



Intel® Stratix® 10 Device Design Guidelines



Subscribe



Send Feedback

S10-GUIDELINES | 2021.04.22

Latest document on the web: [PDF](#) | [HTML](#)

Contents

Intel® Stratix® 10 Device Design Guidelines	4
Design Flow.....	4
System Specification.....	5
Design Specifications.....	6
IP Selection.....	6
Platform Designer.....	7
Device Selection.....	7
Device Variant.....	8
PLLs and Clock Routing.....	8
Logic, Memory, and Multiplier Density.....	8
I/O Pin Count, LVDS Channels, and Package Offering.....	9
Speed Grade.....	10
Vertical Device Migration.....	10
Early System and Board Planning.....	11
Power and Thermal Estimation.....	11
Thermal Management and Design.....	13
Temperature Sensing for Thermal Management.....	13
Voltage Sensor.....	14
Planning for Device Configuration.....	14
Planning for On-Chip Debugging.....	20
Pin Connection Considerations for Board Design.....	23
Device Power-Up.....	23
Power Pin Connections and Power Supplies.....	24
Configuration Pin Connections.....	26
Board-Related Intel Quartus Prime Settings.....	30
Signal Integrity Considerations.....	31
Board-Level Simulation and Advanced I/O Timing Analysis.....	33
I/O and Clock Planning.....	33
Making FPGA Pin Assignments.....	34
Early Pin Planning and I/O Assignment Analysis.....	35
I/O Features and Pin Connections.....	36
Clock and PLL Selection.....	40
PLL Feature Guidelines.....	41
Clock Control Features.....	42
I/O Simultaneous Switching Noise.....	42
Security Considerations.....	43
Design Entry.....	45
Design Recommendations.....	45
Using IP Cores.....	45
Reconfiguration.....	46
Recommended HDL Coding Styles.....	46
Design Implementation, Analysis, Optimization, and Verification.....	47
Selecting a Synthesis Tool.....	47
Device Resource Utilization Reports.....	48
Intel Quartus Prime Messages.....	48
Timing Constraints and Analysis.....	49
Area and Timing Optimization.....	50

Preserving Performance and Reducing Compilation Time.....	51
Designing with Intel Hyperflex™	52
Simulation.....	52
Power Analysis.....	53
Power Optimization.....	54
Document Revision History for Intel Stratix 10 Device Design Guidelines.....	57

Intel® Stratix® 10 Device Design Guidelines

This document provides a set of design guidelines, recommendations, and a list of factors to consider for designs that use Intel® Stratix® 10 FPGAs. It is important to follow Intel recommendations throughout the design process for high-density, high-performance Intel Stratix 10 designs. This document also assists you with planning the FPGA and system early in the design process, which is crucial to successfully meet design requirements.

Note: This document does not include all Intel Stratix 10 device details and features. For more information about Intel Stratix 10 devices and features, refer to the respective Intel Stratix 10 User Guides.

The material references the Intel Stratix 10 device architecture as well as aspects of the Intel Quartus® Prime software and third-party tools that you might use in your design. The guidelines presented in this document can improve productivity and avoid common design pitfalls.

Related Information

[Intel Stratix 10 Documentation](#)

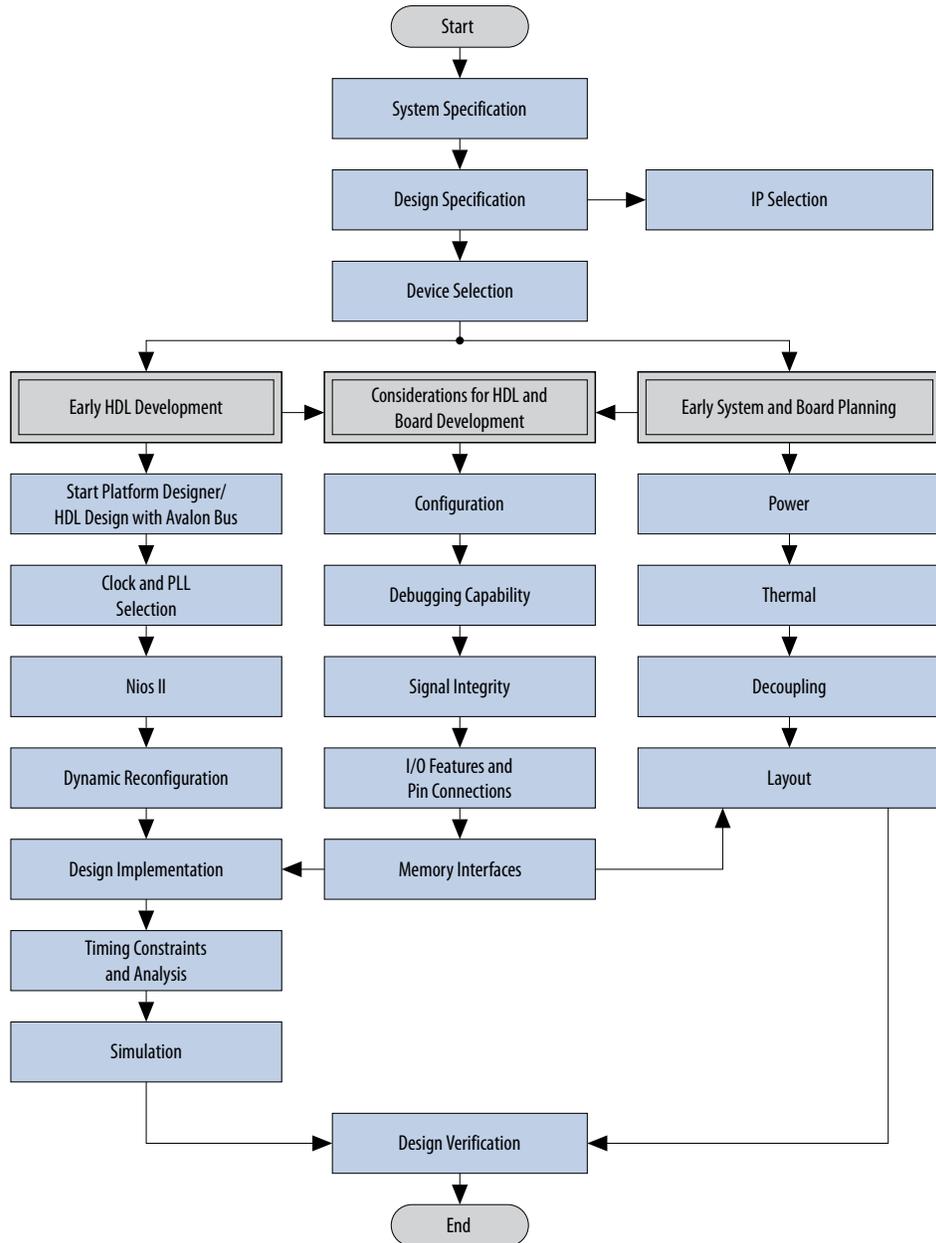
Provides more information about Intel Stratix 10 devices and features.

Design Flow

Table 1. Summary of the Design Flow Stage and Guideline Topics

Stages of the Design Flow	Description
System Specification	Planning design specifications, IP selection
Device Selection	Device information, determining device variant and density, package offerings, migration, speed grade
Early System and Board Planning	Power and thermal estimation, thermal management option, planning for configuration scheme, planning for on-chip debugging
Pin Connection Considerations for Board Design	Power-up, power pins, PLL connections, decoupling capacitors, configuration pins, signal integrity, board-level verification
I/O and Clock Planning	Pin assignments, early pin planning, I/O features and connections, memory interfaces, clock and PLL selection, simultaneous switching noise (SSN)
Design Entry	Coding styles and design recommendations, Platform Designer, planning for hierarchical or team-based design
Design Implementation, Analysis, Optimization, and Verification	Synthesis tool, device utilization, messages, timing constraints and analysis, area and timing optimization, compilation time, verification, power analysis and optimization

Figure 1. Intel Stratix 10 Device Design Flow



System Specification

In systems that contain an Intel Stratix 10 device, the FPGA typically plays a large role in the overall system and affects the rest of the system design. It is important to start the design process by creating detailed design specifications for the system and the FPGA, and determining the FPGA input and output interfaces to the rest of the system.

Design Specifications

Table 2. Design Specifications Checklist

Number	Done?	Checklist Item
1		Create detailed design specifications and a test plan if appropriate.
2		Plan clock resources and I/O interfaces early with a block diagram.

