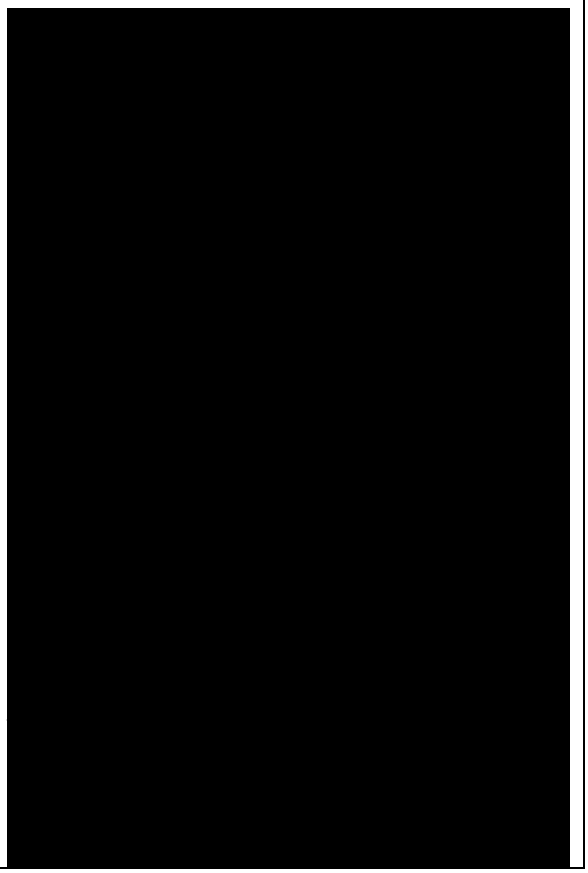
7.6 Pulse Registers 7 h 11 19/6

Until now, we have used the master-slave configuration to create an edge-triggered register. A fundamentally different approach foiconstructing a register uses pulse signals. The idea is to construct a short pulse around the rising (sườn dương, sườn lên) or falling) edge (sườn âm, sườn xuống) of the clock. This pulse acts as the clock input to a latch (e.g., a TSPC (đồng hồ đơn pha thực) flavor is shown in Figure 7.35a), sampling the input only in a short window. Race conditions (điều kiện tương tranh, điều kiện tranh đua, điều kiện cạnh tranh, điều kiện tranh chấp) are thus avoided by keeping the opening time (i.e, the transparent period) of the latch very short. The combination of the glitch-generation circuitry and the latch results in apositive edge-triggered register.

Figure 7.35b shows an example circuit for constructing a short intentional glitch n each rising edge of the clock [Kozo96]. When CLK = 0, node X is charged up to VDD (MN is off since CLKG is low). On the rising edge of the clock, there is a short period of time when both inputs of the AND gate are high causing CLKG to go high. This in turn activates Mn, pulling X and eventually CLKG low (Figure 7.35c). The length of the pulse is controlled by the delay of theAND gate and the two inverters. Note that thereexists also a delay between the rising edges of the input clock (CLK) and the glitch clock (CLKG) — also equal to the delay of the AND gate and the two inverters. If every

7.6 Các thanh ghi xung

Cho đến thời điểm này, chúng ta đã dùng cấu hình chủ-tớ để tạo ra thanh ghi chuyển mạch theo cạnh (sườn). Một phương pháp khác để thiết kế thanh ghi là dùng các tín hiệu xung. Ý tưởng cơ bản là tạo ra một xung ngắn quanh sườn lên hoặc xuống của một xung nhịp. Xung này đóng vai trò như đầu vào của mạch chốt (ví dụ, flavor TSPC biểu diễn trong hình 7.35a), lấy mẫu đầu vào chỉ trong một cửa sổ ngắn. Do đó có thể tránh được các điều kiện tương tranh bằng cách giữ cho thời gian mở (tức là khoảng thời gian trong suốt) của mạch chốt rất ngắn. Sự kết hợp của các mạch tạo glitch và mạch chốt tạo ra thanh ghi chuyển mạch theo cạnh (sườn) dương.



