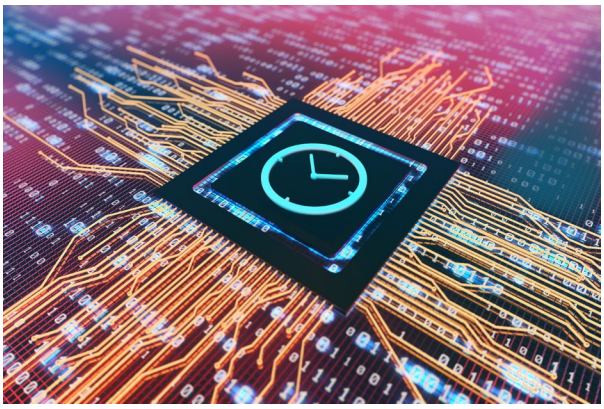
TIME AFTER TIME:
A GUIDE TO
STATIC TIMING ANALYSIS

EMA Design Automation®

INTRODUCTION:
TIMING IS KEY

If you've read our High Speed Design ebook, then you will know that getting the timing correct within a given design is an invaluable step in the process of making a working product.

Many electrical engineers have learned to check timing at certain points of friction, and there are some great basic tools and rules of thumb one can follow to get started. However, it is only through digging deeper and having a solid understanding of static timing analysis and its challenges will you ensure your project runs smoothly.



First, a recap: what is timing?

Static timing analysis (or STA) is defined as a method of validating the timing performance of a design by checking all possible paths for timing violations under worst-case conditions. To put it in less technical terms: it is a way to check all the necessary signals in each circuit are arriving at their destinations within the required time frames. The achievement of this task is called *timing closure*. ***Note:*** this kind of verification is done statically, meaning no simulation is required, and as such these are checks for timing only and not functional correctness.

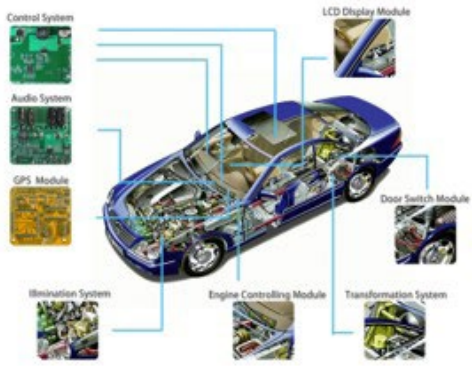
Functional simulations will check for functional correctness of a circuit and are typically ran without regard to timing. You can, however, run a "timed" functional simulation— but this does slow down the process. In "static" timing analysis, you simply add the component and net delays from your library, come up with the path delays and compare them to the timing specifications within your constraints. If a violation occurs, your timing has failed. In this way, the STA check is a faster and more straightforward course of action. The goal is to resolve all violations and achieve system synergy.

PUTTING IT INTO CONTEXT

WHY TIMING MATTERS IRL

Timing system solutions are utilized everywhere, at all interaction levels.

These interactions come in macro form (airlines, trains, and traffic control) and in micro systems of memory, parallel/serial buses, or simple control signals. Some timing tolerances are much more critical than others, such as in class A use-cases in medical, military, or automotive fields. For example, the difference of milliseconds in the auto-braking system of a car can make the difference between a major and a minor car crash, and lives can literally depend on the technology functioning as it should. Likewise, MRI machines use FPGA for digital signal processing, and ensuring precise results is important to the patient's health and treatment.



These systems come with their own unique challenges.

Large and medium systems depend on a growing system of functionality and activity, and much coordination is needed to keep things running smoothly. Larger and medium systems account for things like airport runway scheduling, traffic light patterns, and accurate data transfer within the medical or military fields. These are systems in which public and product safety is dependent on on-time delivery and require precise debugging analysis on a regular basis.

Take note of how the smaller systems influence the larger ones. Shrinking timing and noise margins in more advanced, lower power devices are necessary to push the expediency farther up the chain. Within networking systems, designers are looking to the next-generation memory interfaces to resolve needs for increased efficiency in data exchange and storage. These advances are key in realizing the future of digital wireless communication and its infrastructure. Ensuring accurate data transfer with ever-increasing interface complexity is a difficult balancing act in any system, board, or circuit design, but one that is important to project flow.

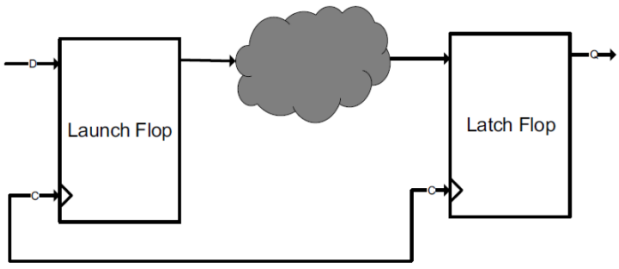
THE PATH NOT TAKEN

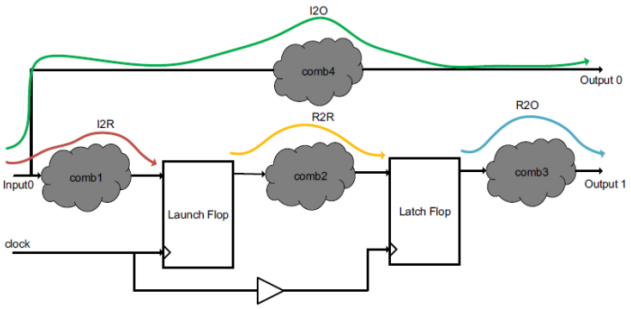
UNDERSTANDING YOUR TIMING OPTIONS

Static timing analysis can be performed on both sequential and combinatorial parts of a given design.

