Chapter 11

Flip-Flops

Contents

Preliminary questions

1. How is time measured in a synchronous circuit?
2. What is the functionality of a flip-flop?
3. What is a stable state? How many stable states does a flip-flop have?
4. How does a flip-flop move from one stable state to another? How fast is this transition?

In this chapter we introduce a memory device called a flip-flop. The definition of flip-flops is rather elaborate and requires that the input be stable during a critical segment. We prove that flip-flops with empty critical segments do not exist.

11.1 The clock

Synchronous circuits depend on a special signal called the *clock*. In practice, the clock is generated by rectifying and amplifying a signal generated by special non-digital devices (i.e. crystal oscillators). Since our course is about digital circuits, we use the following abstraction to describe the clock.

Definition 11.1 *A* clock *is a periodic logical signal that oscillates instantaneously between logical one and logical zero. There are two instantaneous transitions in every clock period: (i) in the beginning of the clock period, the clock transitions instantaneously from zero to one; and (ii) at some time in the interior of the clock period, the clock transitions instantaneously from one to zero.*

Figure 11.1 depicts a clock signal. We use the convention that the clock rise occurs in the beginning of the clock period. Note that we assume that the transitions of the clock signal are instantaneous; this is obviously impossible in practice. We show later how we get around this unrealistic assumption.

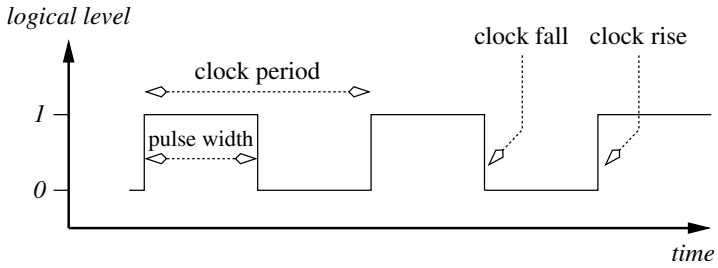

Figure 11.1: A clock signal.

Notation. We denote the clock signal by CLK. We refer to the period of time within a clock period during which the clock equals one as the *clock pulse* (see Fig. 11.1). We denote the clock period by $\varphi(\text{CLK})$. We denote the duration of the clock pulse by CLK_{pw}. A clock signal CLK is *symmetric* if $\text{CLK}_{pw} = \varphi(\text{CLK})/2$. A clock is said to have *narrow pulses* if $\text{CLK}_{pw} < \varphi(\text{CLK})/2$. A clock is said to have *wide pulses* if $\text{CLK}_{pw} > \varphi(\text{CLK})/2$. See Figure 11.2 for three examples.

Clock cycles. A clock partitions time into discrete intervals. Throughout this chapter we denote the starting time of the ith clock periods by t_i. We refer to the half-closed interval $[t_i, t_{i+1})$ as *clock cycle* i. One could use open or closed intervals instead; our convention avoids overlaps or gaps between clock periods.

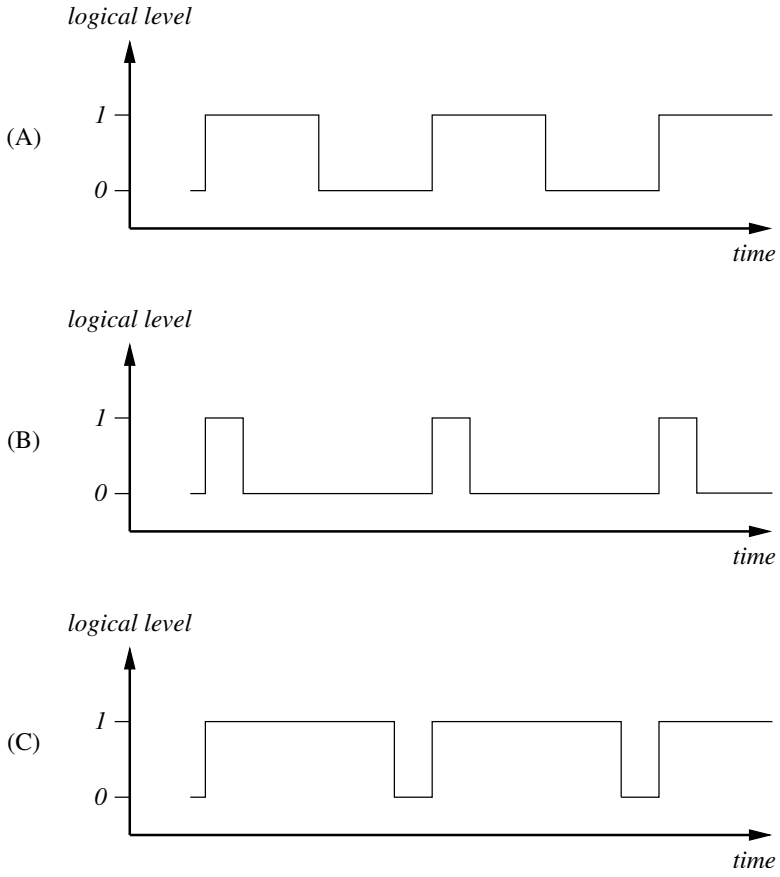

Figure 11.2: (A) A symmetric clock (B) A clock with narrow pulses (C) A clock with wide pulses.