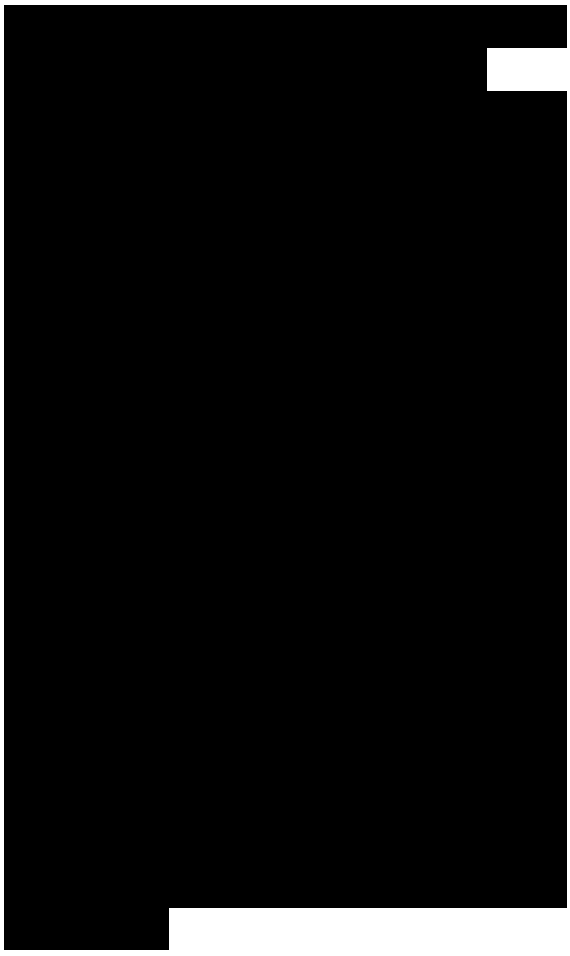
register on the chip uses the same clock generation mechanism,this sampling delay does not matter. However, process variations and load variations may cause the delays through the glitch clockcir- cuitry to be different. This must be taken into account when performing timing verification and clock skew analysis (which is the topic of a later Chapter).

CLK

CLKG

(c) glitch clock

Figure 7.35 Glitch latch - timing generation and register.



If set-up time and hold time are measured in reference to the rising edge of the glitch clock, the set-up time is essentially zero, thehold time is equal to the length of the pulse (if the contamination delay is zero for the gates), and the propagation delay (t) equals two gate delays. The advantage of the approach is the reduced clock load and thesmall number of transistors required. The glitch-generation circuitry can be amortized over multiple register bits. The disadvantage is a substantial increase in verification complexity. This has prevented a wide-spread use. They do however provide an alternate approach to conventional schemes, and have been adopted in some high performance processors (e.g., [Kozo96]).

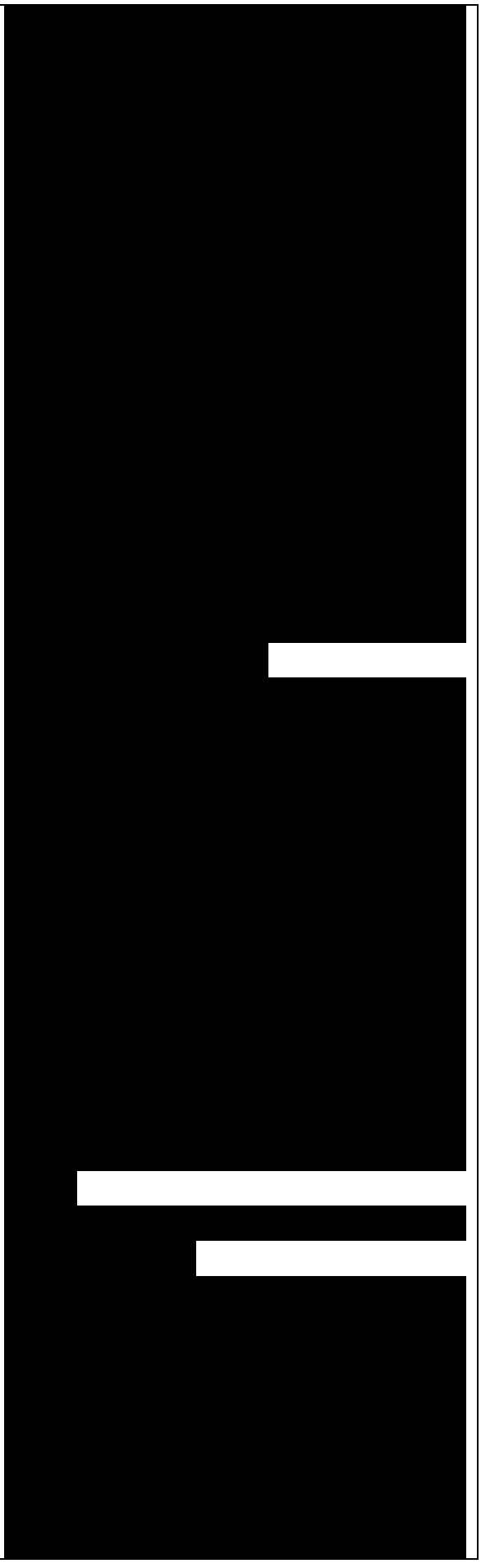
Another version of the pulsed registeris shown in Figure 7.36 (as used in the AMD- K6 processor [Partovi96]). When the clock is low, M3 and M6 are off and device P1 is turned on. Node X is precharged to VDD, the output node (Q) is

