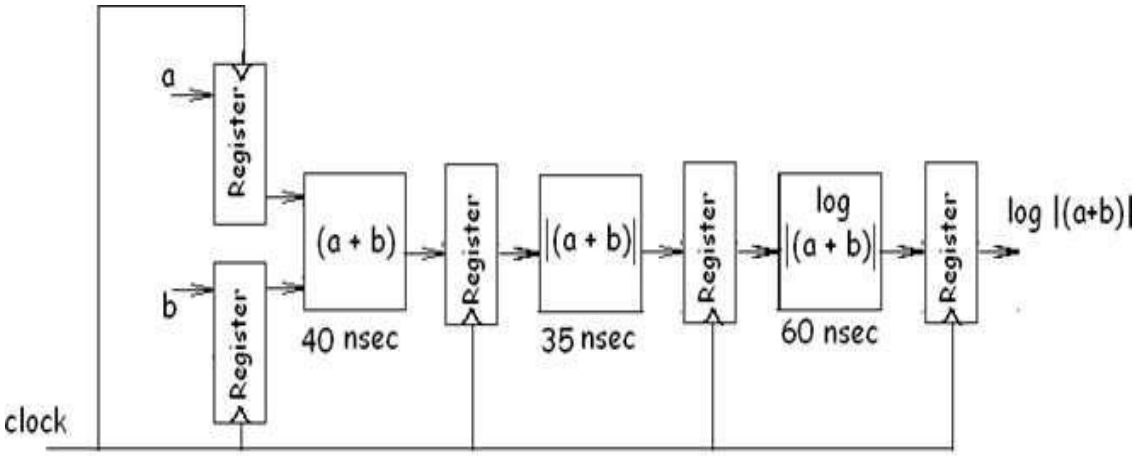


Fig 3.5.1 : Static registers and latches

[Source : Sung-Mo kang, Yusuf leblebici, Chulwoo Kim —CMOS Digital Integrated Circuits:Analysis & Design]

In the pipeline, we can use either an edge triggered D flip flop, a pulse triggered D flip flop or level triggered D latches. While using level triggered latches, the latches are clocked with clock and $\overline{clock}$ alternately. Let us now look in to pipeline system realized with three kinds of registers and the associated timing issues.

Timing Analysis of Pipeline Systems using Positive Edge TriggeredFlip- Flops

The positive edge triggered flip-flops propagate the input value of the data to output node on the positive edge of the clock. A D flip-flop, as you may recall, has three terminals: Data input D_{in}, clock input CLK and data output Q_{out}. When the clock input makes a low-high transition input D_{in} is propagated to output Q_{out}. Let us briefly recollect some of the timings associated with flip flops and their importance.

1. Cycle Time (T_{cycle})

It is the amount of time elapsed between two successive positive edges of the clock. We denote this parameter by T_{cycle}.

2. Clock-to-Out delay time ($t_{clock\text{-}out}$)

It is the delay from the triggering edge of the clock to the new stable output on Q_{out} in both edge triggered and pulse triggered flip flops, assuming that D_{in} has been setup sufficiently early to triggering edge of the clock. This parameter is denoted as $t_{clock\text{-}out}$. In a pulse triggered flip flop, the triggering edge is defined as the trailing edge of the pulse in a positive pulse triggered flip flop and rising edge of the pulse n a negative pulse triggered flip flop.

3. Data-stable-toout delay ($t_{data\text{-}out}$)

It is the delay between the instant when the data input D_{in} stabilizes to the instant when the output Q_{out} stabilizes. As opposed to edge triggered flip flops wherein $t_{clock\text{-}out}$ is of interest to us, for level triggered flip flop $t_{data\text{-}out}$ is of interest. In the case of edge triggered flip flops, the output Q_{out} changes at the clock edge when the slave latch transmits the master output to the flip flop output while in the case of level and pulse triggered latches, at the active clock level input D_{in} is transferred to the output Q_{out}. In most cases, for a given technology, $t_{clock\text{-}out}$ is marginally larger than $t_{data\text{-}out}$.