11.2 Edge-triggered Flip-Flop

In this section we define edge-triggered flip-flops.

Definition 11.2 *An edge-triggered flip-flop is defined as follows.*

Inputs: *A digital signal $D(t)$ and a clock* CLK.

Output: *A digital signal $Q(t)$.*

Parameters: *Four parameters are used to specify the functionality of a flip-flop:*

- Setup-time *denoted by* t_{su},
- Hold-time *denoted by* t_{hold},
- Contamination-delay *denoted by* t_{cont}, *and*
- Propagation-delay *denoted by* t_{pd}.

These parameters satisfy $-t_{\text{su}} < t_{\text{hold}} < t_{\text{cont}} < t_{\text{pd}}$. *We refer to the interval* $[t_i - t_{\text{su}}, t_i + t_{\text{hold}}]$ *as the* critical segment C_i *and to the interval* $[t_i + t_{\text{cont}}, t_i + t_{\text{pd}}]$ *as the* instability segment A_i. *See Figure 11.3 for a depiction of these parameters.*

Functionality: *If $D(t)$ is stable during the critical segment C_i, then $Q(t) = D(t_i)$ during the interval* $(t_i + t_{\text{pd}}, t_{i+1} + t_{\text{cont}})$.

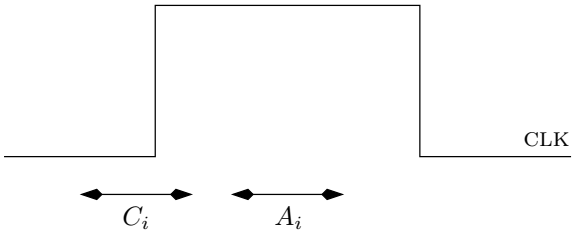

Figure 11.3: The critical segment $C_i = [t_i - t_{su}, t_i + t_{hold}]$ and instability segment $A_i = [t_i + t_{cont}, t_i + t_{pd}]$ corresponding the clock period starting at t_i.

The definition of edge-triggered flip-flops is a rather complicated, so we elaborate.

1. The assumption $-t_{su} < t_{hold} < t_{cont} < t_{pd}$ implies that the critical segment C_i and the instability segment A_i are disjoint.

2. If $D(t)$ is stable during the critical segment C_i, then the value of $D(t)$ during the critical segment C_i is well defined and equals $D(t_i)$.

3. The flip-flop *samples* the input signal $D(t)$ during the critical segment C_i. The sampled value $D(t_i)$ is output during the interval $[t_i + t_{pd}, t_{i+1} + t_{cont}]$. Sampling is successful only if $D(t)$ is stable while it is sampled. This is why we refer to C as a critical segment.

4. If the input $D(t)$ is stable during the critical segments $\{C_i\}_i$, then the output $Q(t)$ is stable in between the instability segments $\{A_i\}_i$.

5. The stability of the input $D(t)$ during the critical segments depends on the clock period. We will later see that slowing down the clock (i.e., increasing the clock period) helps in achieving a stable $D(t)$ during the critical segments.

Figure 11.4 depicts a schematic of an edge-triggered flip-flop. Note the special "arrow" that marks the clock-port. We refer to an edge-triggered flip-flop, in short, as a flip-flop.

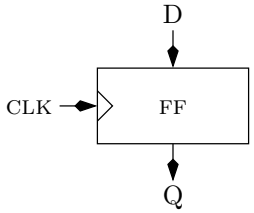

Figure 11.4: A schematic of an edge-triggered flip-flop

Question 11.1 *Prove that an edge-triggered flip-flop is not a combinational circuit.*

11.3 Arbitration

Arbitration in the context of digital design is the problem of deciding which event occurs first. For the sake of simplicity we focus on the event that the digital interpretation of an analog signal becomes 1. Hence, an arbiter is supposed to determine which of two signals reaches first the value one. We formally define arbitration as follows.