decoupled from X and is held at its previous state. CLKD is a delay-inverted version of CLK. On the rising edge of the clock, M3 and M6 turn on while devices M1 and M4 stay on for a short period determined by the delay of the three inverters. During this interval, the circuit is transparent and the input data D is sampled by the latch. Once CLKD goes low, node X is decoupled from the D input and is either held or starts to precharge toVDD by PMOS device P2. On the falling edge of the clock, nodeX is held at VDD and the output is held stable by the cross-coupled inverters.

Note that this circuit also uses a one-shot, but the one-shot is integrated into the register. The transparency period also determines the hold time of the register. The window must be wide enough for the input data to propagate to theQ output. In this particular circuit, the set-up time can be negative. This is the case if the transparency window is longer than the delay from input to output. This is attractive, as data can arrive at the register even after the clock goes high, which means that time is borrowed from the previous cycle.

Example 7.6 Set-up time of glitch register

The glitch register of Figure 7.36 is transparent during the (1-1) overlap of CLK and CLKD. As a result, the input data can actuallychange after the rising edge of the clock, resulting in a negative set-up time (Figure 7.37). The D-input transitions to low after the rising edge of the clock, and



transitions high before the falling edge oCLKD (this is, during the transparency period). Observe how the output follows the input The output Q does go to the correct value of VDD as long as the input D is set up correctly some time before the falling edge oCLKD. When the negative set-up time is exploited, there can be no gurantees on the monotonic behavior of the output. That is, the output can have multiple transitions around the rising edge, and therefore, the output of the register should not be used as a clock to other registers.

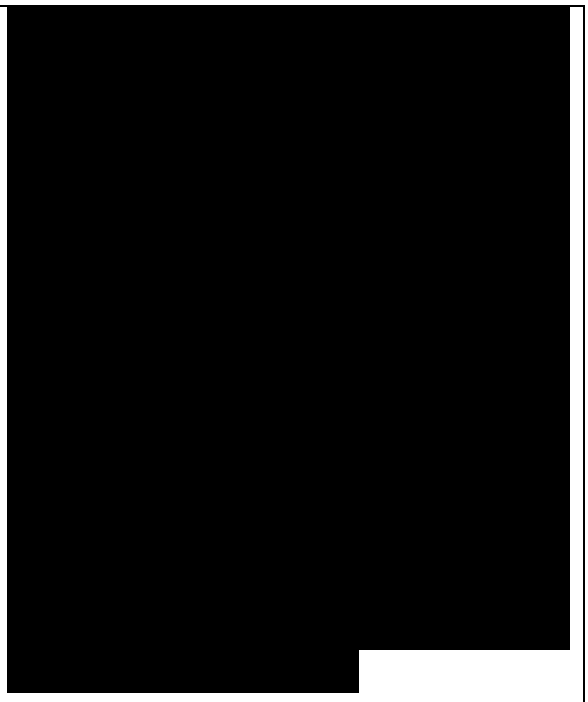


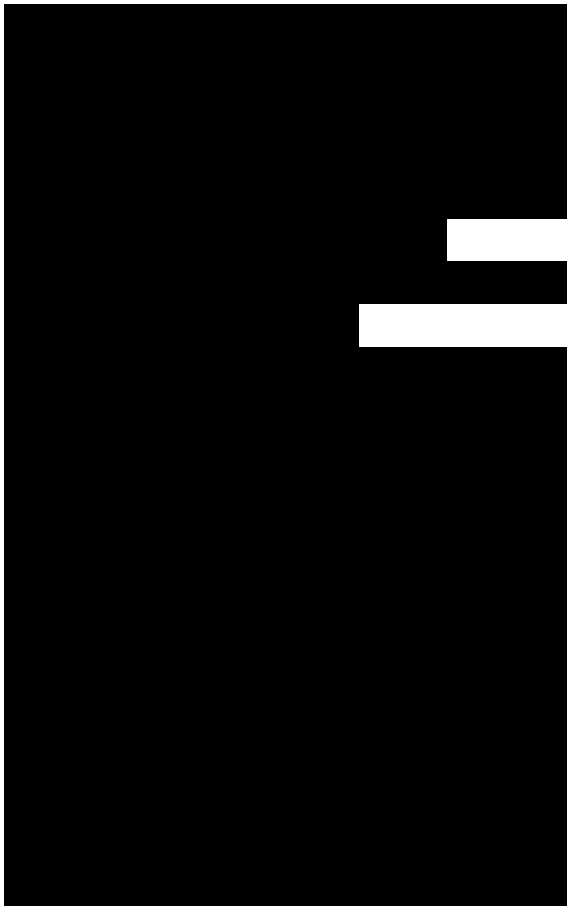
Figure 7.37 Simulation showing a negative set-up time for the glitch register.

