pipelining

Pipelining is a popular design technique often used to accelerate the operation of the datapaths in digital processors. The idea is easily explained with the example of below figure.

The goal of the presented circuit is to compute log (|a - b|), where both a and b represent streams of numbers, that is, the computation must be performed on a large set of input values.

The minimal clock period T_{min} necessary to ensure correct evaluation is given as:

$$T_{min} = t_{c-q} + t_{pd,logic} + t_{su}$$

where t_{c-q} and t_{su} are the propagation delay and the set-up time of the register, respectively.

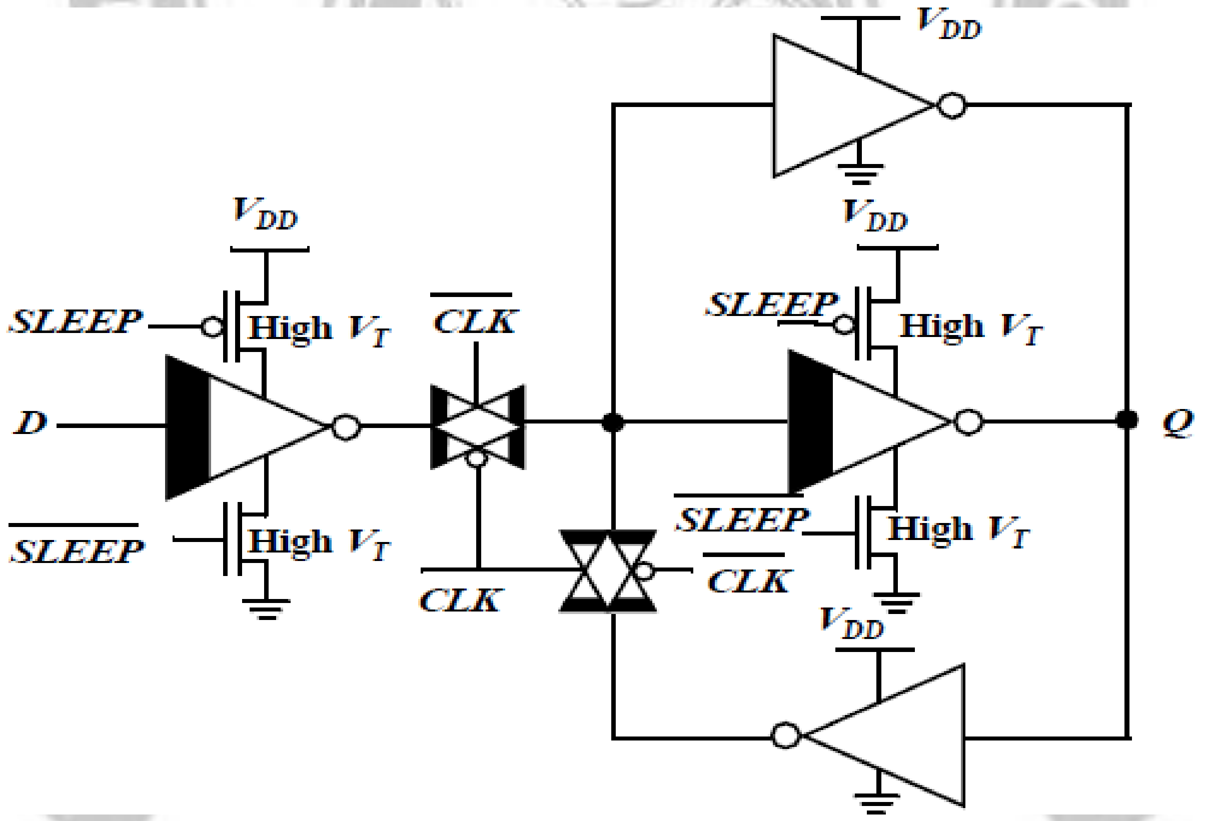

Fig 3.4.1: Datapath for computation

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

We assume that the registers are edge-triggered D registers. The term tpd, logic stands for the worst-case delay path through the combinatorial network, this consists of the adder, absolute value, and logarithm functions. In conventional systems (that don't push the edge of technology), the latter delay is generally much larger than the delays associated with the registers and dominates the circuit performance. Assume that each logic module has an equal propagation delay. We note that each logic module is then active for only 1/3 of the clock period (if the delay of the register is ignored).

For example, the adder unit is active during the first third of the period and remains idle— this is, it does no useful computation— during the other 2/3 of the period. Pipelining is a technique to improve the resource utilization, and increase the functional throughput. Assume that we introduce registers between the logic blocks.

The result for the data set (a1, b1) only appears at the output after three clock-periods. At that time, the circuit has already performed parts of the computations for the next data sets, (a2, b2) and (a3,b3). The computation is performed in an assembly-line fashion, hence the name pipeline.

