



# Guaranteeing Silicon Performance with FPGA Timing Models

## Authors Abstract

### Minh Mac

Member of Technical Staff,  
Technical Services  
Intel® Corporation

### Chris Wysocki

Senior Manager,  
Software Engineering  
Intel® Corporation

Intel® timing models provide a simple and easy way to verify the timing of FPGA designs without the need to perform full physical electrical extractions and simulations. The three different operating corners available for 65 nm and newer FPGAs provide a thorough coverage of the time delays within the recommended operating conditions.

## Introduction

How can a designer accurately predict the time delays of a fully customizable integrated circuit? The answer to this question is “not very easily.” There are many factors that limit and increase the complexity of accurately modeling time delays in an integrated circuit. A few of these factors include, but are not limited to large space of valid operating conditions (voltage, temperature, process, etc.), complex physical phenomena with (often) non-linear and complicated models, and variability of mass-produced silicon. Intel has devised a method to accurately predict the time delays for all designs implemented in its FPGAs.

To accurately model time delays within an FPGA, Intel uses a combination of two tools: a static timing analysis tool in the Intel Quartus® Prime software called Timing Analyzer and a proprietary circuit simulator with a delay database for each FPGA. The simulator combined with the delay database (containing the time delays) are also simply known as timing models. Timing models play a critical part in the FPGA design flow because they are used throughout the FPGA design compilation, from synthesis, through place-and-route, to timing simulation and analysis. This white paper provides an overview of the creation of timing models and the importance of timing models in Intel's FPGA design flow.

## Timing Model Components and Characteristics

Each FPGA has its own unique timing model that contains of all the necessary delay information for all physical elements in the device, such as the combinational adaptive logic modules, memory blocks, interconnects, and registers. The delays encompass all valid combinations of operating conditions for the target FPGA. Also, each element can contain different delay information depending upon the mode or configuration the element is configured to. Essentially, timing models are a software-based representation of the physical delays in the FPGA. To maximize model accuracy and minimize run time, the model is divided into two methods of producing the delay information: the delay database and the proprietary circuit simulator.

## Table of Contents

Abstract .....	1
Introduction .....	1
Timing Model Components and Characteristics .....	1
Delay Database .....	2
Proprietary Circuit Simulator .....	2
Modeling of On-die Variation and Other Uncertainties .....	2
Common Clock Path Pessimism Removal .....	3
Statistical Static Timing Analysis .....	3
Operating Conditions .....	3
Slow 85°C Timing Model .....	4
Slow 0°C Timing Model .....	4
Fast 0°C Timing Model .....	4
Timing Analyzer Corner Commands .....	4
Timing Model Generation Process .....	5
Pre-silicon Timing Model Support .....	7
Post-silicon Timing Model Support .....	7
Conclusion .....	7
Where to Get More Information ..	7

## Delay Database

The first method of supplying timing models to other Intel Quartus Prime software optimization and analysis engines is the delay database. This database is mostly used to store the delay arcs of logic and hard intellectual property (IP) blocks on the FPGA fabric, which are mainly parts of the FPGA architecture that have limited configurability. The delay database accounts for all possible configurations of these blocks as well as any differences in performance and delay between blocks that are functionally identical but have different physical implementation on the chip (i.e., different layout). The main advantage of this method is the low look-up time for delays, which translates into a short run time for the Timing Analyzer.

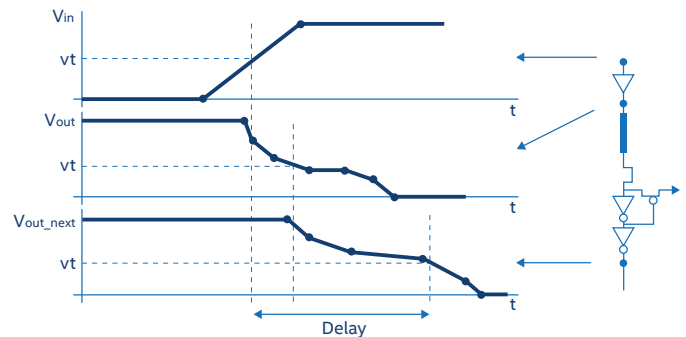
## Proprietary Circuit Simulator

The second method of generating timing models is the proprietary circuit simulator. The circuit simulator is required because many FPGA subcomponents have a very large configuration space, resulting in a wide range of propagation delays. For example, the routing interconnect delays cannot be modeled with a simple static value or even a table of values because there are too many independent electrical parameters leading to too many configurations. The capacitive loading, its distribution along the wire, the listening position, the varying RC as the wire goes through several metal layers, and the input waveform supplied to any of the interconnect wires are all determined by the place and route engine, leading to a wide range of electrical configurations.