Create detailed design specifications that define the system before you create your logic design or complete your system design, by performing the following:

- Specify the I/O interfaces for the FPGA
- Identify the different clock domains
- Include a block diagram of basic design functions
- Include intellectual property (IP) blocks

Note: Taking the time to create these specifications improves design efficiency, but this stage is often skipped by FPGA designers.

- Create a functional verification/test plan
- Consider a common design directory structure

Create a functional verification plan to ensure the team knows how to verify the system. Creating a test plan at this stage can also help you design for testability and design for manufacturability. For example, do you want to perform built-in-self test (BIST) functions to drive interfaces? If so, you could use a UART interface with a Nios® processor inside the FPGA device. You might require the ability to validate all the design interfaces.

If your design includes multiple designers, it is useful to consider a common design directory structure. This eases the design integration stages.

Related Information

- [IP Selection](#) on page 6
Provides suggestions about including intellectual property blocks.
- [Planning for On-Chip Debugging](#) on page 20
Provides the guidelines related to analyzing and debugging the device after the device is in the system.

IP Selection

Table 3. IP Selection Checklist

Number	Done?	Checklist Item
1		Select IP that affects system design, especially I/O interfaces.
2		If you plan to evaluate Intel FPGA IP, ensure that your board design supports JTAG connections.

Intel and its third-party IP partners offer a large selection of off-the-shelf IP cores optimized for Intel devices. You can easily implement these parameterized blocks of IP in your design, reducing your system implementation and verification time, and allowing you to concentrate on adding proprietary value.

IP selection often affects system design, especially if the FPGA interfaces with other devices in the system. Consider which I/O interfaces or other blocks in your system design can be implemented using IP cores, and plan to incorporate these cores in your FPGA design.

The Intel FPGA IP Evaluation Mode feature available for many IP cores allows you to program the FPGA to verify your design in hardware before you purchase the IP license. The evaluation supports an untethered mode, in which the design runs for a limited time, or a tethered mode. The tethered mode requires an Intel serial JTAG cable connected between the JTAG port on your board and a host computer running the Intel Quartus Prime Programmer for the duration of the hardware evaluation period.

Related Information

[Intellectual Property page](#)

Provides more descriptions of the available IP cores.

Platform Designer

Table 4. Platform Designer Checklist

Number	Done?	Checklist Item
1		Take advantage of Platform Designer for system and processor designs.

Platform Designer is a system integration tool included as part of the Intel Quartus Prime software. Platform Designer captures system-level hardware designs at a high level of abstraction and automates the task of defining and integrating customized Hardware Description Language (HDL) components. These components include IP cores, verification IP, and other design modules. Platform Designer facilitates design reuse by packaging and integrating your custom components with Intel and third-party IP components. Platform Designer automatically creates interconnect logic from the high-level connectivity you specify, thereby eliminating the error-prone and time-consuming task of writing HDL to specify system-level connections.

Platform Designer is more powerful if you design your custom components using standard interfaces. By using standard interfaces, your components inter-operate with the components in the Platform Designer Library. In addition, you can take advantage of bus functional models (BFMs), monitors, and other verification IP to verify your design.

Related Information

[Creating a System With Platform Designer chapter, Platform Designer User Guide \(Intel Quartus Prime Pro Edition\)](#)

Provides more information about the Platform Designer.

Device Selection

Device selection is the first step in the Intel Stratix 10 design process—choosing the device family variant, device density, features, package, and speed grade that best suit your design requirements.

Related Information

[Intel Stratix 10 Device Overview](#)

Provides more information about the features available in each device density, including logic, memory blocks, multipliers, and phase-locked loops (PLLs), as well as the various package offerings and I/O pin counts.

Device Variant

Table 5. Device Variant Checklist

Number	Done?	Checklist Item
1		Consider the available device variants.
2		Select a device based on transceivers, protocol IP cores, I/O pin count, LVDS channels, package offering, logic/memory/multiplier density, PLLs, clock routing, and speed grade.

The Intel Stratix 10 device family consists of several device variants optimized to meet different application requirements.

Related Information

[Intel Stratix 10 Family Variants](#), [Intel Stratix 10 Device Overview](#)

Provides more information about the device variants.

PLLs and Clock Routing

Table 6. PLLs and Clock Routing Checklist

Number	Done?	Checklist Item
1		Verify the number of PLLs and clock resources.

Verify that your chosen device density package combination includes enough PLLs and clock routing resources for your design.

Related Information

[PLLs Overview](#), [Intel Stratix 10 Clocking and PLL User Guide](#)

Provides more information about PLLs.

Logic, Memory, and Multiplier Density

Table 7. Logic, Memory, and Multiplier Density Checklist

Number	Done?	Checklist Item
1		Estimate the required logic, memory, and multiplier density.
2		Reserve device resources for future development and debugging.

Intel Stratix 10 devices offer a range of densities that provide different amounts of device logic resources, including memory, multipliers, and adaptive logic module (ALM) logic cells. Determining the required logic density can be a challenging part of the design planning process. Devices with more logic resources can implement larger and potentially more complex designs, but generally have a higher cost. Smaller devices have lower static power utilization. Intel Stratix 10 devices support vertical migration, which provides flexibility.

You may observe 10% – 15% increase in resource utilization when you migrate your design from existing devices to Intel Stratix 10 devices. Review the resource utilization to find out which device density fits the design. Consider that the coding style, device architecture, and optimization options used in the Intel Quartus Prime software can significantly affect the resource utilization and timing performance of a design.

Select a device that meets your design requirements with some safety margin in case you want to add more logic later in the design cycle, upgrade, or expand your design. You might also want additional space in the device to ease design floorplan creation for an incremental or team-based design. Consider reserving resources for debugging.

Related Information

- [Vertical Device Migration](#) on page 10
- [Device Resource Utilization Reports](#) on page 48
Provides more information about determining resource utilization for a compiled design.
- [Documentation: IP and Megafunctions](#)
Provides more information about resource utilization estimates for certain IP design configurations.
- [Planning for On-Chip Debugging](#) on page 20
Provides more information about reserving resources for debugging.
- [Intel Stratix 10 Variable Precision DSP Blocks User Guide](#)

I/O Pin Count, LVDS Channels, and Package Offering

Table 8. I/O Pin Count, LVDS Channels, and Package Offering Checklist

Number	Done?	Checklist Item
1		Estimate the number of I/O pins that you require.
2		Consider the I/O pins you need to reserve for debugging.
3		Verify that the number of LVDS channels are enough.

Determine the required number of I/O pins for your application, considering the design interface requirements with other system blocks.

Larger densities and package pin counts offer more full-duplex LVDS channels for different signaling; ensure that your device density-package combination includes enough LVDS channels. Other factors can also affect the number of I/O pins required for a design, including simultaneous switching noise (SSN) concerns, pin placement guidelines, pins used as dedicated inputs, I/O standard availability for each I/O bank, differences between I/O standards and speed for row and column I/O banks, and package migration options.

You can compile any existing designs in the Intel Quartus Prime software to determine how many I/O pins are used. Also consider reserving I/O pins for debugging.

Related Information

- [Available Options, Intel Stratix 10 Device Overview](#)
Provides more information about the package types.

- [GPIO Buffers and LVDS Channels in Devices, Intel Stratix 10 General Purpose I/O User Guide](#)
Provides more information about the I/O pin counts.
- [LVDS Channels Support, Intel Stratix 10 High-Speed LVDS I/O User Guide](#)
- [Pin Connection Considerations for Board Design](#) on page 23
Provides more information about choosing pin locations.
- [I/O and Clock Planning](#) on page 33
Provides more information about choosing pin locations.
- [Planning for On-Chip Debugging](#) on page 20
Provides more information about reserving I/O pins for debugging.

Speed Grade

Table 9. Speed Grade Checklist

Number	Done?	Checklist Item
1		Determine the device speed grade that you require.

The device speed grade affects the device timing performance and timing closure, as well as power utilization. One way to determine which speed grade your design requires is to consider the supported clock rates for specific I/O interfaces.

You can use the fastest speed grade during prototyping to reduce compilation time (because less time is spent optimizing the design to meet timing requirements), and then move to a slower speed grade for production to reduce cost if the design meets its timing requirements.

Related Information

- [Available Options, Intel Stratix 10 Device Overview](#)
Provides more information about the device speed grades.
- [Performance Support Summary, Intel Stratix 10 External Memory Interfaces User Guide](#)
Provides more information about the supported clock rates for memory interfaces using I/O pins on different sides of the device in different device speed grades.
- [External Memory Interface Spec Estimator](#)
Provides more information about the performance of the supported external memory interfaces in Intel FPGA devices.
- [Switching Characteristics, Intel Stratix 10 Device Datasheet](#)
Provides the transceiver specifications for the L-tile, H-tile, and E-tile devices.