Problem 7.7 Converting a glitch register to a conditional glitch register

Modify the circuit in Figure 7.36 sothat it takes an additioral Enable input. The goal is to convert the register to a conditional register which latches only when the enable signal is asserted.

7.7 Sense-Amplifier Based Registers

So far, we have presented two fundamental approaches towards building edge-triggered registers: the master-slave concept and the glitch technique. Figure 7.38 introduces another technique that uses a sense amplifier structure to implement an edge-triggered register [Montanaro96]. Sense amplifier circuits accept small input signals and amplify them to generate rail-to-rail swings (biên điện áp, dao động điện áp, dải điện áp). As we will see,



sense amplifier circuits are used extensively in memory cores and in low swing bus drivers to amplify small voltage swings

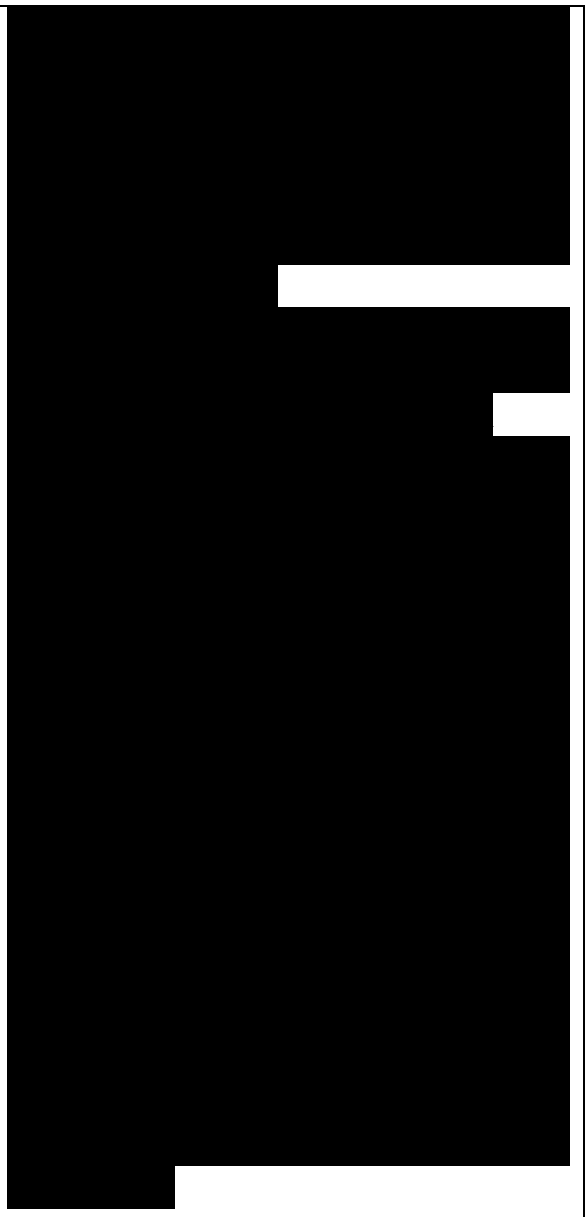


Figure 7.38 Positive edge-triggered register based on sense-amplifier

present in heavily loaded wires. There are many techniques to construct these amplifiers, with the use of feedback (e.g., cross-coupled inverters) being one common approach. The circuit shown in Figure 7.38 uses a precharged front-end amplifier that samples the differential input signal on the rising edge of the clock signal. The outputs of front-end are fed into a NAND cross-coupled SR FF that holds the data and gurantees that the differential outputs switch only once per clock cycle. The differential inputs in this implementation dont have to have rail-to-rail swing and hence this register can be used as a receiver for a reduced swing differential bus.