The circuit simulator is very similar to other industry-standard circuit simulators such as HSPICE, in that it is time-domain based and can analyze linear electrical components (e.g., resistors, capacitors) and non-linear electronic components (e.g., transistors, pass gates). The simulator boasts 1% fidelity to HSPICE in terms of propagation delay results. It runs about 10,000 times faster than HSPICE and achieves such efficiency by being tuned specifically for FPGA subcomponents.

The circuit simulator is capable of extracting all electrical data, such as capacitance and resistance, and all non-linear and linear components, to determine the expected delays. The circuit simulator looks up electrical parameters of a series of subcomponents that are active in a user design within the FPGA, chooses the stimulus V-time waveform, builds an electrical netlist, and simulates in time domain to determine the actual propagation delay and the resulting V-time waveform. That output waveform is propagated as a stimulus for the analysis of the next downstream interconnect resource, and so on. The propagation delays are inferred from the differences in time of successive waveforms crossing the threshold voltage point.

Figure 1 shows a typical input V-time waveform (ramp) and output waveform from simulations of subsequent interconnect components.



**Figure 1.** V-time Waveforms Used by the Intel Quartus Prime Software Internal Electronic Simulator

## Modeling of On-die Variation and Other Uncertainties

The delays that are stored in the delay databases and generated by the circuit simulator are not just static values that are pieced together to predict the overall delay of elements in the FPGA. Instead, delay values vary depending upon modes of configuration, various parameterization, amount of variation, temperature, process, voltage, and performance level. The delay databases capture variation due to manufacturing processes, voltage, and temperature. To build a robust timing model, each delay database models:

- Manufacturing process on-die variation
- Rise and fall skew in uncorrelated N- and P-channel transistor speed
- Clock uncertainty and jitter
- Non-uniform voltage on the power distribution network (PDN)
- End-of-life degradation effects
- Slight variations in hardware design of equivalent blocks
- Crosstalk

The manufacturing process of a silicon wafer may introduce variation in identically designed blocks, which in turn causes delay differences between the blocks. For example, two identically designed combinational blocks located at different coordinates on a silicon die may exhibit local variation. This variation is due to a combination of perturbation in the manufacturing equipment, dopant levels, optical distortions in manufacturing masks, and other effects appearing in the creation of the silicon wafer. This type of variation is captured and modeled as the minimum and maximum delay range in the timing models.

The timing model already contains separate values for rise and fall propagation delays for all components. Independent variation between nominal parameters of N- channel and P-channel transistors, the fundamental building blocks of the FPGAs, can lead to relative variation between rise and fall propagation delays within the FPGA signal paths. In turn, this can lead to distortions in the clock-signal duty cycle—the ratio of risen parts to fallen parts of the clock cycle—or to increased skew in data signals. The intrinsic variation (minimum/maximum) in the Intel Quartus Prime software timing model captures this effect as well.

Clock uncertainty and jitter may exist on external clock sources connected to an FPGA. They may also arise along the dedicated clock networks and locally routed clocks as well as inside other clock management blocks due to variation of parasitics along these interconnect channels. Furthermore, random and deterministic jitter associated with phase-locked loops (PLLs), clock management blocks, and interconnect channels can vary the clock-signal propagation delays in time, which, therefore, might not be constant on a single die throughout the lifetime of operation. This static as well as temporal nature of clock uncertainty is all taken into account in the Intel Quartus Prime timing models and timing analysis. Optionally, designers can also overwrite any of the uncertainty values if they have good reasons for doing so (like extensive testing and knowledge that the operating conditions are tightly controlled).

As with many electrical components, aspects of the FPGAs degrade with time. This degradation causes the transistors that make up the FPGA to slow down due to end-of-life effects such as electromigration, negative bias temperature instability, and hot carrier injection. This type of variation is captured and modeled in the delay databases.

A large current draw from the PDN could be caused by localized and highly concentrated switching of FPGA programmable logic, or simply by switching at very high clock rates. This, in turn, can lead to a sustained or temporary voltage drop on the PDN and an increase in some of the propagation delays. The minimum/maximum timing model accounts for these effects by basing the propagation delay models on the worst-case voltage noise levels.