4. Setup Time (t_{setup})

It is defined as the minimum time between a Data (D_{in}) change and the triggering edge of the clock, also referred to as data-to-clock delay, such that the output Q_{out} will be guaranteed to change so as to become equal to the new value of D_{in}. We denote this parameter as t_{setup}. The setup time is defined such that, any change in Data at D_{in} arriving earlier than the Setup Time, t_{setup}, should not result in any changes in the clock-to-out delay time. In other words, data signal on D_{in} should occur sufficiently early so that making it appear any earlier would have no effect on when the output Q_{out} changes.

The variation of $t_{clock\text{-}out}$ delay with respect to setup time has been well studied. The studies show that, as the setup time is reduced, the $t_{clock\text{-}out}$ increases. The setup time cannot be reduced beyond a certain point after which new data is not latched into the flip-flop. They also defined the stable, transition and failure regions. The stable region is defined as the region of Data-to-Clock axis in which clock-to-out delay does not depend on the setup time. As setup time is reduced, clock-to-out delay starts to rise monotonically and ends in a failure. This region of data-to-clock axis is the transition region. The transition region is defined as the region of unstable Clock-to-out delay. Any change in data appearing in the failure region will not be latched into the flip-flop.

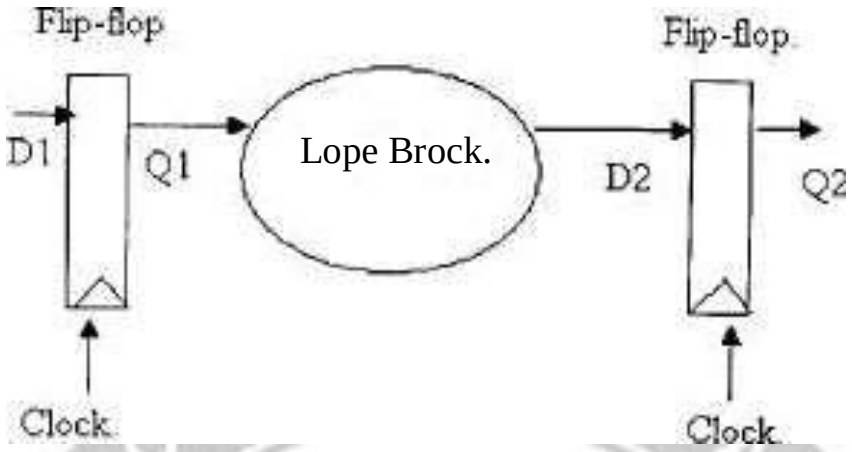
5. Hold time (t_{hold})

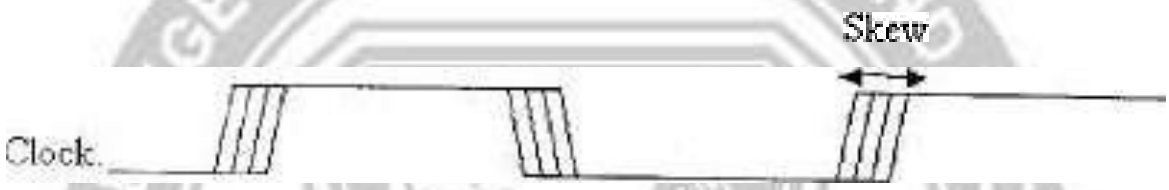
Hold time is the minimum time that the data at D_{in} must be held constant after the triggering edge of the clock signal, assuming that the most recent data change occurred not later than the Setup

time t_{setup}, such the output Q_{out} will remain stable. This is denoted as t_{hold} and this value can be negative.