Definition 11.3 *An* arbiter *is a circuit defined as follows.*

Inputs: *Non-decreasing analog signals $A_0(t), A_1(t)$ defined for every $t \geq 0$.*

Output: *An analog signal $Z(t)$.*

Functionality: *Assume that $A_0(0) = A_1(0) = 0$. Define T_i, for $i = 0, 1$, as follows:*

$$T_i \stackrel{\triangle}{=} \inf\{t \mid \mathrm{dig}(A_i(t)) = 1\}.$$

Let $t' \stackrel{\triangle}{=} 10 + \max\{T_0, T_1\}$. The output $Z(t)$ must satisfy, for every $t \geq t'$,

$$\mathrm{dig}(Z(t)) = \begin{cases} 0 & \text{if } T_0 < T_1 - 1 \\ 1 & \text{if } T_1 < T_0 - 1 \\ 0 \text{ or } 1 & \text{otherwise.} \end{cases}$$

Note that if T_0 or T_1 equals infinity, then t' equals infinity, and there is no requirement on the output $Z(t)$. The idea is that the arbiter circuit is given 10 time units starting from $\max\{T_0, T_1\}$ to determine if $T_0 < T_1$ or $T_1 < T_0$. We refer to the case in which $|T_0 - T_1| \leq 1$ as a "tie". The arbiter is not required to make a specific decision if a tie occurs. However, even in the case of a tie, the arbiter must make some decision after 10 time units and its output $Z(t)$ must have a logical value.

Arbiters are very important in many applications since an arbiter determines the order between events. For example, an arbiter can determine which message arrived first in a network switch.

We will show in this chapter that, under very reasonable assumptions, arbiters do not exist. Moreover, we will show that a flip-flop with an empty critical segment can be used to implement an arbiter. The lesson is that, without critical segments, flip-flops do not exist.

11.4 Arbiters - an impossibility result

In this section we prove that arbiters do not exist.

Claim 11.1 *There does not exist a circuit C that implements an arbiter.*

Proof: Let C denote a circuit with inputs $A_0(t), A_1(t)$ and output $Z(t)$. Define $A_0(t)$ to be the analog signal that rises linearly in the interval $[0, 100]$ from 0 to $V_{high,in}$, and for every $t \geq 100$, $A_0(t) = V_{high,in}$. Let x denote a parameter that defines $A_1(t)$ as follows: $A_1(t)$ rises linearly in the interval $[0, 100+x]$ from 0 to $V_{high,in}$, and for every $t \geq 100+x$, $A_1(t) = V_{high,in}$. Let $f(x)$ denote the function that describes the value of $Z(200)$ (i.e., the value of $Z(t)$ at time $t = 200$) when fed by the signals $A_0(t)$ and $A_1(t)$. We study the function $f(x)$ in the interval $x \in [-2, 2]$. We make the following observations:

1. $f(-2) \geq V_{high,out}$. The reason is that if $x = -2$, then $T_0 = 100$ and $T_1 = 98$. Hence $A_1(t)$ "wins", and by time $t = 200$, the arbiter's output should stabilize on the logical value 1.

2. $f(2) \leq V_{low,out}$. The reason is that if $x = 2$, then $T_0 = 100$ and $T_1 = 102$. Hence $A_0(t)$ "wins", and $dig(Z(200)) = 0$.

3. $f(x)$ is continuous in the interval $[-2, 2]$. This is not a trivial statement and its formal proof is not within the scope of this course. We provide an intuitive proof of this fact. The idea of the proof of the continuity of $f(x)$ is that the output $Z(200)$ depends on the following: (i) The initial state of the device C at time $t = 0$. We assume that the device C is in a stable state and that the charge is known everywhere. (ii) The signal $A_i(t)$ in the interval $[0, 200]$, for $i = 0, 1$.

   An infinitesimal change in x affects only $A_1(t)$ (i.e., the initial state of the circuit and $A_0(t)$ are not affected by x). Moreover, the difference in energy of $A_1(t)$ corresponding to two very close values of x is infinitesimal. Hence, we expect the

difference in $Z(200)$ for two very close values of x to be also infinitesimal. If this were not the case, then noise would cause uncontrollable changes in $Z(t)$ and the circuit C would not be useful anyhow.

By the Mean Value Theorem, it follows that, for every $y \in [V_{low,out}, V_{high,out}]$, there exists an $x \in [-2, 2]$ such that $f(x) = y$. In particular, choose a value y for which $dig(y)$ is not logical. We conclude that circuit C is not a valid arbiter since its output can be forced to be non-logical way past the time it should be logical. □

Claim 11.1 and its proof are very hard to grasp at first. It seems to imply some serious flaw in our perception. Among other things, the claim implies that there does not exist a perfect judge who can determine the winner in a 100-meters dash. This statement remains true even in the presence of high speed cameras located at the finish line and even if the runners run slowly. Moreover, the judge is given several hours to decide, and if the running times of the winner and runner-up are within a second, then the judge may decide arbitrarily! Does this mean that races are pointless since, for every judge, there exist two runners whose running times are such that the judge still hangs after an hour?