Vertical Device Migration

Table 10. Vertical Device Migration Checklist

Number	Done?	Checklist Item
1		Consider vertical device migration availability and requirements.
2		Refer to the <i>External Memory Interfaces Intel Stratix 10 FPGA IP User Guide</i> for the external memory interface (EMIF) pin pairing.

Intel Stratix 10 devices support vertical migration within the same package, which enables you to migrate to different density devices whose dedicated input pins, configuration pins, and power pins are the same for a given package. This feature allows future upgrades or changes to your design without any changes to the board layout, because you can replace the FPGA on the board with a different density Intel Stratix 10 device.

Determine whether you want the option of migrating your design to another device density. Choose your device density and package to accommodate any possible future device migration to allow flexibility when the design nears completion. You should specify any potential migration options in the Intel Quartus Prime software at the beginning of your design cycle or as soon as the device migration selection is possible in the Intel Quartus Prime software. Selecting a migration device can impact the design pin placement, because the Fitter ensures your design is compatible with the selected devices. It is possible to add migration devices later in the design cycle, but it requires extra effort to check pin assignments, and can require design or board layout changes to fit into the new target device. It is easier to consider these issues early in the design cycle than at the end, when the design is near completion and ready for migration.

The Intel Quartus Prime Pin Planner highlights pins that change function in the migration device when compared to the currently selected device.

Related Information

- [Making FPGA Pin Assignments](#) on page 34
Provides more information about changing pin functions using the Intel Quartus Prime Pin Planner.
- [External Memory Interfaces Intel Stratix 10 FPGA IP User Guide](#)
Provides the design guidelines for EMIF pin pairing in the Pin Guidelines for External Memory Interfaces Intel Stratix 10 FPGA IP section of the respective external memory interfaces.
- [Transceiver Design Flow, Intel Stratix 10 Transceiver PHY User Guide](#)
- [Intel Stratix 10 E-Tile Transceiver PHY User Guide](#)
- [Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide](#)
- [AN 778: Intel Stratix 10 Transceiver Usage](#)

Early System and Board Planning

System information related to the FPGA should be planned early in the design process, before designers have completed the design in the Intel Quartus Prime software. Early planning allows the FPGA team to provide early information to PCB board and system designers.

Power and Thermal Estimation

Table 11. Power and Thermal Estimation

Number	Done?	Checklist Item
1		Estimate power consumption with the Intel FPGA Power and Thermal Calculator (PTC) to plan the cooling solution and power supplies before the logic design is complete.

FPGA power consumption is an important design consideration and must be estimated accurately to develop an appropriate power budget to design the power supplies, voltage regulators, decouplers, heat sink, and cooling system. Power estimation and analysis have two significant planning requirements:

- Thermal planning—The cooling solution must sufficiently dissipate the heat generated by the device. In particular, the computed junction temperature must fall within normal device specifications.
- Power supply planning—The power supplies must provide adequate current to support device operation.

Power consumption in FPGA devices is dependent on the logic design. This dependence can make power estimation challenging during the early board specification and layout stages. The Intel PTC tool allows you to estimate power utilization before the design is complete by processing information about the device and the device resources needed in the design, as well as the operating frequency, toggle rates, and environmental considerations. You can use the tool to obtain thermal design parameters, with which you can perform detailed thermal simulation and cooling solution design.

If you do not have an existing design, estimate the number of device resources used in your design and enter it manually. The PTC tool accuracy depends on your inputs and your estimation of the device resources. If this information changes (during or after your design is complete), your power estimation results are less accurate. If you have an existing design or a partially-completed compiled design, use the `Generate the Power and Thermal Calculator Import File` command in the Intel Quartus Prime software to provide input to the PTC.

The PTC can read the Intel Quartus Prime-generated power estimation file (`.qptc` file), or alternatively a `.csv` file generated by an older version of the power estimator tool, called the Early Power Estimator (EPE). You can also transfer the data into the PTC tool manually by referring to the resource usage in the Intel Quartus Prime Pro Edition-generated compilation reports for your design. You should enter additional resources to be used in the final design manually if the existing Intel Quartus Prime project represents only a portion of your full design. You can edit the inputs to the PTC tool and add additional device resources or adjust the parameters after importing the power estimation file information.

When the design is complete, the Power Analyzer tool in the Intel Quartus Prime software provides more accurate estimation of power, ensuring that thermal and supply budgets are not violated. For the most accurate power estimation, use gate-level simulation results with an output file (`.vcd`) from a third-party simulation tool.

Note: To obtain the PTC tool, contact your local sales representative.

Related Information

- [Power Analysis](#) on page 53
Provides more information about power estimation and analysis.
- [Intel FPGA Power and Thermal Calculator User Guide](#)
Provides more information about using the PTC.
- [Power Analysis chapter, Power Analysis and Optimization User Guide \(Intel Quartus Prime Pro Edition\)](#)
Provides more information about power estimation and analysis.

Thermal Management and Design

Table 12. Temperature Design Checklist

Number	Done?	Checklist Item
1		Obtain thermal design power and thermal parameters from PTC.
2		Perform thermal simulation to determine a proper cooling solution.

Intel Stratix 10 is a multi-chip module, depending on package configuration and design information, power distribution on all dies can be quite different. This feature makes the Intel Stratix 10 thermal characteristics design dependent. PTC takes into consideration of your design input, and generates unique thermal parameters for your design in the Thermal Page. You can get power consumption for each die, thermal resistance for all dies (ψ_{JC}), cooling solution requirement (ψ_{CA}), and maximum allowed package case temperature (T_{case}).

The thermal analysis of the Intel Stratix 10 device requires you use a Compact Thermal Model (contact your local Intel representatives to obtain the model) and perform simulation in a Computational Fluid Dynamics (CFD) tool. The result of the CFD analysis gives the T_{case} which should be lower than the required value in the PTC Thermal Page. With the simulated T_{case} , ψ_{JC} , and total package power, you can obtain the actual junction temperature T_j , which needs to stay below your requirement, for example, 95°C. You can adjust your cooling solution (heat sink design, airflow, and so on) to optimize your thermal design.

Related Information

- [AN 787: Intel Stratix 10 Thermal Modeling and Management](#)
Provides information about Intel Stratix 10 thermal modeling and gives an example of a thermal design for Intel Stratix 10 devices.
- [Intel FPGA Power and Thermal Calculator User Guide](#)
Provides more information about using the PTC.

Temperature Sensing for Thermal Management

Table 13. Temperature Sensing Checklist

Number	Done?	Checklist Item
1		Set up the temperature sensing diode (TSD) in your design to measure the device junction temperature for thermal management.
2		Include offset values from PTC calculation to TSD reading.

Intel Stratix 10 devices have internal and external temperature sensing capabilities. You can measure the junction temperature, T_j , with the following methods:

- Using internal TSD by instantiating the Temperature Sensor Intel Stratix 10 FPGA IP core.
- Using external thermal diode designed to interface with third-party sensor chip. Ensure the third-party sensor chip matches the external TSD specifications as documented in the *Intel Stratix 10 Device Datasheet*.

Monitoring the actual junction temperature is crucial for thermal management. Intel Stratix 10 devices include TSD on each die with embedded analog-to-digital converter (ADC) circuitry. You can access the digital temperature readout through the Temperature Sensor IP core.

The Intel Stratix 10 TSD can self-monitor the device junction temperature and can be used with external circuitry for activities such as controlling air flow to the FPGA. This requires including the TSD circuitry by instantiating a Temperature Sensor IP core.

The flexibility in Intel Stratix 10 design can lead to non-uniform power distribution on transceiver dies. Therefore, the hotspot on the transceiver dies may not necessarily be at the TSD location. This results in a temperature difference between TSD readout and real junction temperature. PTC calculates this difference and reports an offset value for each TSD. You need to add the offset values to your TSD reading to get the actual junction temperature.

Related Information

- [Temperature Sensor, Intel Stratix 10 Analog to Digital Converter User Guide](#)
Provides more information about the board design guidelines related to configuration pins and connecting devices for configuration.
- [Temperature Sensing Diode Specifications, Intel Stratix 10 Device Datasheet](#)
Provides the internal and external temperature sensing diode specifications.
- [AN 787: Intel Stratix 10 Thermal Modeling and Management](#)
Provides information about Intel Stratix 10 thermal modeling and gives an example of a thermal design for Intel Stratix 10 devices.