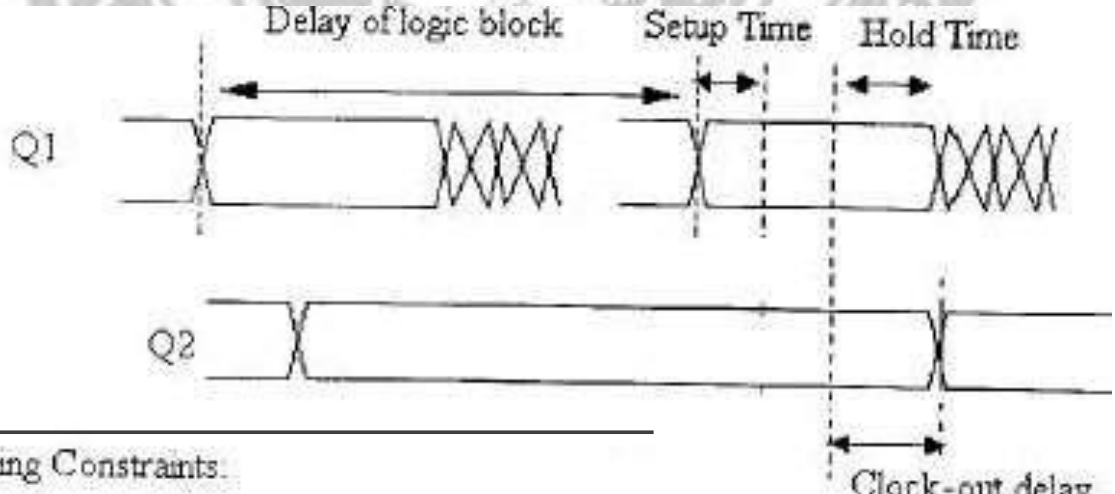
6. Clock Skew (t_{skew})

Clock skew is the difference in arrival time of the active clock edge between two registers. One of the primary concerns in the design of a distribution network in a digital system is the difference in the arrival times of clock edge between two flip-flops. The goal of clock distribution is to minimize, manage or eliminate this term altogether. Clock skew is of importance between sequentially adjacent registers. The presence of clock skew can significantly limit the performance of a synchronous system, and even create race condition that causes incorrect data to be latched into flip-flop. Some of the main sources of clock skew can be attributed to differences in line lengths from clock source to the destination register;; differences in delays of any active buffers in the network, differences in interconnect parameters in the network. It has been predicted that it is hard to maintain the total clock skew less than about 150 pS for high performance microprocessors. But many active clock distribution schemes have been implemented to obtain clock skews as low as 15 to 20 pS. The values reported in literature sometimes differ in their definitions of skews and how they have been measured or estimated. Usually, local clock skew is defined as the clock-skew between any two points on the same clock-grid while global clock-skew is defined as clock-skew between any two points on the entire clock network. This would mean





Tcyrle



Timing Constraints:

Max Delay Constraint:

$$T_{cycle} > t_{ck\text{-}out} + t_{logic} + t_{setup} + t_{skew}$$

Min Delay Constraint:

$$t_{logic} > t_{hold} - t_{ck\text{-}out} - t_{skew}$$

Fig 3.5.2: pipelining

[Source : Sung-Mo kang, Yusuf leblebici, Chulwoo Kim ―CMOS Digital Integrated Circuits:Analysis & Design]

that local clock skew can be significantly lowered while global clock would be hard to contain and probably would be as high as 150 pS even for best clock network designs.

Figure 3 illustrates the timing for a system built using edge-triggered flip- flops. The timing constraints for such a system are also shown in the figure. The Maximum delay constraint (indicated as Max. constraint) indicates that the clock cycle time should be greater than the delay of the logic block and the timing overheads introduced by flip-flops. This is called the Max. constraint as this constraint defines the maximum logic delay (t_{logic}) acceptable for a given clock, given flip flop timing parameters and clock skew. We reproduce the Max. Constraint here for convenience:

(1) $$T_{cycle} > t_{clock-out} + t_{logic}(\max) + t_{setup} + t_{skew}$$

In eqn. 1, the term t_{logic}(max) refers to the maximum delay in the logic block. Let us consider the example of evaluation of (shown in Fig. 2) we have seen that it involves three operations of delay 35nsec., 40nsec. and 60nsec. respectively. Let the $t_{clock-out}$, t_{setup}, t_{skew} and t_{hold} for the registers used be respectively 15nsec., 10nsec., 10nsec. and 5nsec. With the maximum logic delay of the three logic operations is 60nsec. the T_{cycle} has to be greater than 95nsec. (15 + 60 + 10 + 10). The output A natural question that arises in a reader's mind is the non inclusion of t_{hold} in the maximum delay constraint. This becomes obvious when we

realize that t_{hold} is subsumed in the combination of $t_{clock\text{-}out}$ and t_{logic} as the hold process is concurrent with clock-out and combinational delay. The minimum delay constraint poses a limit on the minimum delay a logic block must have in order to avoid a data race between flip-flops. Data to the flip-flop must be setup before the earliest clock might arrive;; yet we cannot guarantee that data will be valid until the clock-out delay after the latest clock. As we can see the amount of cycle time available for logic is reduced by setup time, skew and clock-out delay of the flip-flop. The Min delay constraint is expressed as