The advantage of pipelined operation becomes apparent when examining the minimum clock period of the modified circuit. The combinational circuit block has been partitioned into three sections, each of which has a smaller propagation delay than the original function. This effectively reduces the value of the minimum allowable clock period.

Latch- vs. Register-Based Pipelines

Pipelined circuits can be constructed using level-sensitive latches instead of edge- triggered registers. Consider the pipelined circuit of below figure. The pipeline system is implemented based on pass-transistor-based positive and negative latches instead of edge triggered registers. That is, logic is introduced between the master and slave latches of a Master- slave system.

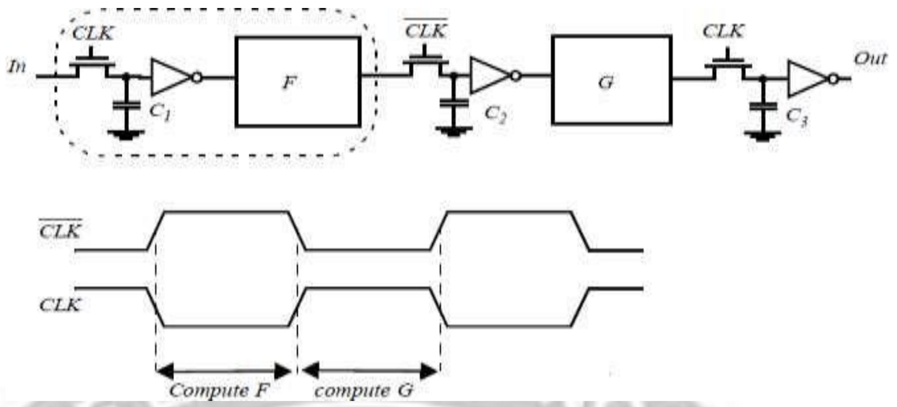

Fig 3.4.2 : Operation of two-phase pipelined circuit using dynamic registers
[Source : Sung-Mo kang, Yusuf leblebici, Chulwoo Kim —CMOS Digital Integrated Circuits:Analysis & Design …]

In the following discussion, we use without loss of generality the CLK-CLK notation to denote a two-phase clock system. Latch-based systems give significantly more flexibility in implementing a pipelined system, and often offer higher performance.

When the clocks CLK and CLK are non-overlapping, correct pipeline operation is obtained. Input data is sampled on C1 at the negative edge of CLK and the computation of logic block F starts; the result of the logic block F is stored on C2 on the falling edge of CLK, and the computation of logic block G starts. The non-overlapping of the clocks ensures correct operation. The value stored on C2 at the end of the CLK low phase is the result of passing the previous input (stored on the falling edge of C LK on C1) through the logic function F.

When overlap exists between CLK and CLK, the next input is already being applied to F, and its effect might propagate to C2 before CLK goes low (assuming that the contamination delay of F is small). Which value wins depends upon the logic functionF , the overlap time, and the value of the inputs since the propagation delay is often a function of the applied inputs.

NORA-CMOS-A Logic Style for Pipelined Structures

The latch-based pipeline circuit can also be implemented using CMOS latches, as shown in Figure. The operation is similar to the one discussed above. This topology has one additional, important property:

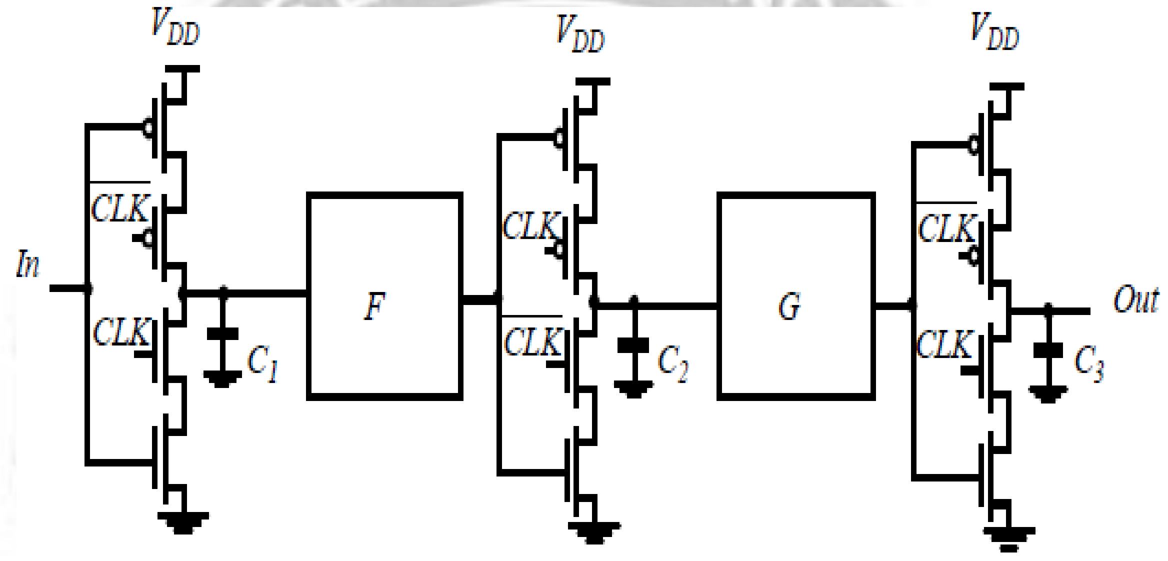

Fig 3.4.3: Pipeline datapath using CMOS latches.
[Source : Sung-Mo kang, Yusuf leblebici, Chulwoo Kim —CMOS Digital Integrated Circuits:Analysis & Design …]

A CMOS-based pipelined circuit is race-free as long as all the logic functions F (implemented using static logic) between the latches are non-inverting. The reasoning for the above argument is similar to the argument made in the construction of a C2MOS register. During a (0-0) overlap between CLK and CLK, all C2MOS latches, simplify to pure pull-up networks. The only way a signal can race from stage to stage under this condition is when the logic function F is inverting, as illustrated in below figure, where F is replaced by a single, static CMOS inverter.

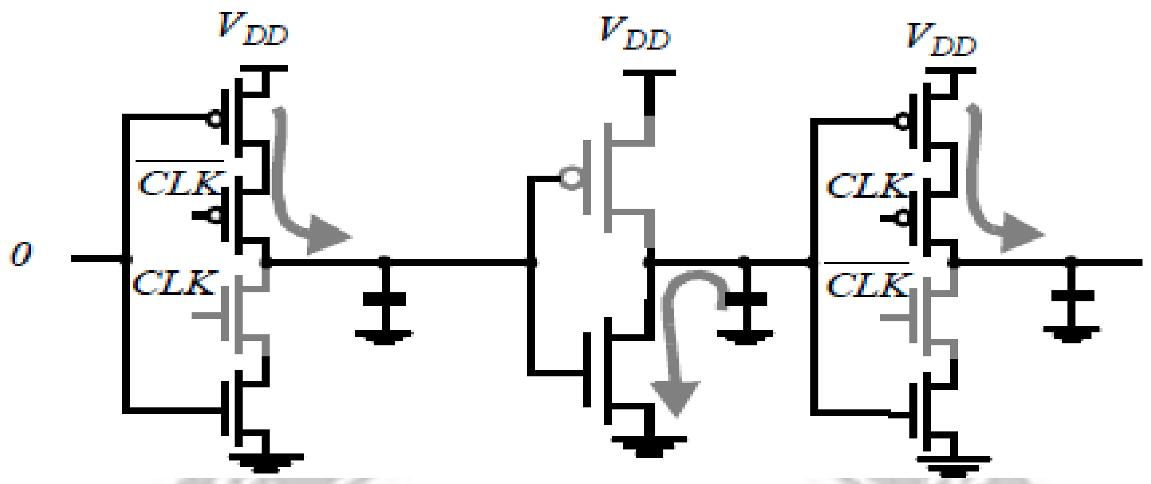

Fig 3.4.4 : Potential race condition during (0-0) overlap in CMOS-based design.
[Source : Sung-Mo kang, Yusuf leblebici, Chulwoo Kim —CMOS Digital Integrated Circuits:Analysis & Design …]

Similar considerations are valid for the (1-1) overlap. Based on this concept, a logic circuit style called NORA-CMOS; it combines CMOS pipeline registers and NORA dynamic logic function blocks. Each module consists of a block of combinational logic that can be a mixture of static and dynamic logic, followed by a CMOS latch. Logic and latch are clocked in such a way that both are simultaneously in either evaluation, or hold (precharge) mode. A block that is in evaluation during CLK = 1 is called a CLK-module, while the inverse is called a CLK- module.

A NORA Datapath consists of a chain of alternating CLK and CLK modules. While one class of modules is precharging with its output latch in hold mode, preserving the previous output value, the other class is evaluating. Data is passed in a pipelined fashion from module to module.

NORA offers designers a wide range of design choices. Dynamic and static logic can be mixed freely, and both CLKp and CLKn dynamic blocks can be used in pipelined form. With this freedom of design, extra inverter stages, as required in DOMINO-CMOS, are most often avoided.In order to ensure correct operation, two important rules should always be followed:

- **The dynamic-logic rule:** Inputs to a dynamic CLKn (CLKp) block are only allowed to make a single 0 (1 0) transition during the evaluation period.

- **The CMOS rule:** In order to avoid races, the number of static inversions between CMOS latches should be even