Voltage Sensor

Table 14. Voltage Sensor Checklist

Number	Done?	Checklist Item
1		Determine if you need to use the voltage sensor.

Intel Stratix 10 devices have an on-chip voltage sensor. The sensor provides a 8-bit digital representation of the analog signal being observed. This feature can be used for live monitoring of critical on-chip power supplies and external analog voltage.

Related Information

- [ADC Design Considerations, Intel Stratix 10 Analog to Digital Converter User Guide](#)
Provides more information about the voltage sensor feature.

Planning for Device Configuration

Table 15. Planning for Device Configuration Checklist

Number	Done?	Checklist Item
1		Consider whether you require multiple configuration schemes.
2		To avoid configuration failure, follow the configuration guidelines and additional clock requirements if your design is using PCIe, transceiver channels, HPS EMIF, High Bandwidth Memory (HBM2) IP core, or SmartVID. Refer to the <i>Intel Stratix 10 Configuration User Guide</i> and <i>Intel Stratix 10 Power Management User Guide</i> for the guidelines.
<i>continued...</i>		

Number	Done?	Checklist Item
3		Intel strongly recommends using the Intel Stratix 10 Reset Release IP in your design to provide a known initialized state for your logic to begin operation. The Reset Release IP is available in the Intel Quartus Prime software version 19.1 and later. Refer to the <i>Intel Stratix 10 Configuration User Guide</i> for the guidelines.
4		Each die in the Intel Stratix 10 GX 10M devices has its own configuration pins. Configure each die separately by using the dedicated configuration bit stream or .sof file. The device cannot be configured using a single .sof file because each die has a unique JTAG ID.
5		<p>Ensure nCONFIG is connected in following manner:</p> <ul style="list-style-type: none"> • If you are using AS x4 configuration, connect nCONFIG to VCCIO_SDM through an external 10 kΩ pull-up resistor. If the nCONFIG pin is connected and controlled by a host, the host must drive the nCONFIG to a known state. However, Intel recommends connecting nCONFIG to an external 10 kΩ pull-up resistor to VCCIO_SDM so that the FPGA can be configured directly from Quad SPI flash when the device is powered up. • If you are using Avalon®-ST configuration scheme, drive the nCONFIG with the host to a known a state. Connecting nCONFIG to an external 10 kΩ pull-up resistor to VCCIO_SDM is optional. • When you use a host to drive nCONFIG regardless of the configuration scheme, the host must monitor nSTATUS appropriately as described in the <i>Intel Stratix 10 Device Family Pin Connection Guidelines</i> and <i>Intel Stratix 10 Configuration User Guide</i> to enable reliable configuration.
6		Ensure that nCONFIG is not directly driven by FPGA, HPS I/Os, or any component that has dependency on FPGA or HPS I/Os.
7		For Intel Stratix 10 devices, do not reset the Quad SPI flash when used as the configuration device and data storage device with FPGA. Resetting the Quad SPI flash during the FPGA configuration and reconfiguration, or during the READ/WRITE/ERASE operations of the Quad SPI, causes undefined behavior for Quad SPI flash and the FPGA. To recover from the unresponsive behavior, you must power cycle your device. To reset the Quad SPI flash via the external host, you must first complete the FPGA configuration and reconfiguration, or a Quad SPI operation, and only then toggle the reset. The Quad SPI operation is complete when the exclusive access to the Quad SPI flash is closed by issuing the QSPI_CLOSE command via the Mailbox Client Intel FPGA IP or CLOSE command via the Serial Flash Mailbox Client Intel FPGA IP.

Intel Stratix 10 devices are based on SRAM cells. You must download configuration data to the Intel Stratix 10 device each time the device powers up, because SRAM is volatile. Consider whether you require multiple configuration schemes, such as one for debugging or testing and another for the production environment.

Choosing the device configuration method early allows system and board designers to determine what companion devices, if any, are required for the system. Your board layout also depends on the configuration method you plan to use for the programmable device, because different schemes require different connections.

In addition, Intel Stratix 10 devices offer advanced configuration features, depending on your configuration scheme. Intel Stratix 10 devices also include optional configuration pins and a reconfiguration option that you should choose early in the design process (and set up in the Intel Quartus Prime software), so you have all the information required for your board and system design.

Related Information

- [Pin Connection Considerations for Board Design](#) on page 23
Provides more information about the board design guidelines related to configuration pins and connecting devices for configuration.
- [Intel Stratix 10 Reset Release IP, Intel Stratix 10 Configuration User Guide](#)
Provides more information about the Reset Release IP.

- [Power Sequencing Considerations for Intel Stratix 10 Devices, Intel Stratix 10 Power Management User Guide](#)
Provides more information about SmartVID settings and power sequencing requirements.
- [Configuration Support Center](#)
Provides more information about configuration.
- [FPGA Configuration Troubleshooter](#)
Provides guide on debug and configuration issues.
- [JTAG Configuration and ISP Troubleshooter](#)
Provides guide on debug and configuration issues.
- [OSC_CLK_1 Clock Input, Intel Stratix 10 Configuration User Guide](#)
Provides the additional configuration clock requirements for Intel Stratix 10 (OSC_CLK_1).
- [Additional Clock and SmartVID Requirements for Transceivers, HPS, High Bandwidth, Intel Stratix 10 Configuration User Guide](#)
Provides the configuration guidelines and additional clock requirements if your design is using PCIe, transceiver channels, HPS, High Bandwidth Memory (HBM2) IP core, or SmartVID.
- [Intel Stratix 10 Device Family Pin Connections Guidelines](#)
Provides more details about nSTATUS and nCONFIG connections.
- [Intel Stratix 10 Configuration Details chapter, Intel Stratix 10 Configuration User Guide](#)
Provides more information about nSTATUS and nCONFIG operations.

Device Power Cycling and Reconfiguration

Table 16. Device Power Cycling and Reconfiguration Checklist

Number	Done?	Checklist Item
1		Consider designing your system to support power cycling the device to ensure error recovery under all reconfiguration circumstances.

Intel Stratix 10 devices use the triple-redundant processors in the SDM to control configuration. To ensure error recovery under all reconfiguration circumstances, Intel recommends designing your system to support power cycling the device if needed. Generally, asserting nCONFIG provides adequate error recovery. If not, you may need to perform a power cycle. A power cycle completely re-initializes the device, samples MSEL, reads the fuses, and runs the SDM BootROM code. Follow the device power up and power down sequences during the power cycle.

Related Information

[Intel Stratix 10 Configuration Details chapter, Intel Stratix 10 Configuration User Guide](#)
Provides more information about power cycling and reconfiguration.

Configuration Scheme Selection

Table 17. Configuration Scheme Selection Checklist

Number	Done?	Checklist Item
1		Select a configuration scheme to plan companion devices and board connections.

Intel Stratix 10 devices offer several configuration schemes.

You can enable any specific configuration scheme by driving the Intel Stratix 10 device MSEL pins to specific values on the board.

All configuration schemes use a configuration device, a download cable, or an external controller (for example, MAX® (MAX II, MAX V, Intel MAX 10) devices or a microprocessor).

Related Information

[Configuration Details, Intel Stratix 10 Configuration User Guide](#)

Provides more information about the Intel Stratix 10 device supported configuration schemes, how to execute the required configuration schemes, and all of the necessary option pin settings, including the MSEL pin settings.

Serial Configuration Devices

Table 18. Serial Configuration Devices Checklist

Number	Done?	Checklist Item
1		If you want to use the AS configuration mode with large device densities, confirm there is a configuration device available that is large enough for your target FPGA density.

Third-party serial flash devices are used in the AS configuration scheme.

Serial configuration devices can be programmed using a Intel FPGA Download Cable II or Intel FPGA Ethernet Cable II download cable with the Intel Quartus Prime software through the active serial interface.

Alternatively, you can use supported third-party programmers such as BP Microsystems and System General, or a microprocessor with the SRunner software driver. SRunner is a software driver developed for embedded serial configuration device programming that designers can customize to fit in different embedded systems.

Serial configuration devices do not directly support the JTAG interface; however, you can program the device with JTAG download cables using the Intel Stratix 10 FPGA as a bridge between the JTAG interface and the configuration device, allowing both devices to use the same JTAG interface.

Programming the serial flash from JTAG using the Intel Stratix 10 FPGA as a bridge is slower than using the standard AS interface.