Capacitive crosstalk is the coupling of two or more large structures on the silicon die (usually interconnect wires), with a signal transition on one producing noise on the other one, effectively delaying or speeding up the signal transitions on one or both. In Intel FPGAs, crosstalk is prevented on key structures, like the dedicated clock networks, by shielding and proper spacing in the chip's layout. The regular interconnect paths are either guard-banded in the timing models or explicitly analyzed for crosstalk-induced delay changes in the timing model.

This modeling of variation and uncertainty is necessary to properly account for various physical effects, and hence accurately predict the worst-case performance of the FPGA with the Intel Quartus Prime software. However, in certain situations, the uncertainty models can introduce undesired pessimism into the timing model, which has a negative impact on the performance of the FPGA as reported by the Timing Analyzer or by any other timing simulation tool. Fortunately, the Intel Quartus Prime software includes two advanced modeling and analysis techniques to reduce or eliminate this pessimism: common clock path pessimism removal and statistical static timing analysis.

### Common Clock Path Pessimism Removal

Many register-to-register data transfers have a common clock signal and the routing of this signal to both registers may share common routing resources. Because the common part of the clock path uses the same physical resources, any on-die variation on these resources affects the source and destination registers the same way. Hence, that part of the on-die variation (minimum/maximum delay spread)

can be removed from the analysis of the transfer, which in turn removes some pessimism as reported by the Timing Analyzer. This type of variation pessimism removal is applicable for variation that is unique on each silicon die but is roughly constant throughout time. The reduction of pessimism due to temporal variation model, such as clock jitter, is also performed but is treated differently in the Intel Quartus Prime software.

### Statistical Static Timing Analysis

Some on-die variation is completely random and not correlated to any spatial positioning. Compared to short combinational paths, long combinational paths are much less likely to exhibit worst-case variation at every point along the way. For this reason, a module in the Intel Quartus Prime software uses statistics (backed by Monte Carlo simulations and silicon characterization) to mitigate the effect of random variation on longer paths. By discounting the minimum/maximum delay spread on these paths, the FPGA performance reported by the Timing Analyzer may increase.

## Operating Conditions

Intel FPGAs must operate in a continuum of conditions. These conditions include the die junction temperature, which varies depending upon the design's requirements. Commercial parts have a legal range of 0°C to 85°C and industrial parts have a legal range of -40°C to 100°C. There are even wider temperature ranges, such as those for automotive and military devices.

Another aspect of the operating conditions is the voltage supply levels. The most critical voltages for maintaining FPGA performance is the Vcc and the various I/O supplies. Each of the supply voltages has a legal operating range. For example, a subset of Stratix® IV FPGAs has a valid Vcc range of 0.87 V to 0.93 V.

The third aspect of the operating conditions is the relative speed of each FPGA versus the limit of the speed grade with which it is marked. This is one aspect that the designer has no control over. It should also be noted that devices within one speed grade can still differ slightly in performance, predominantly due to variation in the manufacturing process. All devices, however, are guaranteed to be faster than the limit of the speed grade.

Given the wide-ranged continuum of multi-parameter operating conditions, how does Intel guarantee that designers' timing constraints, and hence correct functionality of the programmed device, will be met? Intel provides a necessary and sufficient set of timing models, each generated at a specific operating-condition corner, which are subsequently used in the Intel Quartus Prime software compilation flow to run a separate and complete timing analysis at each corner. The operating-condition corners are usually the combinations of end points of the ranges in temperature, voltage, and manufacturing process. Intel ensures that all specified timing constraints are analyzed, including setup, hold, recovery, removal, and skew analysis. To accomplish this, all the available timing corners capture the very fastest performance, the very slowest performance, and everything in between.

Intel picks the minimum number of these timing corners to

ensure that the Intel Quartus Prime software compilation and timing analysis run time is minimized, thereby striking the right balance of guaranteed performance and minimized compile time. For a typical commercial Stratix IV device, as for many other FPGA families at 65 nm or smaller manufacturing nodes, Intel provides three timing models for each device at different operating conditions: Slow 85°C, Slow 0°C, and Fast 0°C. Each operating condition is used to model the timing delays under specific end point of temperature, voltage, and manufacturing process conditions.