The core of the front-end consists of a cross-coupled inverter M5-M8) whose outputs (L, and i2) are precharged using devices M9 and M10 during the low phase of the clock. As a result, PMOS transistors M7 and M8 to be turned off and the NAND FF is holding its previous state. Transistor M, is similar to an evaluate switch in dynamic circuits and is turned off ensuring that the differential inputs dont affect the output during the low phase of the clock. On the rising edge of the

clock, the evaluate transistor turnon and the differential input pair M2 and M3) is enabled, and the difference between the input signals is amplified on the output nodes oni, and i2. The cross-coupled inverter pair flips to one of its the stable states based on the value of the inputs. For example, UN is 1, L, is pulled to 0, and L2 remains at VDD. Due to the amplifying properties of the input stage, it is not necessary for the input to swing all the way up toVDD and enables the use of low-swing signaling on the input wires.

The shorting transistor, M4, is used to provide a DC leakage path from either node i3, or i4, to ground. This is necessary to accommodate the case where the inputs change their value after the positive edge ofCLK has occurred, resulting in either L3 or L4 being left in a high-impedance state with a logical low voltage level stored on the node. Without the leakage path that node would be susceptible to charging by leakage currents. The latch could then actually change state prior to the next rising edge ofCLK! This is best illustrated graphically, as shown in Figure 7.39

7.8 Pipelining: An approach to optimize sequential circuits

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 Figure 7.40a. The goal of the presented circuit is to computelog(|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 periodTmin necessary to ensure correct evaluation is given as:
T = t +1 +1
min c-q pd,logic su

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

where t and tsu are the propagation delay and the set-up time of the register, respectively. We assume that the registers areedge-triggered D registers. The term tpc,logic stands for the worst-case delay path through the combinatorial network, which consists of the adder, absolute value, and logarithm functions. In conventional systems (that dont 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 eachl ogic 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 idl— this isit 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, as shown in Figure 7.40b. This causes the computation for one set of input data to spread over a number of clock periods, as shown in Table 7.1.
(a) Nonpipelined version
T
CLK

Figure 7.40 Datapath for the computation of log(| + b|).

302

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.

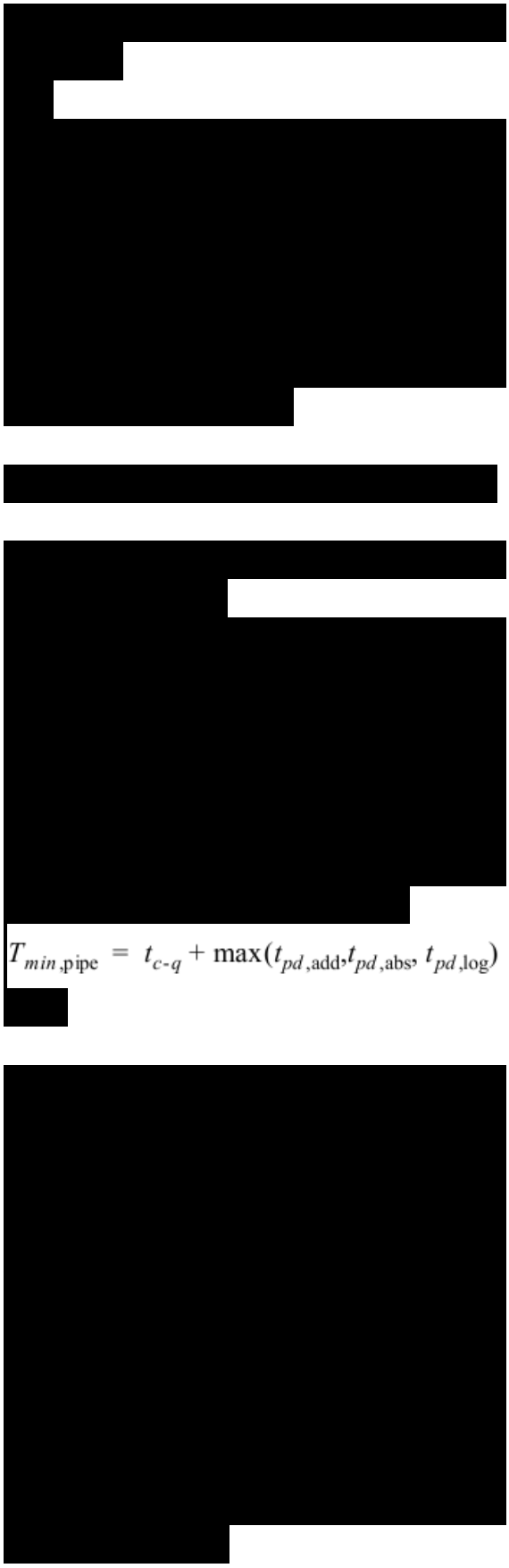
Table 7.1 Example of pipelined computations.