Our predicament can be clarified by the following example depicted in Figure 11.5. Consider a player whose goal is to throw a ball past an obstacle so that it rolls past point P. If the ball is rolled at a speed above v', then it will pass the obstacle and then roll past point P. If the ball is thrown at a speed below v' it will not pass the obstacle. The judge is supposed to announce her decision 24 hours after the player throws the ball. The judge's decision must be either "passed" or "did not pass". Seems like an easy task. However, if the player throws the ball at speed v', then the ball reaches the tip of the obstacle and may remain there indefinitely long! If the ball remains on the obstacle's tip 24 hours past the throw, then the judge cannot announce her decision.

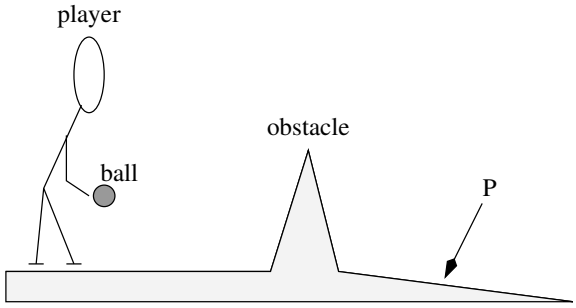

Figure 11.5: A player attempting to roll a ball so that it passes point P.

We refer to the state of the ball when resting on the tip of the obstacle as a *meta-stable* state of equilibrium. Luckily, throwing the ball so that it rests on the tip of the obstacle is a very hard task. Suppose there is some probability distribution for the speed of the ball when thrown. Unless this probability distribution is pathologic, the probability of obtaining a meta-stable state is small. Moreover, the probability of meta-stability occurring can be reduced by sharpening the tip of the obstacle or giving the arbiter more time to decide. This ability to control the probability of the event that a decision cannot be reached plays a crucial role in real life. In VLSI chips, millions of transistors transition

from one state to another millions of times per second. If even one transistor is "stuck" in a meta-stable state, then the chip might output a wrong value. By reducing the probability of meta-stability, one can estimate that meta-stability will not happen during the life-time of the chip (a lightening will hit the chip before meta-stability happens).

The consequence of this discussion is that Claim 11.1 does not make judges unemployed just as a coin toss is not likely to end up with the coin standing on its perimeter (but bear in mind that it could!). The moral of Claim 11.1 is that: (i) Certain tasks are not achievable with probability 1. If we consider the random nature of noise, we should not be surprised at all. In fact, noise could be big enough to cause the digital value of a signal to flip from zero to one. If the noise margin is large enough, then such an event is not likely to occur. However, there is always a positive probability that such an error will occur. (ii) Increasing the amount of time during which the arbiter is allowed to reach a decision (significantly) decreases the chances of meta-stability. As time progresses, even if the ball is resting on the tip of the obstacle, it is likely to fall to one of the sides. Note, however, that increasing the clock rate means that "decisions" must be made faster (i.e., within a clock period) and the chance of meta-stability increases.

Question 11.2 *Does the proof of Claim 11.1 hold only if the signals $A_i(t)$ rise "slowly"? Prove the claim with respect to non-decreasing signals $A_i(t)$ such that the length of the interval during which* $\mathrm{dig}(A_i(t))$ *is non-logical equals ε. (Figure 11.6 depicts slow and fast signals.)*

11.5 Necessity of critical segments

In this section we present a reduction from flip-flops without critical segments to arbiters. Since arbiters do not exist, the implication of this reduction is that flip-flops without critical segments do not exist as well.

We define a flip-flop without a critical segment as a flip-flop in which the setup-time and hold-time satisfy $t_{su} = t_{hold} = 0$. The functionality is defined as follows: For every i,