Related Information

- [Supported Flash Devices for Intel Stratix 10](#)
Provides more information about the third-party serial flash supported by Intel Stratix 10 FPGA.
- [AN 418: SRunner: An Embedded Solution for Serial Configuration Device Programming](#)
Provides more information about the SRunner software.

Download Cables

Table 19. Download Cables Checklist

Number	Done?	Checklist Item
1		Use download cables for device configuration.

The Intel Quartus Prime programmer supports configuration of the Intel Stratix 10 devices directly using JTAG interfaces with Intel programming download cables. You can download design changes directly to the device with Intel download cables, making prototyping easy and enabling you to make multiple design iterations in quick succession. You can use the same download cable to program configuration devices on the board and use JTAG debugging tools such as the Signal Tap Embedded Logic Analyzer.

Related Information

- [Intel FPGA Download Cable II User Guide](#)
Provides more information about how to use the Intel FPGA download cables.
- [Intel FPGA Ethernet Cable II User Guide](#)
Provides more information about how to use the Intel FPGA download cables.

Configuration Features

Table 20. Configuration Features Checklist

Number	Done?	Checklist Item
1		Ensure your configuration scheme and board support the required features: design security, remote upgrades, single event upset (SEU) mitigation.

This section describes Intel Stratix 10 configuration features and how they affect your design process.

Configuration Bitstream Compression

Configuration bitstream compression is always enabled in Intel Stratix 10 configuration. The Intel Quartus Prime software generates configuration files with compressed configuration data. This compressed file reduces the storage requirements in the configuration device or flash memory, and decreases the time required to transmit the configuration bitstream to the Intel Stratix 10 device.

Due to compressed configuration bitstream, passive configuration schemes for example Avalon-ST x8, x16, and x32 might require the external configuration host to pause sending configuration data by deasserting the `AVST_READY` signal.

Design Security Using Configuration Bitstream Encryption

The design security feature ensures that Intel Stratix 10 designs are protected from copying, reverse engineering, and tampering. Intel Stratix 10 devices have the ability to decrypt configuration bitstreams using the AES algorithm, an industry standard encryption algorithm that is FIPS-197 certified. Intel Stratix 10 devices have a design security feature which utilizes a 256-bit security key.

The design security feature is available in the all configuration schemes supported in the Intel Stratix 10 devices.

SEU Mitigation

Dedicated circuitry is built into Intel Stratix 10 devices for error detection and correction. When enabled, this feature checks for SEUs continuously and automatically. This allows you to confirm that the configuration data stored in an Intel Stratix 10 device is correct and alerts the system to a configuration error.

When using the SEU mitigation features, an SDM pin is used to implement the SEU_ERROR function. This pin flags errors for your system to take appropriate actions. Prior to compiling your design, enable the SEU_ERROR function and select an unused SDM pin to implement the SEU_ERROR function in the Intel Quartus Prime software.

Related Information

- [Configuration Features, Intel Stratix 10 Configuration User Guide](#)
Provides more information about the configuration features.
- [Intel Stratix 10 SEU Mitigation User Guide](#)
Provides more information about the SEU mitigation feature.

Intel Quartus Prime Configuration Settings

Table 21. Intel Quartus Prime Configuration Settings Checklist

Number	Done?	Checklist Item
1		Consider the Intel Quartus Prime configuration options when you plan your board and system design.

There are several configuration options that you can set in the Intel Quartus Prime Standard Edition software before compilation to generate configuration or programming files. Your board and system design are affected by these settings and pins, so consider them in the planning stages. Set the options on the General category of the **Device and Pin Options** dialog box.

Optional Configuration Pins

Table 22. Optional Configuration Pins Checklist

Number	Done?	Checklist Item
1		Plan the board design to support optional configuration pins as required.

You can enable the following optional configuration pins:

- OSC_CLK_1—Must be connected to a 25 MHz, 100 MHz, or 125 MHz source if used.
- CONF_DONE
- INIT_DONE

Intel Stratix 10 devices use OSC_CLK_1 pin as the reference clock for transceiver calibration. You must provide a stable and free running clock input at this pin.

Related Information

- [Device Configuration Pins, Intel Stratix 10 Configuration User Guide](#)
Provides more information about the configuration pins.
- [Calibration chapter, Intel Stratix 10 Transceiver PHY User Guide](#)

Dual Purpose Configuration Pins

Table 23. Dual Purpose Configuration Pins Checklist

Number	Done?	Checklist Item
1		Plan the dual purpose pins that can function as configuration pins and user I/O pins.

The below configuration pins used for the Avalon-ST ×16 and ×32 configuration schemes can optionally be used as user I/O pins after configuration has completed. Enable the pins to function as dual purpose pins in the Intel Quartus Prime software prior to compilation, if desired.

- AVST_CLK
- AVST_VALID
- AVST_DATA[15 : 0]
- AVST_DATA[31 : 16]—for Avalon-ST ×32 configuration scheme

Related Information

[Device Configuration Pins, Intel Stratix 10 Configuration User Guide](#)

Provides more information about the configuration pins.

Planning for On-Chip Debugging

On-chip debugging is an optional step in the design flow, and different debugging tools work better for different systems and different designers. Evaluate on-chip debugging options early in your design process to ensure that your system board, Intel Quartus Prime project, and design are able to support the appropriate options. Planning can reduce time spent debugging, and eliminates design changes later to accommodate your preferred debugging methodologies. Adding debug pins might not be enough, because of internal signal accessibility and I/O pin accessibility on the device. First, select your preferred debugging tools.

Related Information

- [On-Chip Debugging Tools](#) on page 20
- [Planning Guidelines for Debugging Tools](#) on page 22

On-Chip Debugging Tools

Table 24. On-Chip Debugging Tools Checklist

Number	Done?	Checklist Item
1		Take advantage of on-chip debugging features to analyze internal signals and perform advanced debugging techniques.

The Intel Quartus Prime portfolio of verification tools includes the following in-system debugging features:

- **Signal Probe incremental routing**—Quickly routes internal signals to I/O pins without affecting the routing of the original design. Starting with a fully routed design, you can select and route signals for debugging to either previously reserved or currently unused I/O pins.
- **Signal Tap Embedded Logic Analyzer**—Probes the state of internal and I/O signals without the use of external equipment or extra I/O pins, while the design is running at full speed in an FPGA device. Defining custom trigger-condition logic provides greater accuracy and improves the ability to isolate problems. It does not require external probes or changes to the design files to capture the state of the internal nodes or I/O pins in the design; all captured signal data is stored in the device memory until you are ready to read and analyze the data. The Signal Tap Embedded Logic Analyzer works best for synchronous interfaces. For debugging asynchronous interfaces, consider using Signal Probe or an external logic analyzer to view the signals more accurately. Signal Tap may affect routing of the original design.
- **Logic Analyzer Interface**—Enables you to connect and transmit internal FPGA signals to an external logic analyzer for analysis, allowing you to take advantage of advanced features in your external logic analyzer or mixed signal oscilloscope. You can use this feature to connect a large set of internal device signals to a small number of output pins for debugging purposes and it can multiplex signals with design I/O pins if required.
- **In-System Memory Content Editor**—Provides read and write access to in-system FPGA memories and constants through the JTAG interface, so you can test changes to memory content and constant values in the FPGA while the device is functioning in the system.
- **In-System Sources and Probes**—Sets up custom register chains to drive or sample the instrumented nodes in your logic design, providing an easy way to input simple virtual stimuli and capture the current value of instrumented nodes.

- Virtual JTAG Intel FPGA IP core—Enables you to build your own system-level debugging infrastructure, including both processor-based debugging solutions and debugging tools in the software for system-level debugging. You can instantiate the SLD_VIRTUAL_JTAG Intel FPGA IP core directly in your HDL code to provide one or more transparent communication channels to access parts of your FPGA design using the JTAG interface of the device.
- EMIF Debug toolkit—Tcl-based GUI communicating via a JTAG connection to enable external memory interface on the circuit board to retrieve calibration status and debug information. The Driver Margining feature of the tool kit allows you to measure margins on your memory interface using a driver with arbitrary traffic patterns. Tcl-based graphical user interface that provides access to memory calibration data gathered by the Nios II sequencer, via a JTAG connection. The Toolkit allows you to mask ranks for calibration, and to request recalibration of the interface. The Driver Margining feature of the toolkit allows you to measure margins on the memory interface using a driver with arbitrary traffic patterns. The EMIF Toolkit can communicate with several different memory interfaces on the same device, but only one at a time.
- Transceiver Toolkit—Uses System Console technology to help FPGA and board designers validate transceiver link signal integrity real time in a system and improve board bring-up time. Test for bit-error rate (BER) while simultaneously running multiple links at your target data rate to validate your board design with Transceiver Toolkit. Tune transceiver analog settings for optimal link performance while using different test metrics to quantify results. Simultaneously test multiple devices across one or more boards using link tests in the Transceiver Toolkit GUI.