Clock Period Adder Absolute Value Logarithm

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 smallepropagation delay than the original function. This effectively reduces the value of the minimum allowable clock period:

Tmin,pipe tc-q + max(tpd,add,tpd,abs> tpd,log) (7.7)

—Suppose that all logic blocks have approximately the samepropagation delay, and that the register overhead is small with respect to the logic delays. The pipelined network outperforms the original circuit by a factor of three under these assumptions, oT minpipe= Tmin/3. The increased performance comes at the relatively small cost of two additional registers, and an increased latency1 This explains why pipelining is popular in the implementation of very high-performance datapaths.



7.8.1 Latch- vs. Register-Based Pipelines

Pipelined circuits can be constructed using level-sensitive latches instead of edge-triggered registers. Consider the pipelined circuit of Figure 7.41. The pipeline system is implemented based on pass-transistor-basedpositive and negative latches instead of edge- triggered registers. That is, logic is introduced between the master and slave latches of a master-slave system. 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 offers higher performance. When the clocks CLK and CLK are nonoverlapping, correct pipeline operation is obtained. Input data is sampled onC1 at the negative edge ofCLK and the computation of logic block F starts; the result of the logic blockF is stored on C2 on the falling edge of CLK, and the computation of logic block G starts. The nonoverlapping of the clocks

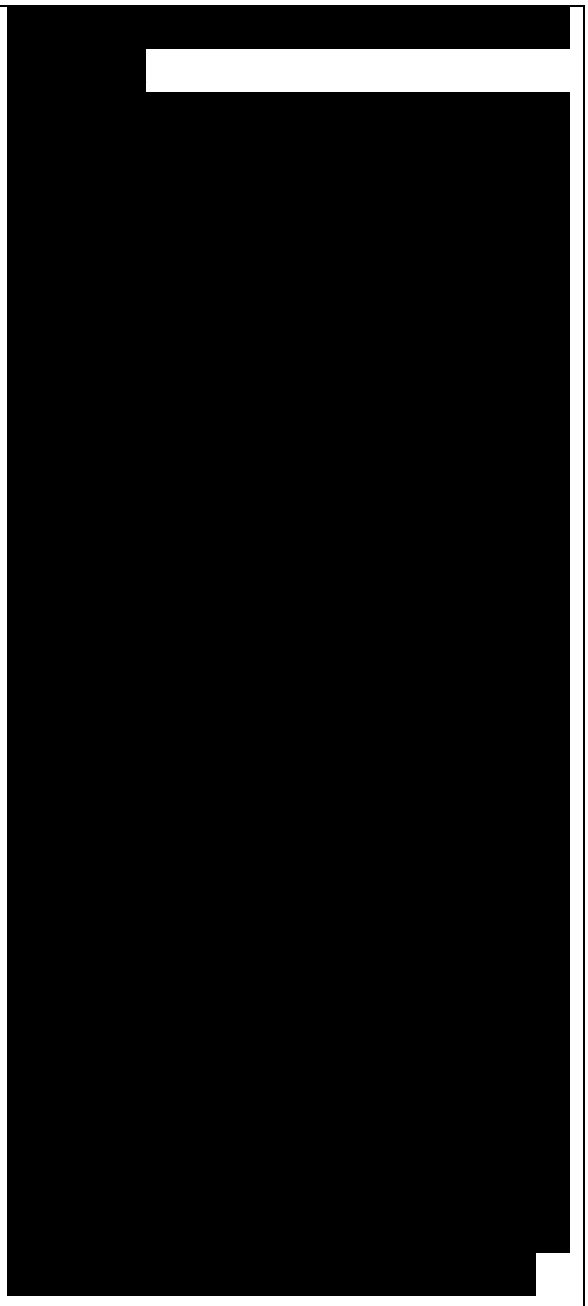


Figure 7.41 Operation of two-phase pipelined circuit using dynamic registers.

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 oCLK 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