### Slow 85°C Timing Model

The Slow 85°C timing model provides timing delays for the FPGA operating under the following conditions:

- Slowest silicon for the specific speed grade
- Low voltage (factoring in the lowest allowable user voltage supply levels and bounds of the voltage drop on the FPGA's internal PDN)
- 85°C junction temperature

This operating condition provides one of two possible worst-case conditions for the device. Because there are variations in silicon fabrication, each die that is cut from a silicon wafer has delay differences. The slow silicon is taken from the low end of the binning range for the specified speed grade. This gives the timing models the ability to capture worst-case and slowest delays for the FPGA. Low voltage decreases the transistor switching speed by decreasing electron mobility through the transistors. High temperature also affects transistor speed by changing the characteristics of the silicon material, leakage current, and electron mobility.

The combination of slow silicon, low voltage, and high temperature provides the first of two possible worst-case conditions for the FPGA. This condition is ideal for performing a setup check in static timing analysis but the Timing Analyzer engine performs all analyses (setup, hold, recovery, removal, and skew) at this and every other condition.

### Slow 0°C Timing Model

The Slow 0°C timing model provides timing delays for the FPGA operating under the following conditions:

- Slowest silicon for the specific speed grade
- Low voltage (factoring in the lowest allowable user voltage supply levels and bounds of the voltage drop on the FPGA's internal PDN)
- 0°C junction temperature

Similar to the Slow 85°C, the Slow 0°C model provides worst-case conditions for the device, which can arise at 0°C because of a transistor phenomenon known as temperature inversion. When temperature inversion occurs, a transistor's threshold voltage, which generally decreases with low temperatures, may overcome the speedup from larger carrier mobility at the low temperature; this may result in slower delays. Essentially, worst-case conditions may occur at low temperatures instead of high temperatures.

This combination of slow silicon, low voltage, and low temperature provides the second of two possible worst-case conditions. The condition is also ideal for performing setup

and recovery checks in static timing analysis.

### Fast 0°C Timing Model

The Fast 0°C timing model provides timing delays for the FPGA operating under the following conditions:

- Fast silicon
- High voltage (factoring in the highest allowable user voltage supply levels)
- 0°C junction temperature

Opposite to the Slow 85°C and 0°C operating conditions, the Fast 0°C provides best- case operating conditions for the device and results in overall shorter delays in the device. This condition is ideal for performing a hold and removing checks in static timing analysis.

With the Slow 0°C, Slow 85 °C, and Fast 0°C timing models of a commercial-grade FPGA, a thorough static timing analysis can be run that provides full coverage of all delays in the entire space of valid voltage, temperature, and process operating conditions. Other grades, such as industrial, also have their respective timing corners, all simulated and characterized at appropriate operating conditions by Intel. Given that the user has properly constrained all transfers and paths within the design, a Timing Analyzer result showing all constraints passing should provide a robust guarantee that the design will function correctly on a physical FPGA.

Timing models also provide the ability to adjust the legal operating temperature range that is used in the compilation flow. Industrial and other extended device models can be analyzed under a specific temperature operating range. For example, an industrial device can be compiled and analyzed at the 0°C to 100°C range, improving performance over the default industrial temperature range because some aspects of the performance would be lowered if the range extended to -40°C.

## Timing Analyzer Corner Commands

To take advantage of the various corners, simple commands can be issued to the Timing Analyzer. The following Timing Analyzer corner commands show the ease associated with modeling the complexities associated with predicting delays for an FPGA:

```
#Lists the available operating conditions get _
available _operating _conditions -all

#Specifies a slow process, speed grade 7, industrial
grade, voltage 1200mV, 100C #temperature
set _operating _conditions -model slow -speed 7
-grade i -voltage 1200 - temperature 100
<reporting commands>

#Specifies a slow process, speed grade 7, industrial
grade, voltage 1200mV, -40C #temperature
set _operating _conditions -model slow -speed 7
-grade i -voltage 1200 - temperature -40
<reporting commands>

#Specifies a fast process, speed grade 7, industrial
grade, voltage 1200mV, -40C #temperature
set _operating _conditions -model fast -speed 7
-grade i -voltage 1200 - temperature -40
<reporting commands>
```

Figure 2 shows the Timing Analyzer dialog box for a commercial-grade Stratix IV FPGA. The Operating Condition dialog box is used to specify a specific corner for timing analysis.

