Sequential Circuits: Timing

Digital Design and Computer Architecture
Mohammad Sadrosadati
Frank K. Gürkaynak

http://safari.ethz.ch/ddca
What will we learn?

- Timing of Sequential Logic
  - Setup time
  - Hold time

- Clock frequency

- Metastability and prevention
Timing

- Flip-flop samples D at clock edge
- D must be stable when it is sampled
- Similar to a photograph, D must be stable around the clock edge
- If D is changing when it is sampled, metastability can occur
  - Recall that a flip-flop copies the input D to the output Q on the rising edge of the clock. This process is called sampling D on the clock edge. If D is stable at either 0 or 1 when the clock rises, this behavior is clearly defined. But what happens if D is changing at the same time the clock rises?
Input Timing Constraints

- **Setup time**: \( t_{\text{setup}} = \) time before the clock edge that data must be stable (i.e. not changing)
- **Hold time**: \( t_{\text{hold}} = \) time after the clock edge that data must be stable
- **Aperture time**: \( t_a = \) time around clock edge that data must be stable \( (t_a = t_{\text{setup}} + t_{\text{hold}}) \)
Output Timing Constraints

- **Propagation delay**: $t_{pcq} = \text{time after clock edge that the output Q is guaranteed to be stable (i.e., to stop changing)}$

- **Contamination delay**: $t_{ccq} = \text{time after clock edge that Q might be unstable (i.e., start changing)}$
Dynamic Discipline

- The input to a synchronous sequential circuit must be stable during the aperture (setup and hold) time around the clock edge.

- Specifically, the input must be stable
  - at least $t_{\text{setup}}$ before the clock edge
  - at least until $t_{\text{hold}}$ after the clock edge
Dynamic Discipline

- The delay between registers has a minimum and maximum delay, dependent on the delays of the circuit elements.
Setup Time Constraint

- The clock period or cycle time, $T_c$, is the time between rising edges of a repetitive clock signal. Its reciprocal, $f_c = 1/T_c$, is the clock frequency.

- All else being the same, increasing the clock frequency increases the work that a digital system can accomplish per unit time.

- Frequency is measured in units of Hertz (Hz), or cycles per second:
  - 1 megahertz (MHz) \(10^6\) Hz
  - 1 gigahertz (GHz) \(10^9\) Hz.
Setup Time Constraint

- The setup time constraint depends on the maximum delay from register R1 through the combinational logic.

- The input to register R2 must be stable at least $t_{\text{setup}}$ before the clock edge.

![Diagram showing setup time constraint with variables $T_c$, $t_{\text{pcq}}$, $t_{\text{pd}}$, and $t_{\text{setup}}$.]
Setup Time Constraint

- The setup time constraint depends on the maximum delay from register R1 through the combinational logic.

- The input to register R2 must be stable at least $t_{\text{setup}}$ before the clock edge.

\[ T_c \geq t_{\text{pcq}} + t_{\text{pd}} + t_{\text{setup}} \]

\[ t_{\text{pd}} \leq \]
Setup Time Constraint

- The setup time constraint depends on the maximum delay from register R1 through the combinational logic.

- The input to register R2 must be stable at least $t_{\text{setup}}$ before the clock edge.

\[
T_c \geq t_{\text{pcq}} + t_{\text{pd}} + t_{\text{setup}}
\]

\[
t_{\text{pd}} \leq T_c - (t_{\text{pcq}} + t_{\text{setup}})
\]
Hold Time Constraint

- The hold time constraint depends on the minimum delay from register R1 through the combinational logic.

- The input to register R2 must be stable for at least $t_{\text{hold}}$ after the clock edge.
Hold Time Constraint

- The hold time constraint depends on the minimum delay from register R1 through the combinational logic.

- The input to register R2 must be stable for at least $t_{\text{hold}}$ after the clock edge.

$t_{\text{hold}} < t_{\text{ccq}} + t_{\text{cd}}$

$t_{\text{cd}} >$
Hold Time Constraint

- The hold time constraint depends on the minimum delay from register R1 through the combinational logic.