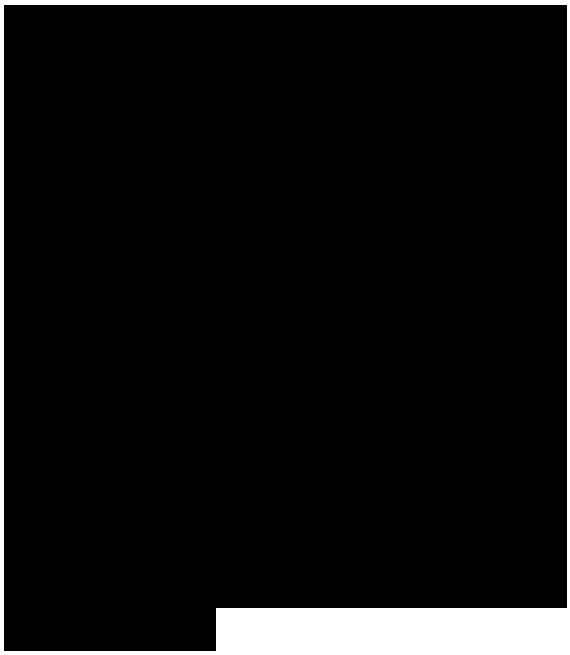
the contamination delay of F is small). In other words, a race develops between the previous input and the current one. Which value wins depends upon the logic function7, the overlap time, and the value of the inputs since thepropagation delay is often a function of the applied inputs. The latter factor makes the detection and elimination of race conditions non-trivial.

7.8.2 NORA-CMOS— A Logic Style for Pipelined Structures

The latch-based pipeline circuit can also be implemented using C?MOS latches, as shown in Figure 7.42. The operation is similar to the one discussed above. This topology has one additional, important property:

A C2MOS-based pipelined circuit is race-free as long as all the logic functionsF (implemented using static logic) between the latches are noninverting.

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 (see Figure 7.27). The only way a signal can race from stage to stage under this condition is when the logic functionF is inverting, as illustrated in Figure 7.43, where F is replaced by a single, static CMOS inverter. Similar considerations are valid for the (1-1) overlap.



Based on this concept, a logic circuit style calledNORA-CMOS was

conceived [Goncalves83]. It combines C2MOS 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

Figure 7.43 Potential race condition during (0-0) overlap in (?MOS-based "=" ~ design.

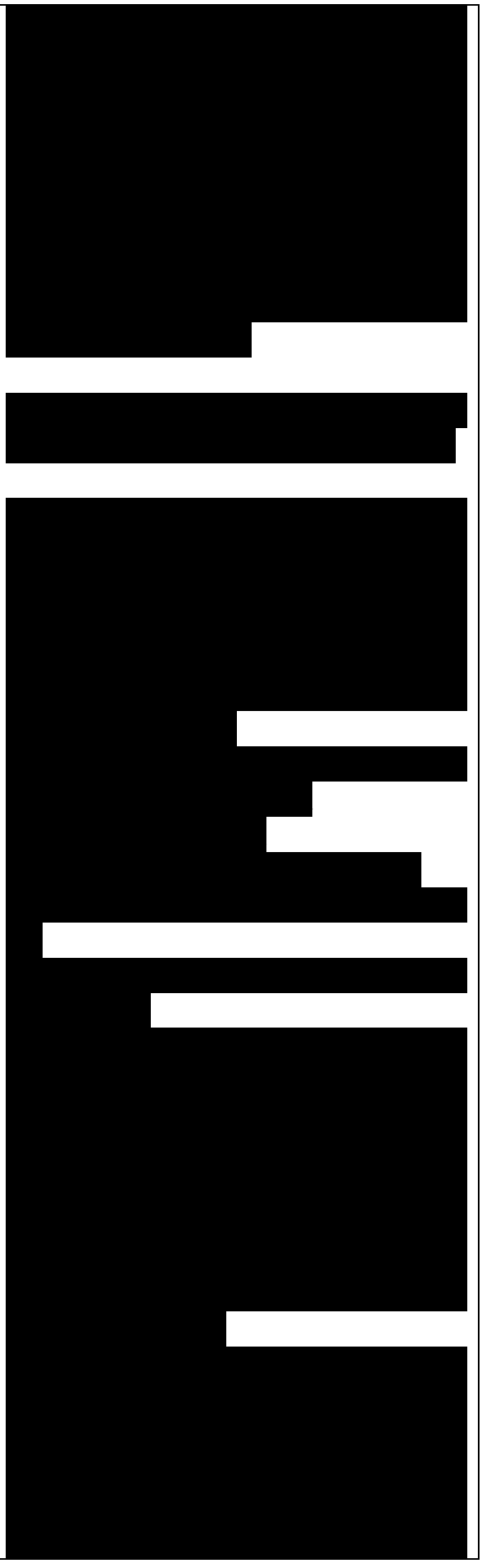
that is in evaluation duringCLK = 1 is called a CLK-module, while the inverse is called a CLK-module. Examples of both classes are shown in Figure 7.44 a and b, respectively. The operation modes of the modules are summarized in Table 7.2.

Table 7.2 Operation modes for NORA logic modules.