Related Information

- [Virtual JTAG \(altera_virtual_jtag\) Intel FPGA IP Core User Guide](#)
Provides more information about the debugging tool.
- [System Debugging Tools Overview, Debug Tools User Guide \(Intel Quartus Prime Pro Edition\)](#)
Provides more information about choosing a debugging solution.

Planning Guidelines for Debugging Tools

Table 25. Planning Guidelines for Debugging Tools Checklist

Number	Done?	Checklist Item
1		Select on-chip debugging schemes early to plan memory and logic requirements, I/O pin connections, and board connections.
2		If you want to use Signal Probe incremental routing, the Signal Tap Embedded Logic Analyzer, Logic Analyzer Interface, In-System Memory Content Editor, In-System Sources and Probes, or Virtual JTAG IP core, plan your system and board with JTAG connections that are available for debugging.
3		Plan for the small amount of additional logic resources used to implement the JTAG hub logic for JTAG debugging features.
4		For debugging with the Signal Tap Embedded Logic Analyzer, reserve device memory resources to capture data during system operation.
5		Reserve I/O pins for debugging with Signal Probe or the Logic Analyzer Interface so you do not have to change the design or board to accommodate debugging signals later.
6		Ensure the board supports a debugging mode where debugging signals do not affect system operation.
<i>continued...</i>		

Number	Done?	Checklist Item
7		Incorporate a pin header or mictor connector as required for an external logic analyzer or mixed signal oscilloscope.
8		To use debug tools incrementally and reduce compilation time, ensure incremental compilation is on so you do not have to recompile the design to modify the debug tool.
9		To use the Virtual JTAG IP core for custom debugging applications, instantiate it in the HDL code as part of the design process.
10		To use the In-System Sources and Probes feature, instantiate the IP core in the HDL code.
11		To use the In-System Memory Content Editor for RAM or ROM blocks, turn on the Allow In-System Memory Content Editor to capture and update content independently of the system clock option for the memory block in the IP catalog.

If you intend to use any of the on-chip debugging tools, plan for the tool(s) when developing the system board, Intel Quartus Prime project, and design.

Pin Connection Considerations for Board Design

When designing the interfaces to the Intel Stratix 10 device, various factors can affect the PCB design.

Related Information

- [I/O and Clock Planning](#) on page 33
Provides more information about the I/O signal connections for the FPGA, which also affect the board design.
- [Intel Stratix 10 Device Family Pin Connection Guidelines](#)

Device Power-Up

Table 26. Device Power-Up Checklist

Number	Done?	Checklist Item
1		Design board for power-up: All Intel Stratix 10 GPIO pins are tri-stated until the device is configured and configuration pins drive out. The transceiver pins are at high impedance before the device periphery could get programmed. Once the periphery is programmed, the termination and V_{cm} are set immediately after transceiver calibration is complete.
2		Design voltage power supply ramps to be monotonic.
3		Set POR time to ensure power supplies are stable.
4		Design power sequencing and voltage regulators for best device reliability. Connect the GND between boards before connecting the power supplies.

The minimum current requirement for the power-on-reset (POR) supplies must be available during device power-up.

The Intel Stratix 10 device has Power-On Reset Circuitry, which keeps the device in a reset state until the power supply outputs are within the recommended operating range. The device must reach the recommended operating range within the maximum power supply ramp time. If the ramp time is not met, the device I/O pins and programming registers remain tri-stated and device configuration fails. For the Intel Stratix 10 device to exit POR, you must power the V_{CCBAT} power supply even if you do not use the volatile key.

In Intel Stratix 10 devices, a pin-selectable option (MSEL) allows you to select between a typical POR time setting of 4 ms or 100 ms. In both cases, you can extend the POR time by using an external component to assert the `nSTATUS` pin low. Extend POR time if the board cannot meet the maximum power ramp time specifications to ensure the device configures properly and enters user mode.

Intel Stratix 10 devices have power-up sequencing and power-down sequencing requirements. You should consider the power-up timing and power-down timing for each rail in order to meet the power sequencing requirements.

Intel uses GND as a reference for I/O buffer designs. Connecting the GND between boards before connecting the power supplies prevents the GND on your board from being pulled up inadvertently by a path to power through other components on your board. A pulled-up GND could otherwise cause an out-of-specification I/O voltage or current condition with the Intel device.

Related Information

[Power Sequencing Considerations for Intel Stratix 10 Devices](#)

Provides more information about power-up and power-down sequences.

Power Pin Connections and Power Supplies

Table 27. Power Pin Connections and Power Supplies Checklist

Number	Done?	Checklist Item
1		Connect all power pins correctly as specified in the <i>Intel Stratix 10 Device Family Pin Connection Guidelines</i> .
2		Connect <code>VCCIO</code> pins and <code>VREF</code> pins to support each bank I/O standards.
3		Explore unique requirements for FPGA power pins or other power pins on your board, and determine which devices on your board can share a power rail.
4		Follow the suggested power supply sharing and isolation guidance, and the specific guidelines for each pin in the <i>Intel Stratix 10 Device Family Pin Connection Guidelines</i> .
5		Refer to <i>AN 692: Power Sequencing Considerations for Intel Arria® 10 and Intel Stratix 10 Devices</i> to understand the power sequencing design requirements.
6		For SmartVID devices (-1V, -2V, and -3V speed grade devices), you must use PMBus-compliant voltage regulator to supply the <code>VCC</code> and <code>VCCP</code> pins. The recommended PMBus-compliant voltage regulator is LTM4677. For more details, refer to the <i>Intel Stratix 10 Power Management User Guide</i> .

Intel Stratix 10 devices require various voltage supplies depending on your design requirements.

Intel Stratix 10 devices support a wide range of industry I/O standards. The device output pins do not meet the I/O standard specifications if the `VCCIO` level is out of the recommended operating range for the I/O standard.

Voltage reference (`VREF`) pins serve as voltage references for certain I/O standards. The `VREF` pin is used mainly for a voltage bias and does not source or sink much current. The voltage can be created with a regulator or a resistor divider network.

The `VREFP_ADC` pin is not a power supply pin. It provides the reference voltage for the ADC for the voltage sensor. For better voltage sensor performance, connect the `VREFP_ADC` pin to an external reference 1.25 V source. Connecting the `VREFP_ADC` pin to GND activates an on-chip reference source.

Related Information

- [Recommended Operating Conditions, Intel Stratix 10 Device Datasheet](#)
Provides the supply voltages required for the Intel Stratix 10 devices and their recommended operation conditions.
- [I/O Standards Voltage Support, Intel Stratix 10 General Purpose I/O User Guide](#)
Provides the complete list of the supported I/O standards and V_{CCIO} voltages.
- [Intel Stratix 10 Device Family Pin Connection Guidelines](#)
- [Selectable Standards and Flexible I/O Banks](#) on page 37
Provides more information about V_{CCIO} voltages and V_{REF} pins for different I/O banks.
- [AN 692: Power Sequencing Considerations for Intel Cyclone® 10 GX, Intel Arria 10, and Intel Stratix 10 Devices](#)
Provides the power sequencing design requirements.
- [Intel Stratix 10 Power Management User Guide](#)
Provides more information about power management and VID parameters.

Decoupling Capacitors

Table 28. Decoupling Capacitors Checklist

Number	Done?	Checklist Item
1		Use the PDN tool to plan your power distribution netlist and decoupling capacitors.

Board decoupling is important for improving overall power supply integrity while ensuring the rated device performance.

Intel Stratix 10 devices include on-die decoupling capacitors to provide high-frequency decoupling. These low-inductance capacitors suppress power noise for excellent power integrity performance, and reduce the number of external PCB decoupling capacitors, saving board space, reducing cost, and greatly simplifying PCB design.

Intel has created an easy-to-use power distribution network (PDN) design tool that optimizes the board-level PDN graphically. The purpose of the board-level PDN is to distribute power and return currents from the voltage regulating module (VRM) to the FPGA power supplies. By using the PDN tool, you can quickly arrive at an optimized PDN decoupling solution for your specific design.