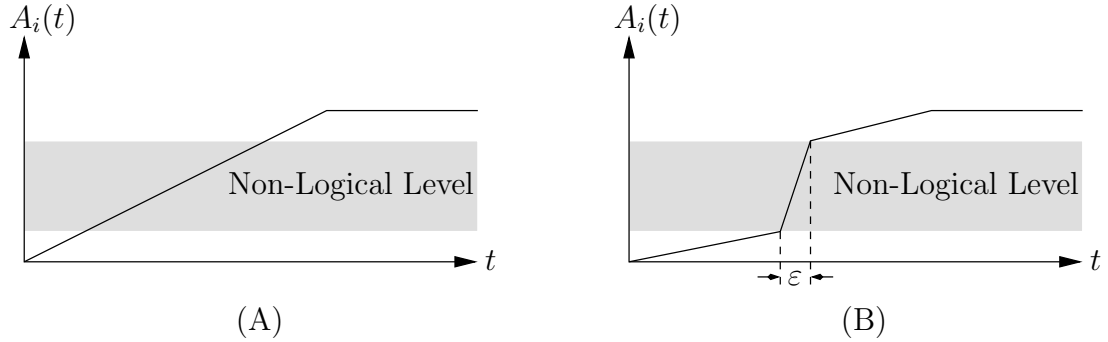

Figure 11.6: (A) Slowly rising signals $A_i(t)$ used in proof of Claim 11.1. (B) Fast signals $A_i(t)$.

$Q(t)$ is logical (either zero or one) during the interval $t \in (t_i + t_{pd}, t_{i+1} + t_{cont})$ regardless of whether $D(t_i)$ is logical. If $D(t_i)$ is logical, then $Q(t) = D(t_i)$ during the interval $t \in (t_i + t_{pd}, t_{i-1} + t_{cont})$.

The definition of a flip-flop without a critical segment is similar to an arbiter. Just as the arbiter's decision is free if a tie occurs, the flip-flop is allowed to output zero or one if $D(t_i)$ is not logical. However, the output of the flip-flip must be logical once the instability segment ends.

Consider the circuit depicted in Figure 11.7 in which the flip-flop is without a critical segment. Assume that the parameters t_{cont} and t_{pd} are significantly smaller than one time unit (e.g., at most 10^{-9} second, where one time unit equals one second). Assume also that the intervals during which the inputs $A_0(t)$ and $A_1(t)$ are non-logical are also very short (e.g., 10^{-9} second).

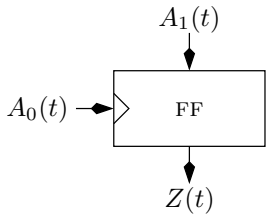

Figure 11.7: An arbiter based on a flip-flop without a critical segment.

Note that the signal $A_0(t)$ is input as a clock to the flip-flop. Our requirements from $A_0(t)$ are somewhat weaker than the requirements from a clock. Instead of periodic instantaneous transitions from zero to one and back, $A_0(t)$ is non-decreasing. Claim 11.2 assumes only one "tick of the clock", so we may regard $A_0(t)$ as a clock with a very long period. On the other hand, we do not rely on $A_0(t)$ rising slowly; the claim holds regardless of the rate of change of $A_0(t)$.

Claim 11.2 *The circuit depicted in Figure 11.7 is an arbiter.*

Proof: We need to show that: (i) if $T_1 < T_0 - 1$, then $dig(Z(t)) = 1$, for every $t \geq T_0 + t_{pd}$, and (ii) if $T_0 < T_1 - 1$, then $dig(Z(t)) = 0$, for every $t \geq T_0 + t_{pd}$. The case $T_1 - 1 \leq T_0 \leq T_1 + 1$ is solved because the flip-flop's output $Z(t)$ is always logical at time $T_0 + t_{pd}$.

If $T_1 < T_0 - 1$, then $dig(A_1(T_0)) = 1$, and hence, $dig(Z(t)) = 1$, for every $t \geq T_0 + t_{pd}$.

If $T_0 < T_1 - 1$, then we claim that $dig(A_1(T_0)) = 0$. The reason that since $T_0 < T_1$, it follows that $dig(A_1(T_0))$ must be zero. Obviously, $dig(A_1(T_0)) \neq 1$. But if it is non-logical, then the assumption on the fast transition of $dig(A_1(t))$ from zero to one implies that $dig(A_1(T_0 + 10^{-9})) = 1$, and hence, $T_1 \leq T_0 + 10^{-9}$. But then we have a contradiction to the assumption that $T_1 > T_0 + 1$. Since $dig(A_1(T_0)) = 0$, it follows that $dig(Z(t)) = 0$, for every $t \geq T_0 + t_{pd}$, as required. □

Claims 11.1 and 11.2 imply that a flip-flop without a critical segment does not exist. In other words, for every flip-flop, if there is no critical segment requirement, then there

exist input signals that can cause it to output a non-logical value outside of the instability segment.

Corollary 11.3 *There does not exist an edge-triggered flip-flop without a critical segment.*

11.6 An example

Figure 11.8 depicts a circuit consisting of two identical flip-flops and a combinational circuit C in between. A simplified timing diagram of this circuit is depicted in Figure 11.9. Instead of drawing the clock signal, only the times t_i and t_{i+1} are marked on the time axis. In addition, the critical segment and instability segment are depicted for each clock period. The digital signals $D_0(t), Q_0(t), D_1(t), Q_1(t)$ are depicted using a simplified timing diagram. In this diagram, intervals during which a digital signal is guaranteed to be stable are marked by a white block. On the other hand, intervals during which a digital signal is possibly non-logical are marked by a gray block.