(2) $$t_{logic}\,(\min) > +t_{hold} - t_{clock-out} - t_{skew}$$

The Minimum logic delay that must be inserted to avoid a min delay failure is the sum of hold-time and skew minus the minimum clock-out delay. Also in these systems the logic delay is exactly known at the time cycles are partitioned, so some cycle have more logic and some have less logic. The clock cycle time should be long enough for the longest cycle to work correctly, which would imply that excess time in shorter cycles is wasted. Furthermore, for a pipeline which is of N stages,there is a total latency (a period when nothing is available at the output) of N clock cycles. In the case of the example considered, it will be 3 cycles and will be a minimum of 285 nsec.

In summary, the total overhead of the flip-flop based systems is distributed in clock-out delay, setup time, clock skew and overhead due to imbalancelogic.

2 Timing Analysis of Systems using level-triggered latched

Using level-triggered transparent latches can eliminate the clock-skew penalties of flip-flop systems. Such a system would operate with

two-phase transparent latches. Transparent latches have same terminals as flip-flops;; data input D, clock input and data output Q. The latch is transparent when the clock input is high and the input is copied to the output. When the clock goes low, the previous value of data is retained. Transparent latches are characterized by clock- out delay, data-out delay ($t_{data\text{-}out}$) and hold time. The data-out delay is defined as the time taken for the Q_{out} to be stable after D_{in} gets to be stable. Some refer to data-out delay as set up time for latches. Again we assume t_{skew} of uncertainty in the arrival of clock input. Figure 4 illustrates the timing analysis of a two-phase system built using transparent latches. One latch is controlled by clock input while the other input is controlled by its complement. The latching operation can be initiated anywhere within the phase and beyond the skewed clock and hence eliminating clock skew from the overhead. We reproduce the Max. Constraint and Min. constraints here for convenience:

(3) $$T_{cycle} > 2 * t_{data-out} + t_{logic}(\max)$$

(4) $$t_{logic}(\min) > t_{hold} - 2 * t_{data-out} - t_{skew}$$

Level sensitive latches avoid the problem of imbalanced logic using time borrowing (also known as cycle stealing). As we can see from the figure, each latch can be placed anywhere in the wide ranges of locations in its half-cycle and still be transparent when the data arrives. This means that not all half-cycles need to have the same amount of logic. Some can have more and some can have less, such that data arrives at the latch later or earlier as long as latch is transparent. Hence, longer cycles may borrow time from shorter logic cycles if the pipelines are not perfectly balanced (not all logic delays are identical) so that latency is not determined by the total logic delay.

Let us consider the example of evaluation of $\log(|a+b|)$, redrawn in Fig. 5 by using level sensitive latches for registers. We have seen that it involces three