For each power supply, PDN designers must choose a network of bulk and decoupling capacitors. While SPICE simulation could be used to simulate the circuit, the PDN design tool provides a fast, accurate, and interactive way to determine the right number of decoupling capacitors for optimal cost and performance trade-offs.

Related Information

- [Power Distribution Network page](#)
Provides the PDN tool and user guide.

PLL Board Design Guidelines

Table 29. PLL Board Design Guidelines Checklist

Number	Done?	Checklist Item
1		Connect all PLL power pins to reduce noise even if the design does not use all the PLLs.
2		Power supply nets should be provided by an isolated power plane, a power plane cut out, or thick trace of at least 20 mils.

Plan your board design when you design a power system for PLL usage and to minimize jitter, because PLLs contain analog components embedded in a digital device.

Related Information

- [Clock and PLL Selection](#) on page 40
- [PLL Feature Guidelines](#) on page 41
- [Intel Stratix 10 Device Family Pin Connection Guidelines](#)
- [Support Resources : Board Design](#)
Provides more board design guidelines related to PLL power supplies in the “General Board Design Considerations/Guidelines” section.

Transceiver Board Design Guidelines

Table 30. Transceiver Board Design Guidelines Checklist

Number	Done?	Checklist Item
1		Review the transceiver board design guidelines when designing your board.

Related Information

- [Intel Stratix 10 Transceiver PHY User Guide](#)
Provides more information about the transceiver board design guidelines.
- [Intel Stratix 10 E-Tile Transceiver PHY User Guide](#)
- [Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide](#)
- [AN 766: Intel Stratix 10 Devices, High Speed Signal Interface Layout Design Guideline](#)
- [AN 778: Intel Stratix 10 Transceiver Usage](#)
Provides more information about the transceiver channel placement.
- [Intel Stratix 10 Device Datasheet](#)
Provides more information about the transceiver specifications.

Configuration Pin Connections

Table 31. Configuration Pin Connections Checklist

Number	Done?	Checklist Item
1		Check that all configuration pin connections and pull-up/pull-down resistors are set correctly for your configuration schemes.

Depending on your configuration scheme, different pull-up/pull-down resistor or signal integrity requirements might apply. Some configuration pins also have specific requirements if unused. It is very important to connect the configuration pins correctly. The following guidelines address the common issues.

Related Information

- [Intel Stratix 10 Device Family Pin Connection Guidelines](#)
- [Device Configuration Pins, Intel Stratix 10 Configuration User Guide](#)
Provides a list of the dedicated and dual-purpose configuration pins.

Configuration Pin Voltage Level

Table 32. Configuration Pin Voltage Level Checklist

Number	Done?	Checklist Item
1		Ensure V_{CCIO_SDM} and V_{CCIO} of the configuration pins match the voltage level of the external devices used for configuration. When using the Avalon-ST $\times 32$ or $\times 16$ configuration scheme, V_{CCIO} of the I/O bank which the $AVST_CLK$, $AVST_VALID$, and $AVST_DATA$ pins are located in, must match the V_{CCIO_SDM} level.

Configuration pins from the Intel Stratix 10 device connect to external devices, for example the serial configuration device or Avalon-ST host. The voltage level of the configuration pins need to match the voltage level of the devices connected to them. The JTAG and SDM I/Os used as configuration pins are powered by the V_{CCIO_SDM} supply. For Avalon-ST $\times 32$ and $\times 16$ configuration schemes, the $AVST_CLK$, $AVST_VALID$, and $AVST_DATA$ pins are powered by the V_{CCIO} of the I/O bank in which the pins reside in.

Clock Trace Signal Integrity

Table 33. Clock Trace Signal Integrity Checklist

Number	Done?	Checklist Item
1		Design configuration clock traces to be noise-free.

Board trace for clocks used in configuration, for example TCK , AS_CLK , $AVST\times 8_CLK$, $AVST_CLK$, and OSC_CLK_1 clock input, should produce clean signals with no overshoot, undershoot, or ringing. When designing the board, lay out the configuration clock traces with the same techniques used to lay out a clock line. Any overshoot, undershoot, ringing, or other noise on the clock signal can cause configuration failure. Make sure to have clock routing as stripline. Keep the clock routing away from any high-speed signals to isolate the clock signals from other signals.

JTAG Pins

Table 34. JTAG Pins Checklist

Number	Done?	Checklist Item
1		Connect JTAG pins to a stable voltage level if not in use.

Because JTAG configuration takes precedence over all other configuration methods, the JTAG pins should not be left floating or toggling during configuration if you do not use the JTAG interface. If you are using the JTAG interface, adhere to the following guidelines.

JTAG Pin Connections

Table 35. JTAG Pin Connections Checklist

Number	Done?	Checklist Item
1		Connect JTAG pins correctly to the download cable header. Ensure the pin order is not reversed.
2		To disable the JTAG state machine during power-up, pull the TCK pin low through a resistor to ensure that an unexpected rising edge does not occur on the TCK pin.
3		Pull the TMS and TDI pins high through a resistor.

A device operating in JTAG mode uses four required pins—TDI, TDO, TMS, and TCK. The TCK pin has an internal weak pull-down resistor, while the TDI and TMS pins have weak internal pull-up resistors.

If you have more than one device in the chain, connect the TDO pin of a device to the TDI pin of the next device in the chain.

Noise on the JTAG pins during configuration, user mode, or power-up can cause the device to go into an undefined state or mode.

Download Cable Operating Voltage

Table 36. Download Cable Operating Voltage Checklist

Number	Done?	Checklist Item
1		Ensure the download cable and JTAG pin voltages are compatible because the download cable interfaces with the JTAG pins of your device.

The operating voltage supplied to the Intel download cable by the target board through the 10-pin header determines the operating voltage level of the download cable.

JTAG pins in the Intel Stratix 10 device are powered up by V_{CCIO_SDM} . In a JTAG chain containing devices with different V_{CCIO} levels, ensure that the $V_{IL\ max}$, $V_{IH\ min}$, and the maximum V_I specifications of the device JTAG input pins are not violated. Level shifter might be required between devices to meet the voltage specifications of the devices input pin.

JTAG Signal Buffering

Table 37. JTAG Signal Buffering Checklist

Number	Done?	Checklist Item
1		Buffer JTAG signals per the recommendations, especially for connectors or if the cable drives more than three devices.
2		If your device is in a configuration chain, ensure all devices in the chain are connected properly.

You might have to add buffers to a JTAG chain, depending on the JTAG signal integrity, especially the TCK signal, because it is the JTAG clock and the fastest switching JTAG signal. Intel recommends buffering the signals at the connector because cables and board connectors tend to make bad transmission lines and introduce noise to the signals. After this initial buffer at the connector, add buffers as the chain gets longer or whenever the signals cross a board connector.

If a cable drives three or more devices, buffer the JTAG signal at the cable connector to prevent signal deterioration. This also depends on the board layout, loads, connectors, jumpers, and switches on the board. Anything added to the board that affects the inductance or capacitance of the JTAG signals increases the likelihood that a buffer should be added to the chain.

Each buffer should drive no more than eight loads for the TCK and TMS signals, which drive in parallel. If jumpers or switches are added to the path, decrease the number of loads.

MSEL Configuration Mode Pins

Table 38. MSEL Configuration Mode Pins Checklist

Number	Done?	Checklist Item
1		Connect the SDM pins with MSEL function to select the configuration scheme; do not leave them floating. Do not hardwire the pins to V _{CCIO_SDM} or GND if they have other configuration functions based on the configuration scheme selected.

Select the configuration scheme by pulling the SDM pins with MSEL function high or low with external resistors. JTAG configuration is always available, regardless of the MSEL settings. The SDM pins with MSEL function are powered by the V_{CCIO_SDM} power supply, and they have internal weak pull-up resistors.

During POR and reconfiguration, the SDM pins with MSEL function must be at LVTTL V_{IL} and V_{IH} levels to be considered as logic low and logic high, respectively. The SDM pins used for MSEL function also have other configuration functions, depending on the configuration schemes used. Do not hardwire the SDM pins with MSEL function to V_{CCIO_SDM} or GND without pull-up or pull-down resistors, if they are needed for the configuration scheme used.

Other Configuration Pins

Table 39. Other Configuration Pins Checklist

Number	Done?	Checklist Item
1		Use the SDM pins which have multiple configuration functions if power management function is required.
2		When a -V device is used, you must enable the SmartVID connection between the device and the VCC voltage regulator to allow the FPGA to directly control its core voltage requirements. Refer to the <i>Intel Stratix 10 Device Family Pin Connection Guidelines</i> and <i>Intel Stratix 10 Power Management User Guide</i> for the pin connections and implementation.