In this example, we assume that the signal $D_0(t)$ is stable only during the critical segments. As a result, the signal $Q_0(t)$ is stable in the complement of the instability segments. The signal $D_1(t)$ is output by the combinational circuit C. The signal $D_1(t)$ becomes instable as soon as $Q_0(T)$ (the input of C) becomes instable. We denote the propagation delay of C by $d(C)$. The signal $D_1(t)$ stabilizes at most $d(C)$ time units after $Q_0(t)$ stabilizes. Note that we do not assume that the contamination delay of C is positive (often combinational devices do have guarantees for positive contamination delays, but we do not rely on it in this course). The signal $D_1(t)$ is stable during the critical segment C_{i+1}, and therefore, $Q_1(t)$ is stable during the complement of the instability segments.

From a functional point of view, stability of $D_0(t)$ during the critical segments implies that $D_0(t_i)$ is logical. We denote $D_0(t_i)$ by $\sigma \in \{0, 1\}$. During the interval $[t_i + t_{pd}, t_{i+1} + t_{cont}]$ the flip-flop's output $Q_0(t)$ equals σ. The circuit C outputs a logical value $\sigma' \in \{0, 1\}$ which is a Boolean function of σ. The value σ' is output by C during the interval $[t_i + t_{pd} + d(C), t_{i+1} + t_{cont}]$. It follows that $Q_1(t)$ equals σ' during the interval $[t_{i+1} + t_{pd}, t_{i+2} + t_{cont}]$.

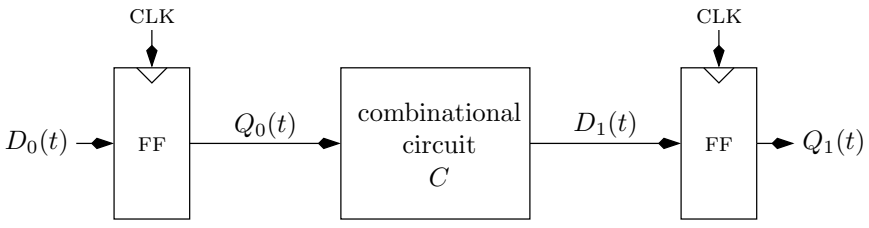

Figure 11.8: A circuit with two identical flip-flips and a combinational circuit in between.

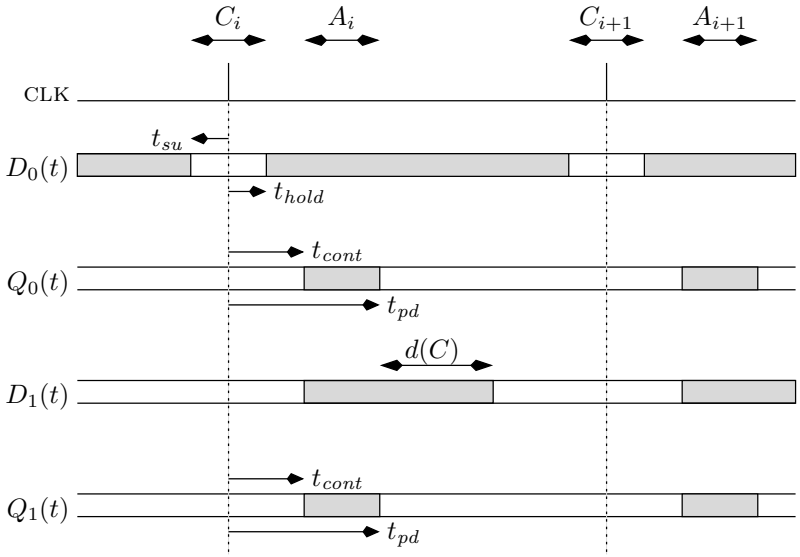

Figure 11.9: A simplified timing diagram of circuit depicted in Fig. 11.8. Gray areas denote potential instability of a signal, and white areas denote guaranteed stability of a signal.

11.6.1 Non-empty intersection of C_i and A_i

The above analysis fails if the critical segment C_i and the instability segment intersect, namely,

$$C_i \cap A_i \neq \emptyset.$$

This could happen, if $t_{hold} > t_{cont}$ (in contradiction to Definition 11.2).