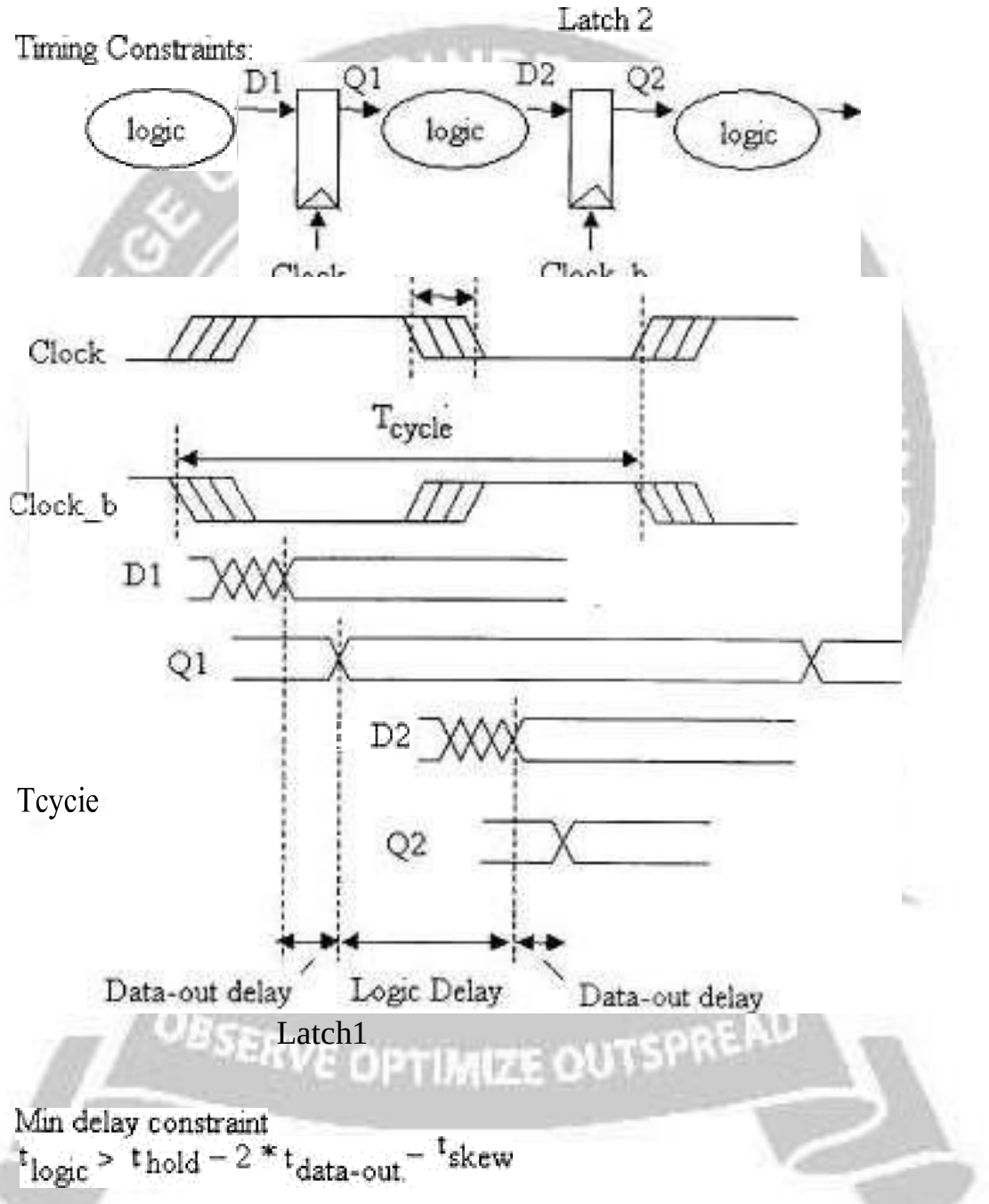


Fig 3.5.3 : Timing Analysis of Systems using level-triggered latched

[Source : Sung-Mo kang, Yusuf leblebici, Chulwoo Kim —CMOS Digital Integrated Circuits:Analysis & Design]

operations with delays 35nsec., 40nsec. and 60nsec. respectively. Consider that each latch has $t_{clock\text{-}out}$ and t_{hold} respectively 15nsec., and 5nsec. and the clock