- The input to register R2 must be stable for at least $t_{\text{hold}}$ after the clock edge.

\[
t_{\text{hold}} < t_{\text{ccq}} + t_{\text{cd}}
\]

\[
t_{\text{cd}} > t_{\text{hold}} - t_{\text{ccq}}
\]
Timing Analysis

Timing Characteristics

- $t_{ccq} = 30 \text{ ps}$
- $t_{pcq} = 50 \text{ ps}$
- $t_{setup} = 60 \text{ ps}$
- $t_{hold} = 70 \text{ ps}$

- $t_{pd} = 35 \text{ ps}$
- $t_{cd} = 25 \text{ ps}$ per gate

Setup time constraint:

$$T_c \geq \frac{1}{f_c}$$

Hold time constraint:

$$t_{ccq} + t_{cd} > t_{hold}?$$
Timing Analysis

CLK

CLK

A

B

C

D

X'

Y'

X

Y

$t_{pd} = 3 \times 35 \text{ ps} = 105 \text{ ps}$

$t_{cd} = 25 \text{ ps}$

Setup time constraint:

$T_c \geq (50 + 105 + 60) \text{ ps} = 215 \text{ ps}$

$f_c = 1/T_c = 4.65 \text{ GHz}$

Timing Characteristics

- $t_{ccq} = 30 \text{ ps}$
- $t_{pcq} = 50 \text{ ps}$
- $t_{setup} = 60 \text{ ps}$
- $t_{hold} = 70 \text{ ps}$

Hold time constraint:

$t_{ccq} + t_{cd} > t_{hold}$?

$(30 + 25) \text{ ps} > 70 \text{ ps}$? **No!**
Fixing Hold Time Violation

Add buffers to the short paths:

Timing Characteristics

\[ t_{ccq} = 30 \text{ ps} \]
\[ t_{pcq} = 50 \text{ ps} \]
\[ t_{setup} = 60 \text{ ps} \]
\[ t_{hold} = 70 \text{ ps} \]

\[ t_{pd} = 35 \text{ ps} \]
\[ t_{cd} = 25 \text{ ps} \]

Setup time constraint:
\[ T_c \geq \]
\[ f_c = \]

Hold time constraint:
\[ t_{ccq} + t_{cd} > t_{hold} ? \]
Fixing Hold Time Violation

Add buffers to the short paths:

- $t_{pd} = 3 \times 35 \text{ ps} = 105 \text{ ps}$
- $t_{cd} = 2 \times 25 \text{ ps} = 50 \text{ ps}$

Setup time constraint:

$T_c \geq (50 + 105 + 60) \text{ ps} = 215 \text{ ps}$

$T_c = \frac{1}{f_c} = 4.65 \text{ GHz}$

Hold time constraint:

$t_{ccq} + t_{cd} > t_{hold} \text{ ?}$

$(30 + 50) \text{ ps} > 70 \text{ ps} \text{ ? Yes!}$

Timing Characteristics

- $t_{ccq} = 30 \text{ ps}$
- $t_{pcq} = 50 \text{ ps}$
- $t_{setup} = 60 \text{ ps}$
- $t_{hold} = 70 \text{ ps}$
- $t_{pd} = 35 \text{ ps}$ per gate
- $t_{cd} = 25 \text{ ps}$ per gate
Clock Skew

- The clock doesn’t arrive at all registers at the same time
- Skew is the difference between two clock edges
- Examine the worst case to guarantee that the dynamic discipline is not violated for any register – many registers in a system!
Setup Time Constraint with Clock Skew

- In the worst case, the CLK2 is earlier than CLK1

\[ T_c \geq t_{\text{pd}} + t_{\text{pcq}} + t_{\text{setup}} + t_{\text{skew}} \]
Setup Time Constraint with Clock Skew

- In the worst case, the CLK2 is *earlier* than CLK1

\[
T_c \geq t_{pcq} + t_{pd} + t_{setup} + t_{skew}
\]

\[
t_{pd} \leq \]

![Diagram of setup time constraint with clock skew]
Setup Time Constraint with Clock Skew

In the worst case, the CLK2 is earlier than CLK1

\[ T_c \geq t_{pcq} + t_{pd} + t_{setup} + t_{skew} \]

\[ t_{pd} \leq T_c - (t_{pcq} + t_{setup} + t_{skew}) \]
Hold Time Constraint with Clock Skew

- In the worst case, CLK2 is later than CLK1

$$t_{\text{hold}} + t_{\text{skew}} \leq \text{constraint}$$
Hold Time Constraint with Clock Skew