We now explain why this can cause the circuit to fail (see Figure 11.10). The period during which $D_1(t)$ is guaranteed to be stable is $[t_i + t_{pd} + d(C), t_{i+1} + t_{cont}]$. However, if $t_{cont} < t_{hold}$, then $D_1(t)$ is not guaranteed to be stable during the critical segment C_{i+1}. This is a violation of the assumptions we require in order to guarantee correct functionality.

In many flip-flop implementations it so happens that $t_{hold} > t_{cont}$. How are such flip-flops used? The answer is that one needs to rely on the contamination delay of the combinational circuit C. Let $cont(C)$ denote the contamination delay of C. The interval during which $D_1(t)$ is guaranteed to be stable is

$$[t_i + t_{pd} + d(C), t_{i+1} + t_{cont} + cont(C)].$$

If $t_{cont} + cont(C) > t_{hold}$, then the signal $D_1(t)$ is stable during the critical segment C_{i+1}, and correct functionality is obtained.

In this course we simplify by adopting the more restrictive assumption that the contamination delay of every combinational circuit is zero. This means that we need to be more restrictive with respect to flip-flops and require that the critical segment and the instability segments are disjoint. Note, however, that even if the contamination delay of C is positive (although we assumed it is zero), then our analysis is still valid. Hence, not

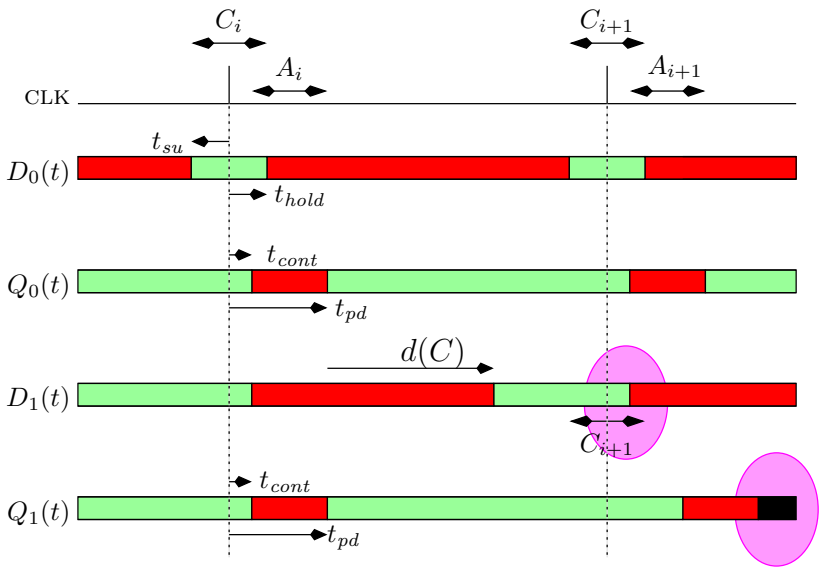

Figure 11.10: The simplified timing diagram in the case that $A_i \cap C_i \neq \emptyset$.

relying on a positive contamination delay of combinational circuits does not introduce errors even if the contamination delay is positive.

Question 11.3 *Assume that we have an edge-triggered flip-flop* FF *in which* $t_{\text{hold}} > t_{\text{cont}}$. *Suppose that we have an inverter with a contamination delay* $\text{cont}(\text{INV}) > 0$. *Suggest how to design an edge-triggered flip-flop* FF′ *that satisfies* $t_{\text{hold}}(\text{FF}') < t_{\text{cont}}(\text{FF}')$. *What are the parameters of* FF′*?*

11.7 Other types of memory devices

Edge triggered flip-flops are not the only memory device that exist. We briefly overview some of these devices.

11.7.1 D-Latch

A D-latch, like an edge-triggered flip-flop, is characterized by two parameters t_{su} and t_{hold}. However, the critical segment is defined with respect to the falling edge of the clock. Let t'_i denote the time of the falling edge of the clock during the ith clock cycle. The critical segment of a D-latch is defined to be $[t'_i - t_{su}, t'_i + t_{hold}]$. In addition, the D-latch is characterized by a combinational delay d. The functionality of a D-latch is defined as follows.

1. During the interval $[t_i + d, t'_i)$, the output $Q(t)$ satisfies: $Q(t) = D(t)$, provided that $D(t)$ is stable during the interval $[t - d, t]$. We say that the D-latch is *transparent* during the interval $[t_i + d, t'_i)$.

2. During the interval $(t'_i + t_{hold}, t_{i+1})$, if $D(t)$ is stable during the critical segment $[t'_i - t_{su}, t'_i + t_{hold}]$, then $Q(t) = D(t'_i)$. We say that the D-latch is opaque during the interval $(t'_i + t_{hold}, t_{i+1})$.