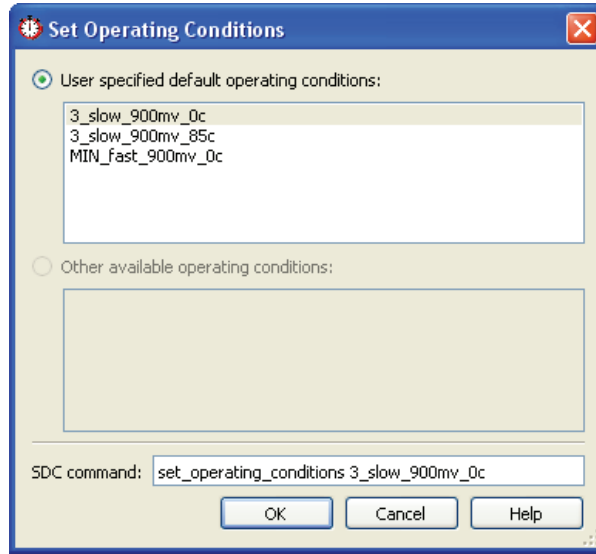


Figure 2. Operating Condition Dialog Box

## Timing Model Generation Process

Every device timing model goes through a process that involves two status designations, each of which indicates the state the timing model is in during the generation flow. The two status designations that a timing model goes through are:

- Preliminary—The model's initial release for the device in the Intel Quartus Prime software is considered preliminary until all planned characterization and correlation work is completed. This designation involves the generation of preliminary timing models for a new FPGA.
- Final—The timing model is fully correlated to silicon, so no further timing model changes are expected. This designation is the finalization of the preliminary timing models.

Figure 3 shows the typical flow a timing model takes to get from “preliminary” to “final.”

