ELEX 2117 : Digital Techniques 2
2020 Fall Term

Static Timing Analysis

This lecture describes how static timing analysis is used to ensure timing constraints for a digital design are met. After this lecture you should be able to be able to apply the terms defined in this lecture and do calculations involving clock rate, propagation delays and setup/hold time requirements.

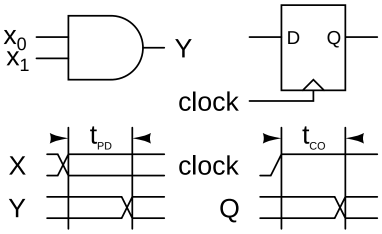
Introduction

Timing constraints are requirements such as the clock rate or setup and hold time requirements. Meeting these constraints is often as difficult as ensuring a design is logically correct.

Reliable operation requires that the designer correctly specify the timing constraints and modify the design until they are met.

Propagation Delays

Propagation delay, t_{PD}, is the delay from a change at an input to the change at the output of a combinational logic circuit. The clock-to-output delay, t_{CO}, is the delay from the rising edge at a flip-flop clock input to the change at the Q output.

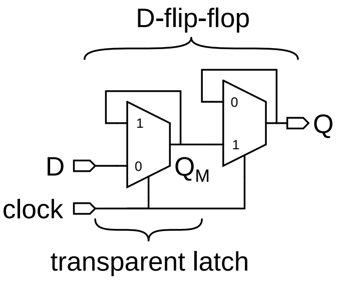


These delays are caused by the time required to charge the parasitic capacitances of transistors and interconnects.

In the timing diagrams above the parallel lines are times during which the signal is at a high or low level. The lines cross at times where the signal may change.

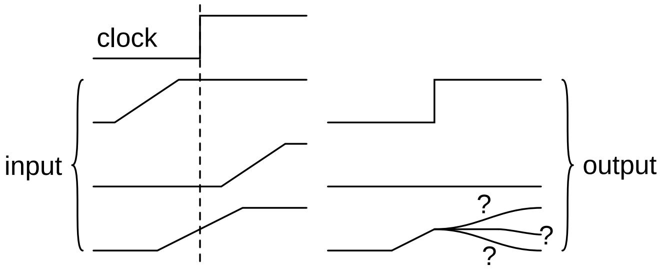
Metastability, Setup and Hold Times

Consider the following implementation of an edge-triggered D flip-flop:



When the clock input is 0, the output of the first multiplexer follows the input – the latch is "transparent". When the clock input is 1, the output level is fed back to the input and held at that level[1].

If the latch output is at the logic threshold voltage when the clock changes from 0 to 1 then the multiplexer might not be able to decide whether to feed back a 0 or 1. The multiplexer output could remain at an invalid level for much longer than t_{CO}. This behaviour is called "metastability" and can result in unreliable circuit operation.



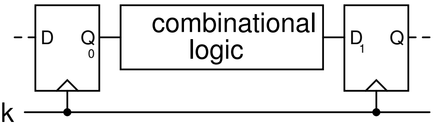
To avoid metastability we must ensure the voltage at the latch input is at valid level long enough to drive the latch output to a valid voltage level. The time required for this is called the "setup" time, t_{SU}.

The input level must also be held at the correct level until the multiplexer has switched off. This is typically a much shorter time – often zero – and is called the "hold" time, t_{H}.

[1] This is a "master-slave" flip-flop. The second, "slave," latch holds the previously latched value when the clock is 0

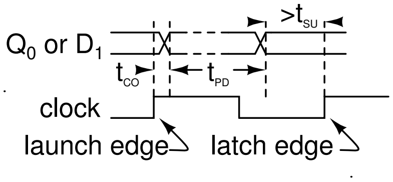
Synchronous Design

To avoid metastability almost all digital circuits are "synchronous." These circuits are composed of edge-triggered flip-flops with combinational logic between their outputs and inputs:



By ensuring the propagation delays through the combinational logic will meet the setup and hold requirements we can avoid metastable behaviour.

The timing diagram below shows the relationship between the clock edges and the valid times at the inputs and outputs of each flip-flop:



Q changes t_{CO} after the rising clock edge. t_{PD} later, the input at the D input of the right flip-flop will have a valid (and correct) logic level. This must happen t_{SU} at the latest before the next rising edge of the clock. This level must also be held for at least t_H before it changes.

The diagram above identifies two clock edges, the "launch" and "latch" edges. In this example the edges are separated by the clock period. However, the clocks may arrive at different times due to different interconnect delays. This is known as "clock skew." It's also possible that the two clocks have different frequencies or latch on the falling edge.

These setup/hold times are often called a "library" or "micro" times to distinguish them from an IC's IO setup and hold times.

Static Timing Analysis

In most cases the timing requirement that is most difficult to meet is the minimum setup time.

From the timing diagram above we can write an expression for the guaranteed minimum available setup time:

$$t_{SU} = T_{clock} - t_{CO} - t_{PD}$$

where t_{SU} is the guaranteed minimum set up time, T_{clock} is the clock period, t_{CO} is the maximum clock to output delay of the launching flip-flop and t_{PD} is the maximum propagation delay through the combinational logic. Both t_{CO} and t_{PD} are the maximum delays specified by the manufacturer.

The amount by which the minimum available setup time exceeds the minimum required setup time is known as the "slack":

$$\text{slack} = t_{SU}\ (\text{available}) - t_{SU}\ (\text{required})$$

If the slack is positive then the setup time requirements are met, otherwise they are not and the circuit may not operate reliably.

Exercise 1: For a particular circuit f_{clock} is 50 MHz, t_{CO} is 2 ns (maximum), the worst-case (maximum) t_{PD} in a circuit is 15 ns and the minimum setup time requirement is 5 ns. What is the setup time slack? Will this circuit operate reliably? If not, what it the maximum clock frequency at which it will?

Closing Timing

"Closing" timing is the process of iterating a design until all paths have positive slack. There are various options when a design does not meet its timing requirements:

- ask the EDA software to spend more time (effort) optimizing the layout and routing
- use a larger or faster device or process – this makes it easier to optimize PAR
- modify the design to speed up critical timing paths. This might mean having more logic in parallel or dividing up the computation into more clock cycles.
- relax the design constraints (e.g. reduce the clock rate)

the choice will depend on the project requirements and available resources.

PVT and Corners

The propagation delays on one die will depend on the temperate and voltage. There will also be differences between die due to process differences. STA should be repeated using delays for the expected "PVT" (Process, Voltage, Temperature) extremes. The PVT combination that results in the maximum or minimum delays is called a "corner."

Asynchronous Clocks and Inputs

If all clocks are derived from the same source clock (e.g. through clock division or using a PLL) the time relationships between clocks remains constant and it's possible to verify that timing constraints will be met.

However, if two clocks are physically independent then this is not possible – the clock edges will drift relative to each other and the setup and hold timing requirement of flip-flops with asynchronous inputs are bound to be violated at some point. Even though it's not possible to do timing analysis on asynchronous signals, it is possible to estimate the mean time between failure (MTBF) due to metastable events when signals cross clock "domains."

Timing Simulations

A timing-annotated netlist can be used by a simulator to run simulations that take into account delays. During the simulation the simulator can check that the setup and hold requirements of each flip-flop are met.

The advantage of this "dynamic" timing analysis is that the verification results are independent of, and can serve as a check on, user-provided timing constraints. The disadvantage is that the simulation may not cover all possible events. Timing simulations can be time-consuming for large designs and are primarily used for ASIC "sign-off."