D-latches are very important devices. They are cheaper than flip-flops, and in fact, D-latches are the building blocks of flip-flops. Moreover, using D-latches wisely leads to faster designs. However, designs based on D-latches require multiple clock phases (or at least a clock CLK and its negation $\overline{\text{CLK}}$). Although timing with multiple clock phases is an important and interesting topic, we do not deal with it in this course.

11.7.2 Clock enabled flip-flips

We use the terminology and notation of an edge-triggered flip-flop in the definition of a clock enabled flip-flop.

Definition 11.4 *A clock enabled flip-flop is defined as follows.*

Inputs: *Digital signals* $D(t), \text{CE}(t)$ *and a clock* CLK.

Output: *A digital signal* $Q(t)$.

Functionality: *If* $D(t)$ *and* $\text{CE}(t)$ *are stable during the critical segment* C_i*, then for every* $t \in (t_i + t_{\text{pd}}, t_{i+1} + t_{\text{cont}})$

$$Q(t) = \begin{cases} D(t_i) & \text{if } \text{CE}(t_i) = 1 \\ Q(t_i) & \text{if } \text{CE}(t_i) = 0. \end{cases}$$

We refer to the input signal $\text{CE}(t)$ as the clock-enable signal. Note that the input $\text{CE}(t)$ indicates whether the flip-flop samples the input $D(t)$ or maintains its previous value.

Part (A) of Figure 11.11 depicts a successful implementation of a clock enabled flip-flop. This implementation uses a MUX and an edge-triggered flip-flop. Part (B) of Figure 11.11 depicts a weak implementation of a clock enabled flip-flop.

The main weakness of the design depicted in part (B) is that the output of the AND-gate is not a clock signal. For example, the output of the AND-gate is allowed to fluctuate when $\text{CE}(t)$ is not logical. Such fluctuations (called *glitches*) can cause the flip-flop to sample the input when not needed. In addition, the transitions of the output of the AND-gate might be slow and require increasing the hold time. Moreover, in some technologies, the flip-flop does not retain the stored bit forever. For example, consider the case in which the stored value is retained only for 2-3 clock cycles. In such a case, if the clock-enable signal is low for a long period then the flip-flop's output may become non-logical.

Question 11.4 *Compute the parameters of the clock-enabled flip-flop depicted in part (A) of Figure 11.11 in terms of the parameters of the edge-triggered flip-flop and the* MUX.

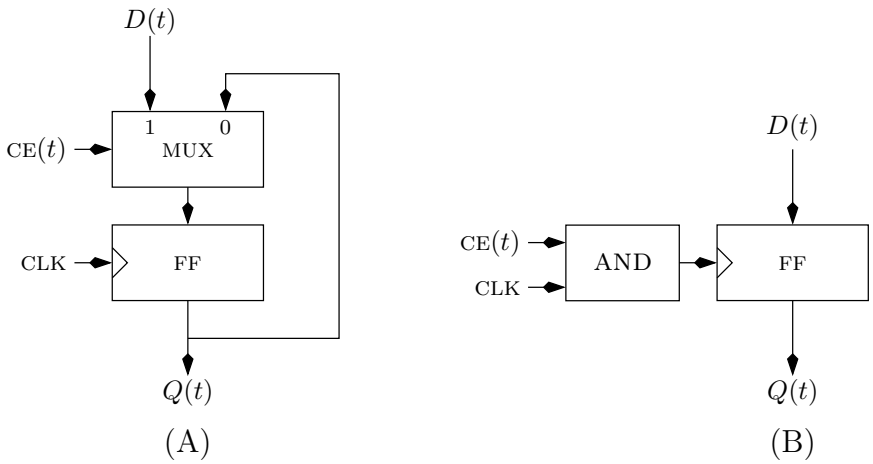

Figure 11.11: (A) a successful implementation of a clock enabled flip-flop. (B) A wrong design.

11.8 Summary

In this chapter we presented memory devices called *flip-flops*. We consider using flip-flops in the presence of a *clock signal*. The clock signal causes the flip-flop to sample the value of the input towards the end of a clock cycle and output the sampled value during the next clock cycle. Flip-flops play a crucial role in bounding the segments of time during which signals may be instable.

In a sense, flip-flops and combinational circuits have opposite roles. Combinational circuits compute interesting Boolean functions but increase uncertainty (namely, lengthen segments of time during which signals may be instable). Flip-flops, one the other hand, output the same value that is fed as input but they limit uncertainty.

We considered a task called *arbitration*. We proved that no circuit can implement an arbiter. We then proved that a flip-flop with an empty critical segment can be used to build an arbiter. This proves that a flip-flop must have a non-empty critical segment.