In any sequential design path, there is always one *launch flip-flop* to drive the data, and one *latch flip-flop* to capture it. In other words, the essential places are where the signal is coming *from*, and where it is going *to*. The key lies in how long it takes to get from one to the other.

Furthermore, a circuit also has *setup* and *hold* requirements (aka constraints). These constraints dictate the *slowest* and *fastest* possible paths a signal may take to still be within a timing tolerance. If a design fulfills all required constraints, you have achieved timing closure. Performing STA will prove or disprove the setup and hold constraints by analyzing all the possible timing paths in the design and providing the data necessary to view your options for a successful finished product.





Digital designs can be divided into four categories of paths, and timing analysis is run on each:

1. Register to Register (R2R)
2. Input to Register (I2R)
3. Register to Output (R2O)
4. Input to Output (I2O)

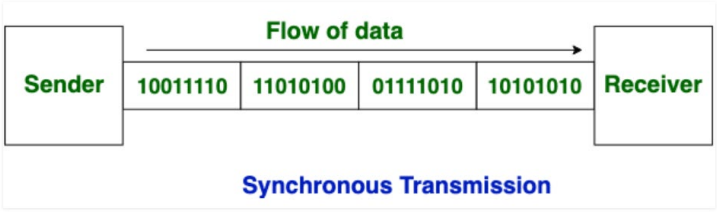
Each path category is analyzed separately by defining start and end points in that path. The R2R category is the most common and often becomes the basis for the others. The type of design you desire to make is what will determine what course of action is needed to move forward. These options tend to delve into the rabbit hole specific to IC design, so you will need to do your research on what will work best for your project.

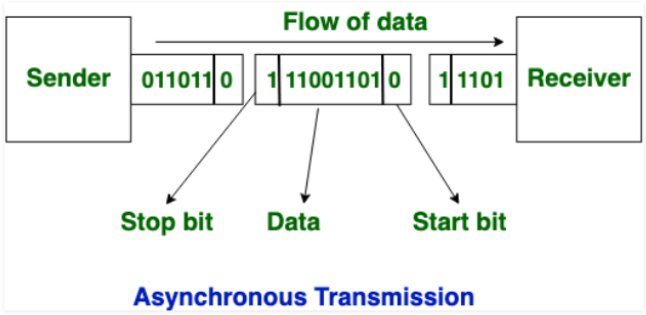
TIMING SYNCHRONIZATION

WHAT MAKES IT TICK?

Exact signal transmission is the crux of how timing works in a circuit.

Much like Morse code translates long and short blips into letters, signal transmission translates binary code into data that is transferred from one end to the other. *Synchronous transmission* and *asynchronous transmission* are two categories in which the data can be transferred.





In synchronous transmission, data is sent in blocks, with no gaps present between the data. The transmission utilizes signal delay coupled with clocking signal changes to transfer the data. This method is more efficient and reliable than asynchronous transmission in the transfer of large amounts of data without corruption or interference. You may find this method used in IC design, FPGA design, and memory interfaces.

For asynchronous transmission, the data is sent in the form of a byte or character where start and stop bits are added into the data. This form of transmission is slower, based purely on signal delay, and does not require synchronization. Asynchronous timing is used in system-level interfaces and general PCB designs that do not require tight levels of signal constraints.

Once the signal transmission is determined, STA analysis is run after initial design. Then, circuit specifications are laid out and register-transfer levels (RTL) are captured and converted to logic gates. STA is then performed once more to determine which components to use. Once the components are placed, STA is run yet again in final verification checks before the design is signed off and sent on to layout. While it may seem like many rounds of analysis, it is important to catch any possible failures before it becomes too late.

DYNAMIC VS STATIC TIMING

WHAT'S THE DIFFERENCE?

While it would be a disservice if we did not compare static and dynamic timing analysis, it is true that STA is the more efficient method.

The most notable difference between the two is dynamic timing analysis *verifies functionality* of the design whereas static timing analysis *validates timing performance.* Functionally, dynamic timing analysis applies input vectors and checks for the correct output vectors, whereas static timing analysis checks static delay requirements of the circuit without any input or output vectors.

Dynamic Timing Analysis	Static Timing Analysis
Simulation-based	Formula-based
Highly accurate	More pessimistic
Slow run time	Very fast
Not possible for whole chip	Most popular for sign-off checks

Note: DTA and STA are not full alternatives of each other, but there are instances where one may have a greater benefit.