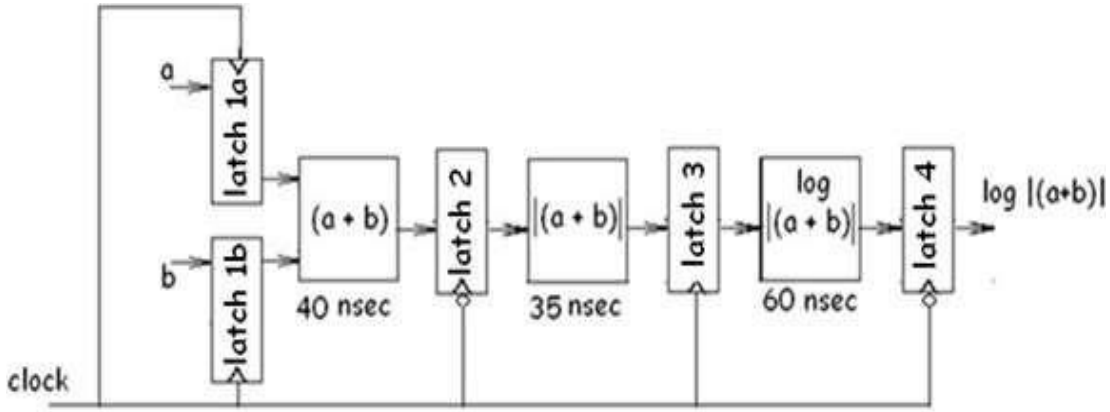


Fig 3.5.4 : Timing Analysis of Systems using level-triggered latched

[Source : Sung-Mo kang, Yusuf leblebici, Chulwoo Kim —CMOS Digital Integrated Circuits:Analysis & Design]

skew t_{skew} = 0 nsec.

In Fig. 6 we give an epoch tracing in the top portion and the clock diagram at the lower portion of the circuit. From the epoch tracing we find out that the operation is completed in 195 nsec. and can be accommodated in 2 cycles. For convenience we have chosen the period to be 98 nsec. In other words, we have a latency of only two cycles and with a clock period of 98 nsec. and it works out to be 196 nsec. It is left to the students to figure out that for the minimum logic delay defined by the expression in eqn.4, t_{skew} has no effect on the clock cycle time at all.

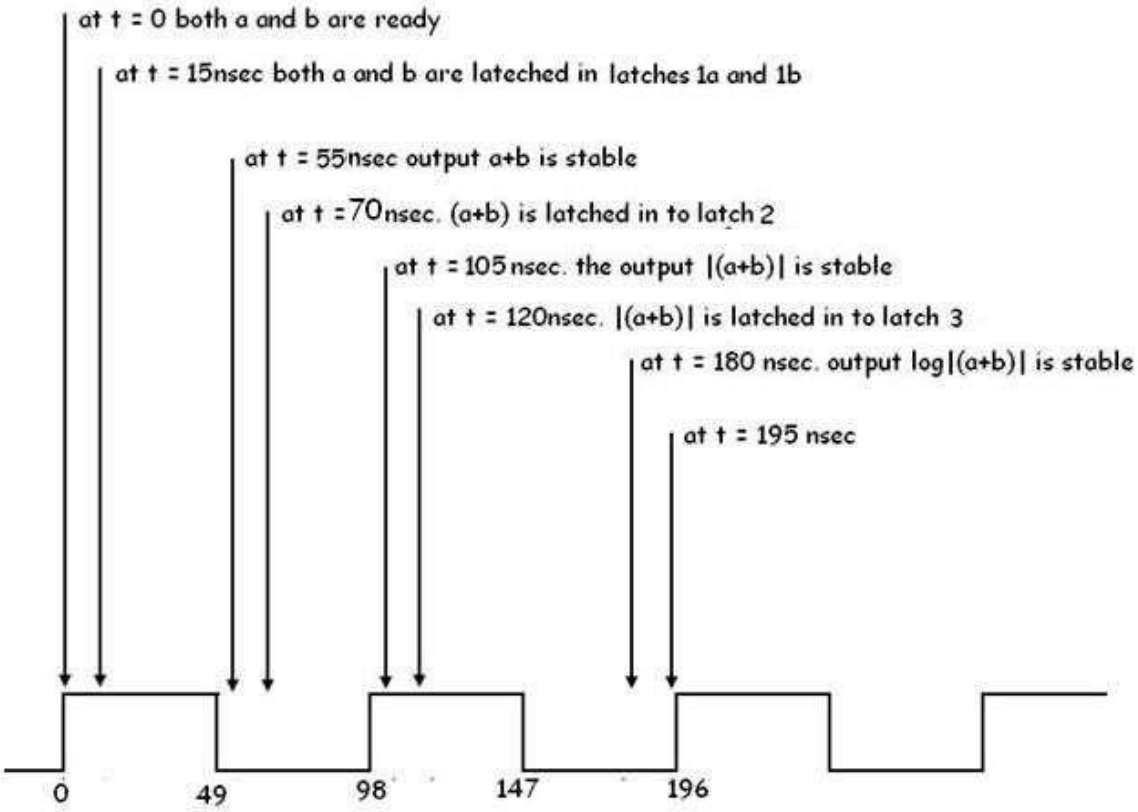


Fig. 6

has $t_{clock\text{-}out}$ and t_{hold} respectively 15nsec., and 5nsec. and the clock