	CLK block Logic	Latch	CLK block Logic	Latch
CLK = 0	Precharge	Hold	Evaluate	Evaluate
CLK = 1	Evaluate	Evaluate	Precharge	Hold

A NORA datapath consists of a chain of alternatingCLK 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 bothCLKp and CLKn dynamic blocks can be used in cascaded or in pipelined form. With this freedom of



design, extra inverter stages, as required in DOMINO-CMOS, are most often avoided.

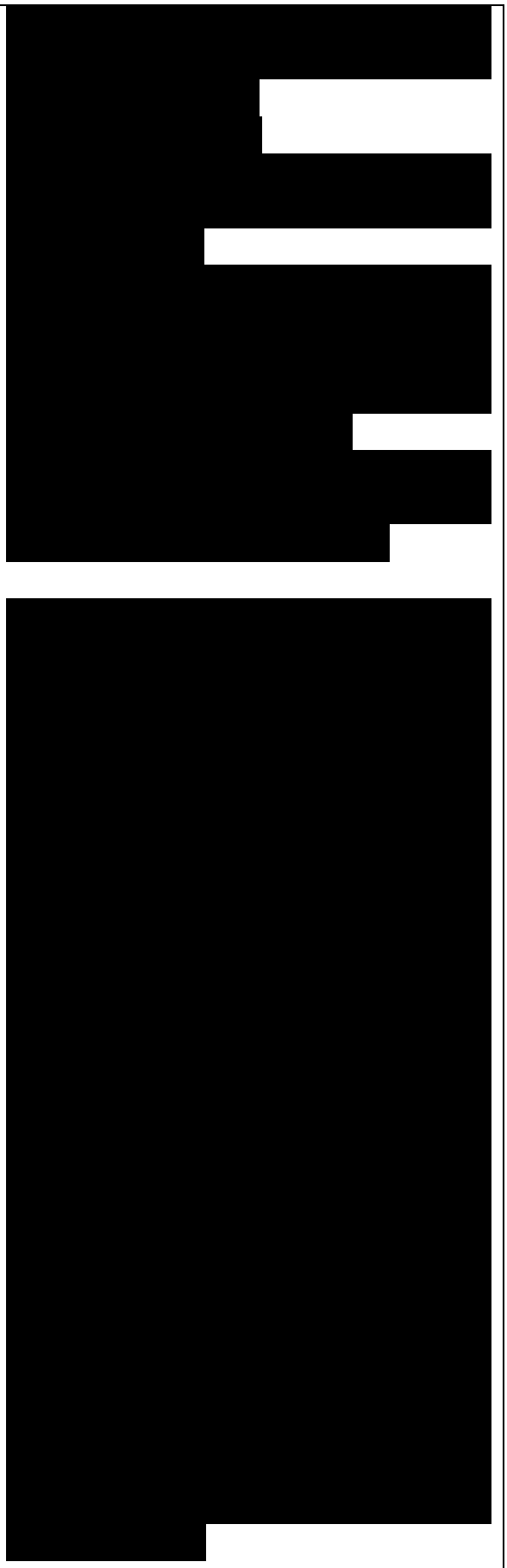
Design Rules

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 (1 ® 0) transition during the evaluation period (Chapter6).

The C2MOS rule: In order to avoid races, the number of static inversions between C2MOS latches should be even.

The presence of dynamic logic circuits requires the introduction of some extensions to the latter rule. Consider the situation pictured in Figure 7.45a. During precharge CLK = 0), the output register of the module has to be in hold mode, isolating the output node from the internal events in the module. Assume now that a (0-0) overlap occurs. Node4 gets precharged to VDD, while the latch simplifies to a pull-up network (Figure 7.45b). It can be observed that under those circumstances the output node charges toVDD, and the stored value is erased! This malfunctioning is caused by the fact that the number of static inversions between the last dynamic node in the module and the latch is odd, which creates an active path between the precharged node and the output. This translates into the following rule: The number of static inversions between the last dynamic block in a logic function and the C2MOS latch should be even. This and similar considerations



lead to a reformulated C2MOS rule [Goncalvez83].

(a) Circuit with odd number of static inversions between dynamic logic stage and register

Figure 7.45 Extended C2MOS rules.

Revised C2MOS Rule

The number of static inversions betweenC2MOS latches should be even (in the absence of dynamic nodes); if dynamic nodes are present, the number of static inverters between a latch and a dynamic gate in the logic block should be even. The number of static inversions between the last dynamic gate in a logic block and the latch should be even as well.

Adhering to the above rules is not always trivial and requires a careful analysis of the logic equations to be implemented. This often makes the design of an operational NORA-CMOS structure cumbersome. Its use should only be considered when maximum circuit performance is a must.