The overall quality of DTA increases with more input test vectors, but it also increases simulation time. STA is best suited for designs with clocks across multiple domains, and is usually preformed on larger circuits, such as SOC (system-on-chip), ASIC design or FPGA, which are usually very large circuits.

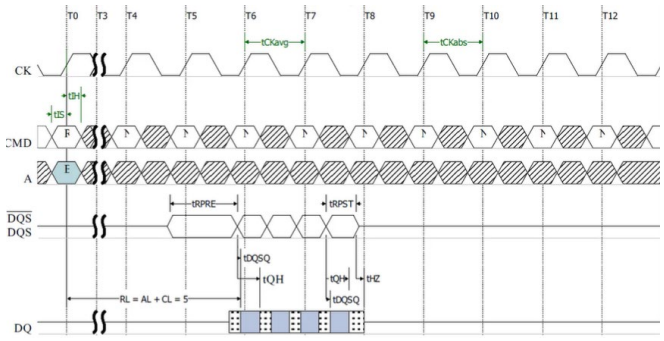
DTA can be carried out using SPICE or functional simulators. Such simulation is calculated based on mathematical equations to represent the electrical properties of a custom design. SPICE characterized data is tabulated in technology libraries, which become basic delay information for the STA. This data is the most accurate compared to any other form of simulation, but takes the longest to compile. For the most part, STA will do the job, as it is faster and more applicable in the design flow and is simple to apply.

STATIC TIMING ANALYSIS

BECAUSE TIME WAITS FOR NO ONE

If static timing analysis is the best course of action for your project, then it is best to use a graphical tool to render it.

STA is conservative in that it gives leeway in both long and short paths, guaranteeing the design will function at least as predicted and not suffer from hold-time violations. A graphical tool is made to be quick, easy, and gives great support for design re-use and flexibility to run many variations in a short amount of time. Within most tools, STA algorithms have become mature enough in recent years to address a plethora of key timing issues, though not all will provide you with a big enough picture of the most useful data, so you may need to run multiple checks for specific problems.



In setting up your STA, there are some key actions to dial in your results:

1. Start constraining your design by defining the clock(s). This will give you a starting tolerance range to work within.

2. Always define the constraints for all primary inputs and outputs.

3. Constrain all the paths in the design. If the constraints file is not complete, STA will not be complete. Remember, static timing analysis works with a ‘garbage in, garbage out’ principal, where the results are only as accurate as the data you give it.

4. If a path is not critical, you can define exceptions for them (false path, multi cycle path, etc.)

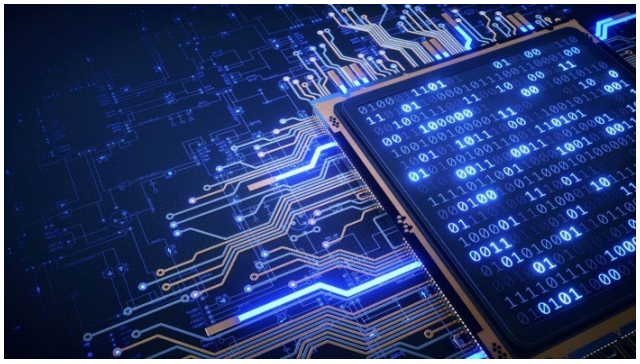
While the specifics of an issue vary from project to project, these tips will get you on the right path in understanding the full scope of timing. It is often critical to work out any bugs at this stage, long before the design is finalized and passed onto the layout phase.

TIMINGDESIGNER

SAVING TIME, EVERY TIME

It is often crucial for a graphical tool to give you the best and most reliable data when running static timing analysis.

When timing is not met, you will find yourself with a myriad of performance and delivery issues. TimingDesigner® can evaluate comprehensive sets of timing alternatives and provide direction to the most complex challenges, enabling designers to manage and monitor timing margins through the design process. This is why TimingDesigner® is the industry standard for STA and the number one interactive timing analysis tool users trust.



What if you caught these issues early in the process?

The accuracy of TimingDesigner is ideal for high-speed, multi-frequency designs where it is essential to accurately model and analyze signal relationships between devices on a board, or embedded functions on an ASIC or programmable IC. As a graphical tool, TimingDesigner models complex digital circuit timing by combining an interactive timing diagram editor with a patented, dynamically linked spreadsheet. It also provides the option to model path delays, rise/fall times, effects of loading and temperature, and other complex formulas. The tool can trace all delay paths specified in the timing diagram, remove common uncertainties, adjust for track delays, select critical pats, and then compute worst-case timing margins. The effects of complex design changes can be instantly visualized in a custom report. Automatically calculated constraints identify timing violations so problem areas can be addressed ASAP.

Overall, TimingDesigner is the productivity-enhancing solution created to analyze, verify, and document your timing information, so you can get the right design at the right time.

More info on TimingDesigner, or contact us at info@ema-eda.com.

EMA Design Automation®