Fig 3.5.5 : Timing Analysis of Systems output

[Source : Sung-Mo kang, Yusuf leblebici, Chulwoo Kim —CMOS Digital Integrated Circuits:Analysis & Design]

In summary, systems constructed from transparent latches eliminate the overhead of clock-skew and imbalance logic but still incur penalty introduced by data-out delays of the latches. Another major disadvantage with such a system would be distribution of two-phase clocks throughout the chip.

2 Timing Analysis of Systems using Positive Pulse Triggered Flip Flops:

A pulsed triggered flip flop behaves very similar to level triggered latches. The clock input to pulse triggered flip flop is a pulse with pulse width less than 50% of the clock cycle. Pulse triggered flip flops also have three terminals: data

input D, clock (pulse) input and output Q. The latch is transparent during the pulse width and input is copied to the output terminal. In a positive pulse triggered flip flop, when the clock goes low, previous value of the data is retained. These flip flops are characterized by their pulse width, setup time, hold time and Data-to-out delays.

Figure 7 illustrates system timing for a system built with pulsed latches. The timing constraints are also shown in the figure. The data must arrive at least setup time before the earliest falling transition of the clock. We reproduce the Max. Constraint and Min. constraints here for convenience:

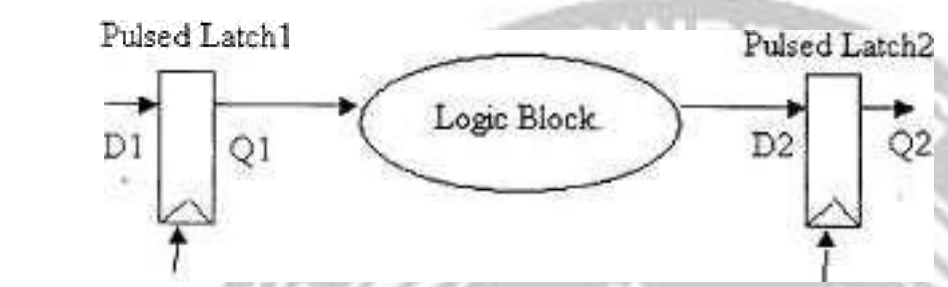
$$\{0, \quad T_{cycle} > t_{clock-out} + t_{logic}(\max) + Max \quad (t_{setup} + t_{skew} - t_{pulse-width})\} \tag{5}$$

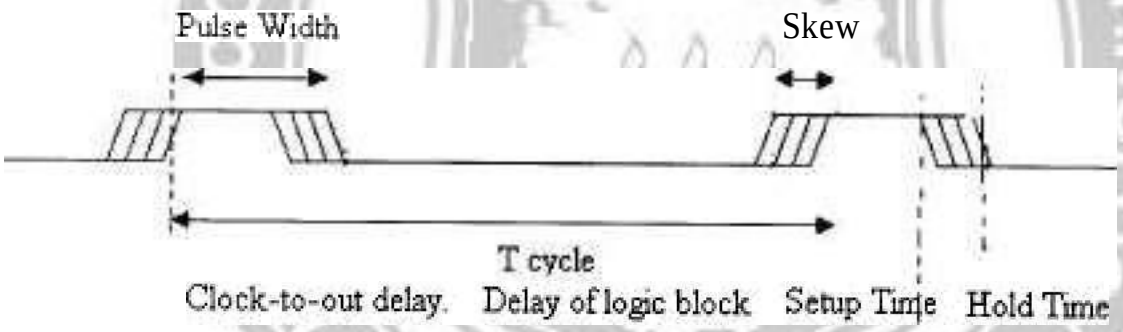
$$t_{logic}(\min) > t_{hold} - t_{clock-out} + t_{pulse-width} - t_{skew} \tag{6}$$