- In the worst case, CLK2 is later than CLK1

\[ t_{\text{hold}} + t_{\text{skew}} \leq t_{\text{ccq}} + t_{\text{cd}} \]

\[ t_{\text{cd}} \geq \]
Hold Time Constraint with Clock Skew

- In the worst case, CLK2 is later than CLK1

\[ t_{\text{hold}} + t_{\text{skew}} \leq t_{\text{ccq}} + t_{\text{cd}} \]
\[ t_{\text{cd}} \geq t_{\text{hold}} + t_{\text{skew}} - t_{\text{ccq}} \]
Asynchronous (for example, user) inputs might violate the dynamic discipline.
Metastability

- Any bi-stable device has two stable states and a metastable state between them

- A flip-flop has two stable states (1 and 0) and one metastable state

- If a flip-flop lands in the metastable state, it could stay there for an undetermined amount of time
**Metastability**

- $T_0/T_c$ describes the probability that the input changes at a bad time, i.e., during the aperture time

$$P(t_{\text{res}} > t) = (T_0/T_c) \ e^{-t/\tau}$$

- $\tau$ is a time constant indicating how fast the flip-flop moves away from the metastable state; it is related to the delay through the cross-coupled gates in the flip-flop

- In short, if a flip-flop samples a metastable input, if you wait long enough (t), the output will have resolved to 1 or 0 with high probability.
Synchronizers

- Asynchronous inputs (D) are inevitable (user interfaces, systems with different clocks interacting, etc.).

- The goal of a synchronizer is to make the probability of failure (the output Q still being metastable) low.

- A synchronizer cannot make the probability of failure 0.
Synchronizer Internals

- A synchronizer can be built with two back-to-back flip-flops.
- Suppose the input D is changing when it is sampled by F1.
- Internal signal D2 has \((T_c - t_{\text{setup}})\) time to resolve a 1 or 0.
For each sample, the probability of failure of this synchronizer is:

\[ P(\text{failure}) = \frac{T_0}{T_c} e^{-\frac{T_c - t_{\text{setup}}}{\tau}} \]

![Diagram of a synchronizer with labels for t_{\text{res}}, t_{\text{setup}}, t_{\text{pcq}}, T_c, F1, F2, D, D2, Q, and CLK signals.]}
Synchronizer Mean Time Before Failure

- If the asynchronous input changes once per second, the probability of failure per second of the synchronizer is simply $P(failure)$. 

- In general, if the input changes $N$ times per second, the probability of failure per second of the synchronizer is:

$$P(failure)/sec = N \frac{T_0}{T_c} e^{\frac{T_c - t_{\text{setup}}}{\tau}}$$

- Thus, the synchronizer fails, on average, $1/[P(failure)/second]$ This is called the mean time between failures, MTBF:

$$MTBF = \frac{1}{P(failure)/sec} = \frac{T_c - t_{\text{setup}}}{\tau}$$

$$= \frac{T_c e^{\frac{T_c - t_{\text{setup}}}{\tau}}}{NT_0}$$


Example Synchronizer

Suppose:  \( T_c = \frac{1}{500 \text{ MHz}} = 2 \text{ ns} \) \hspace{1cm} \( \tau = 200 \text{ ps} \)
\( T_0 = 150 \text{ ps} \) \hspace{1cm} \( t_{\text{setup}} = 100 \text{ ps} \)
\( N = 10 \text{ events per second} \)

What is the probability of failure? MTBF?

\[
P(\text{failure}) = \]
\[
P(\text{failure})/\text{second} = \]
\[
\text{MTBF} = \]
Suppose: \( T_c = \frac{1}{500} \text{ MHz} = 2 \text{ ns} \)  
\( \tau = 200 \text{ ps} \)  
\( T_0 = 150 \text{ ps} \)  
\( t_{\text{setup}} = 100 \text{ ps} \)  
\( N = 10 \text{ events per second} \)

What is the probability of failure? MTBF?

\[
P(\text{failure}) = \left(\frac{150 \text{ ps}}{2 \text{ ns}}\right) e^{-\frac{1.9 \text{ ns}}{200 \text{ ps}}} \\
= 5.6 \times 10^{-6}
\]

\[
P(\text{failure})/\text{second} = 10 \times (5.6 \times 10^{-6}) \\
= 5.6 \times 10^{-5} / \text{second}
\]

\[
\text{MTBF} = \frac{1}{[P(\text{failure})/\text{second}]} \approx 5 \text{ hours}
\]
What did we learn?

- **Timing constraints for sequential circuits**
  - *Setup time*, time needed for the inputs to be present before clock
  - *Hold time*, time needed after the clock where inputs should not change
  - *Aperture time*, time around clock event where inputs should stay stable

- **Problems related to clock skew**

- **Metastability and Synchronization**