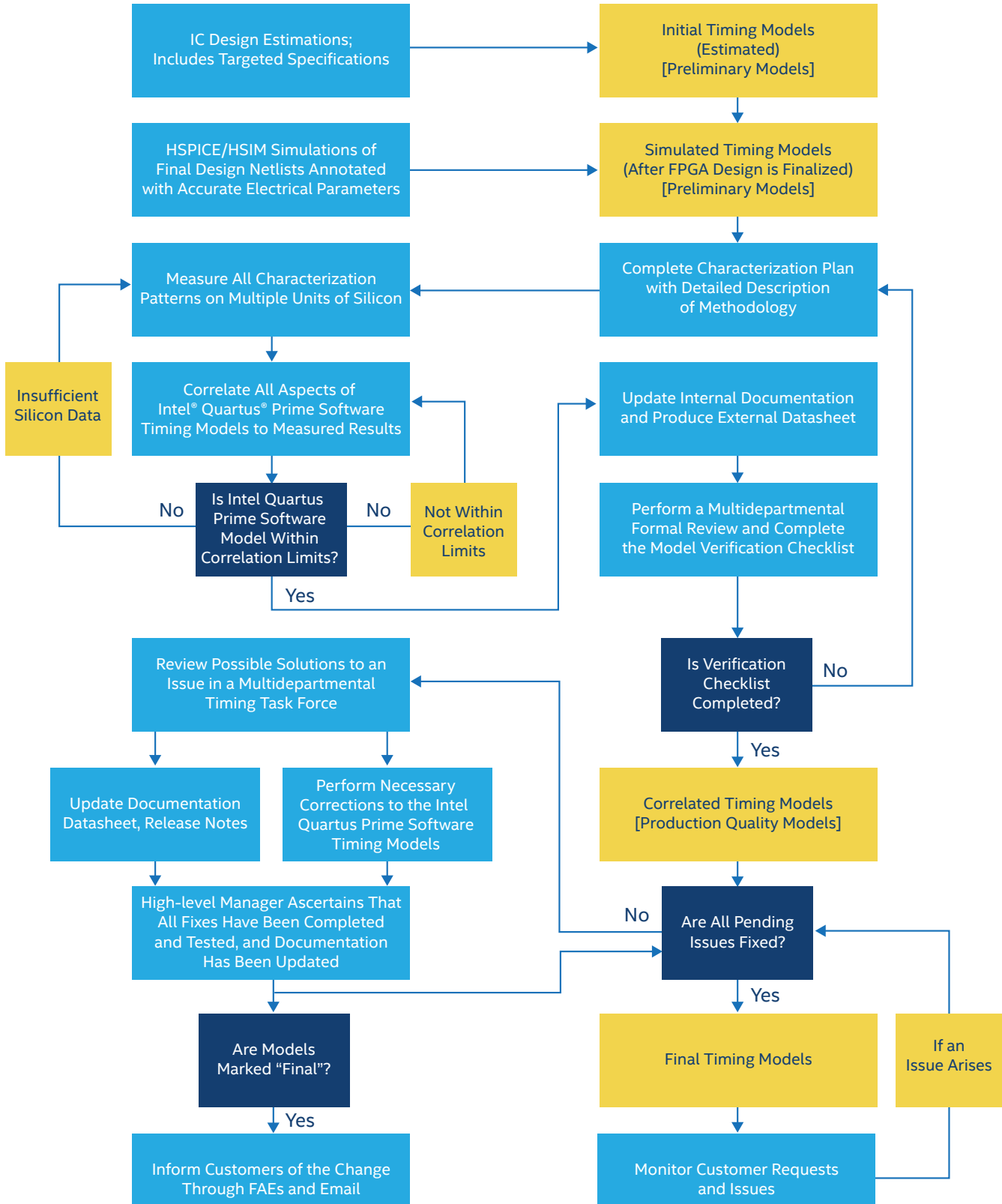


Figure 3. Timing Model Generation Flow



## Pre-silicon Timing Model Support

When a device family is initially introduced into the Intel Quartus Prime software, physical devices may not be available due to a lag between when device features are implemented in the software and when the foundry releases the silicon device. In the absence of silicon devices, the timing models are generated from simulation data derived from sources such as HSPICE models and standard-cell libraries. The simulated delays include information on electrical characteristics such as current-voltage tables, resistance, inductance, and capacitance for all the FPGA components.

All delays are initially generated in this manner for every element in the FPGA. The models are generated for multiple sets of operating conditions to capture the entire valid operating range. Guard bands are added to the preliminary models to account for various physical device effects, including on-die variation, electromigration, and end-of-life deterioration. Additionally, the guard band protects the timing models further to ensure that the timing analysis results are safe (or slightly conservative). If the Timing Analyzer shows that timing constraints are met (positive slack), then the FPGA will not encounter any timing failures under any valid operating condition.

The use of simulation data for the generation of preliminary timing models provides a robust method to accurately predict the physical delays of the FPGA. Using a preliminary timing model does not imply that the timing model will remain static between different Intel Quartus Prime software versions. For example, a device timing model marked preliminary in the Intel Quartus Prime version 10.0 will not be equivalent to the preliminary timing model in the Intel Quartus Prime version 10.0 SP1 for the same device, because the preliminary timing model is constantly updated to reflect actual physical delays and silicon characterization work. The act of characterizing physical delays and measuring the delays against the simulated delays is accomplished through a process called silicon correlation. The correlation stage happens throughout the life of the timing model until the timing model is marked "final."

## Post-silicon Timing Model Support

Once a physical device is available, silicon delays are measured and characterized. The second designation involves correlating the timing models with characterized measurements. All silicon delays are measured with thousands of correlation patterns. Each correlation pattern is used to measure specific delays in the device. Multiple devices are used in the correlation effort to produce statistically sound measurement data to gauge the timing performance across a wide range of process variations. The preliminary timing models are updated to match the delays characterized in the physical device. After the correlation work is completed, all correlation patterns have been measured and the timing models reflect silicon measurements, the timing models are marked "final."

## Conclusion

Intel timing models provide a simple and easy way to verify the timing of designs without the need to perform a full physical electrical extraction and simulation. The three different operating corners available for 65 nm and newer devices provide a thorough coverage of the timing delays for FPGA within the recommended operating conditions. Timing models include all electrical characteristics of transistors in the device. Using Intel's timing models results in a level of complexity that is removed from the timing verification flow for FPGAs when compared to the timing verification flow for ASICs.

## Where to Get More Information

- Timing Analyzer section, Volume 3 of the Intel Quartus Prime Software Handbook:  
[www.altera.com/content/dam/altera-www/global/en\\_US/pdfs/literature/hb/qts/qts\\_qii5v3.pdf](http://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/qts/qts_qii5v3.pdf)
- Intel FPGA Development Tools:  
[www.altera.com/products/design-software/overview.html](http://www.altera.com/products/design-software/overview.html)