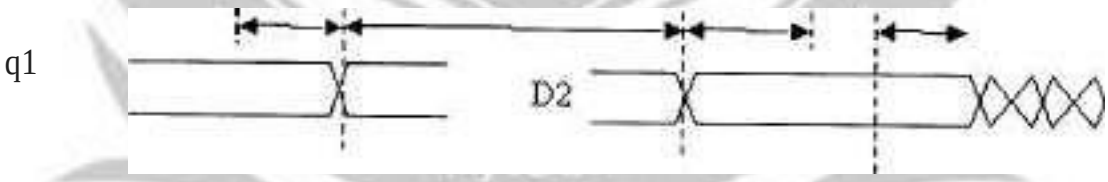
A wider pulse allows time borrowing where data from previous block can arrive late allowing it to borrow time from the subsequent stage. But the pulse width should also be short enough to avoid min delay failures arising when the delay of the logic block is shorter than the pulse width of the subsequent pulsed latch. It is difficult to distribute a clock pulse globally and usually a pulse is generated within the latch itself, which controls its operation. As we can see a pulsed latch allows time borrowing and provides better immunity against clock skews. The latency introduced by a pulsed latch is typically the data-to-out delay for data arriving in the pulse width.

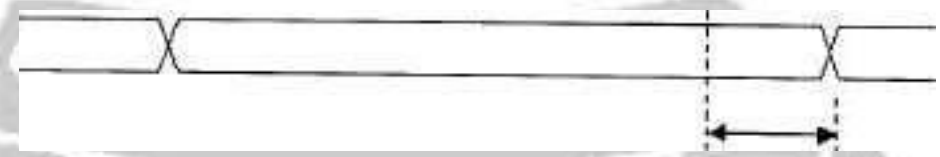
Unlike most of the edge triggered flip-flops, pulsed triggered flip flops have a positive hold time requirement (which is close to the pulse width). Hence this parameter also needs to be considered for a pulsed latch. Clock skew usually eats into the available time for logic. Pulsed latches absorb clock skew while flip-flops introduce hard edges and provide no immunity against clock skew.

Let us now consider the same example as in the earlier two cases. The circuit diagram will be the same as Fig. 2 with the registers being realized using pulse


Pulsed Latch1
Pulsed Latch2
Logic Block
D1
Q1
D2
Q2


Pulse Width
Skew
T cycle
Clock-to-out delay.
Delay of logic block
Setup Time
Hold Time


q1
D2



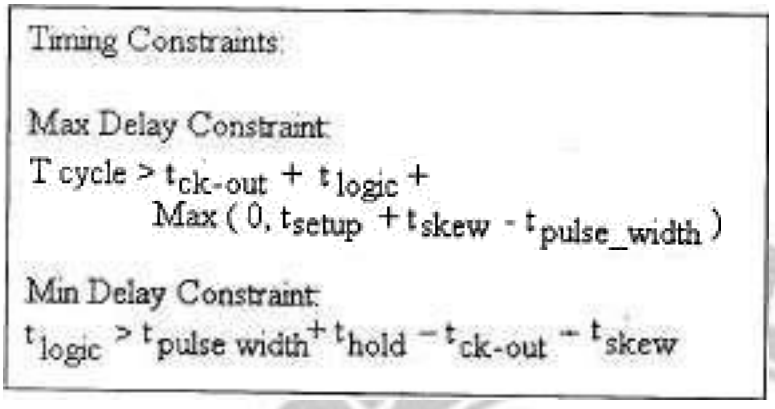


Fig 3.5.6 : Timing Constraints

[Source : Sung-Mo kang, Yusuf leblebici, Chulwoo Kim ―CMOS Digital Integrated Circuits:Analysis & Design]

triggered flip flops. In this case let $t_{clock-out}$, t_{setup}, t_{skew} and t_{hold} for the registers used be respectively 15nsec., 10nsec., 10nsec. and 15nsec. with the pulse width $t_{pulse-width}$ equal to 15 nsec. Using eqn. 5, we can evaluate the T_{cycle} to be 75 nsec. and the minimum logic delay to be 25 nsec. This configuration also gives a latency of three clock cycles which works out to be 225 nsec.

In summary, comparing the three methods of pipelining, pulse triggered flip flop based pipelining gives the highest possible frequency of operation. In pulse triggered flip flop based pipeline system, we do not distribute pulses but generate it internally for each flip flop and distribute only the global clock for all the flip flops.