Most of the SDM pins have multiple configuration functions, depending on the configuration schemes used. Some SDM pins also have power management functions. If power management function is required, choose the SDM pins which do not need to be used for configuration to implement the power management function.

Connect the SDM pins on your board to the external configuration host or configuration device based on the configuration scheme to be used. If more than one configuration scheme are used, ensure there is no contention between configuration host or configuration devices connected to the SDM pins.

Related Information

- [Secure Device Manager \(SDM\) Pins, Intel Stratix 10 Device Family Pin Connection Guidelines](#)
Provides the SmartVID connection between the device and the VCC voltage regulator.
- [SmartVID, Intel Stratix 10 Power Management User Guide](#)
Provides more information about the SmartVID feature implementation.

Board-Related Intel Quartus Prime Settings

Table 40. Board-Related Intel Quartus Prime Settings Checklist

Number	Done?	Checklist Item
1		Set the settings for the FPGA I/O pins correctly and plan for the functionality during board design.

The Intel Quartus Prime software provides options for the FPGA I/O pins that you should consider during board design. Ensure that these options are set correctly when the Intel Quartus Prime project is created, and plan for the functionality during board design.

Unused Pins

Table 41. Unused Pins Checklist

Number	Done?	Checklist Item
1		Specify the reserved state for unused I/O pins.
2		Carefully check the pin connections in the Intel Quartus Prime software-generated .pin file. Do not connect RESERVED pins.

You can specify the state of unused pins in the Intel Quartus Prime software to allow flexibility in the board design by choosing one of the five allowable states for **Reserve all unused pins** on the **Unused Pins** category in the **Device and Pin Options** dialog box:

- **As inputs tri-stated**
- **As output driving ground**
- **As outputs driving an unspecified signal**
- **As input tri-stated with bus-hold circuitry**
- **As input tri-stated with weak pull-up**

The common setting is to set unused pins **As inputs tri-stated with weak pull-up**. To improve signal integrity, set the unused pins to **As output driving ground**. Doing this reduces inductance by creating a shorter return path and reduces noise on the neighboring I/Os. This approach should not be used if this results in many via paths causing congestion for signals under the device.

To reduce power dissipation, set clock pins and other unused I/O pins **As inputs tri-stated**, and tie them to ground.

Signal Integrity Considerations

Signal integrity considerations include detailed board design guidelines, as well as a few guidelines related to V_{REF} pins, SSN, and I/O termination.

High-Speed Board Design

Table 42. High-Speed Board Design Checklist

Number	Done?	Checklist Item
1		Refer to the Board Design Resource Center.

If your design has high-speed signals, especially with Intel Stratix 10 GX/SX device high-speed transceivers, the board design has a major impact on the signal integrity in the system.

Related Information

- [Support Resources : Board Design](#)
Provides more information about signal integrity and board design.
- [AN 528: PCB Dielectric Material Selection and Fiber Weave Effect on High-Speed Channel Routing](#)
Provides more information about high-speed board stack-up and signal routing layers.
- [AN 529: Via Optimization Techniques for High-Speed Channel Designs](#)
Provides more information about high-speed board stack-up and signal routing layers.
- [AN 530: Optimizing Impedance Discontinuity Caused by Surface Mount Pads for High-Speed Channel Designs](#)
Provides more information about high-speed board stack-up and signal routing layers.
- [AN 766: Intel Stratix 10 Devices, High Speed Signal Interface Layout Design Guideline](#)

Voltage Reference Pins

Table 43. Voltage Reference Pins Checklist

Number	Done?	Checklist Item
1		Design V_{REF} pins to be noise-free.

Voltage deviation on a V_{REF} pin can affect the threshold sensitivity for inputs.

Related Information

[Guideline: \$V_{REF}\$ Sources and \$V_{REF}\$ Pins, Intel Stratix 10 General Purpose I/O User Guide](#)

Provides more information about the V_{REF} pins and I/O standards.

Simultaneous Switching Noise

Table 44. Simultaneous Switching Noise Checklist

Number	Done?	Checklist Item
1		Break out large bus signals on board layers close to the device to reduce cross talk.
2		Route traces orthogonally if two signal layers are next to each other, if possible. Use a separation of two to three times the trace width.

SSN is a concern when too many pins (in close proximity) change voltage levels at the same time. Noise generated by SSN can reduce the noise margin and cause incorrect switching. Although SSN is dominant on the device package, plan the board layout according to the board layout recommendations in the PCB guidelines can help with noise reduction.

Related Information

Support Resources : Board Design

Provides more information about the PCB guidelines and board layout recommendations that can help with noise reduction.

I/O Termination

Table 45. I/O Termination Checklist

Number	Done?	Checklist Item
1		Check I/O termination and impedance matching for chosen I/O standards, especially for voltage-referenced standards.

Voltage-referenced I/O standards require both an V_{REF} and a termination voltage (V_{TT}). The reference voltage of the receiving device tracks the termination voltage of the transmitting device. Each voltage-referenced I/O standard requires a unique termination setup.

Although single-ended, non-voltage-referenced I/O standards do not require termination, impedance matching is necessary to reduce reflections and improve signal integrity.

Intel Stratix 10 on-chip series and parallel termination provides the convenience of no external components. Alternatively, you can use external pull-up resistors to terminate the voltage-referenced I/O standards such as SSTL and HSTL.

Differential I/O standards typically require a termination resistor between the two signals at the receiver. The termination resistor must match the differential load impedance of the signal line. Intel Stratix 10 devices provide an optional on-chip differential resistor when using LVDS.

Related Information

On-Chip I/O Termination in Devices, Intel Stratix 10 General Purpose I/O User Guide

Provides a complete list of OCT support for each I/O standard, and more information about the OCT features and limitations.

Board-Level Simulation and Advanced I/O Timing Analysis

Table 46. Board-Level Simulation and Advanced I/O Timing Analysis Checklist

Number	Done?	Checklist Item
1		Perform board-level simulation using IBIS models (when available).
2		Configure board trace models for Intel Quartus Prime advanced I/O timing analysis.

To ensure that the I/O signaling meets receiver threshold levels on your board setup, perform full board routing simulation with third-party board-level simulation tools using an IBIS model.

When this feature is available in the Intel Quartus Prime software, select **IBIS** under **Board-level signal integrity analysis** on the **Board-Level** page in **EDA Tool Settings** of the **Settings** dialog box.

When you include an FPGA device with high-speed interfaces in a board design, knowing the signal integrity and board routing propagation delay is vital for proper system operation. You should analyze board level timing as part of the I/O and board planning, especially for high-speed designs.

You can configure board trace models of selected I/O standards and generate “board-aware” signal integrity reports with the Intel Quartus Prime software. When **Enable Advanced I/O Timing** is turned on (**Timing Analyzer** page in the **Settings** dialog box), the Timing Analyzer uses simulation results for the I/O buffer, package, and the board trace model to generate more accurate I/O delays and extra reports to give insight into signal behavior at the system level. You can use these advanced timing reports as a guide to make changes to the I/O assignments and board design to improve timing and signal integrity.

Related Information

[Signal Integrity Analysis with Third-Party Tools](#) chapter, *PCB Design Tools User Guide (Intel Quartus Prime Pro Edition)*

Provides more information about board-level simulation using IBIS models.

I/O and Clock Planning

Planning and allocating I/O and clock resources is an important task with the high pin counts and advanced clock management features in Intel Stratix 10 devices. Various considerations are important to effectively plan the available I/O resources to maximize utilization and prevent issues related to signal integrity. Good clock management systems are also crucial to the performance of an FPGA design.

The I/O and clock connections of your FPGA affect the rest of your system and board design, so it is important to plan these connections early in your design cycle.

Making FPGA Pin Assignments

Table 47. Making FPGA Pin Assignments Checklist

Number	Done?	Checklist Item
1		Use the Intel Quartus Prime Pin Planner to make pin assignments.
2		Use Intel Quartus Prime Fitter messages and reports for sign-off of pin assignments.
3		Verify that the Intel Quartus Prime pin assignments match those in the schematic and board layout tools.
4		Plan interfaces and device periphery using the Interface Planner. After design synthesis, use the Interface Planner to rapidly define a legal device floorplan. Planning using the Interface Planner involves initialization of the Interface Planner, reconciliation of project assignments, placement of periphery elements and clocks, and export of plan constraints to your Intel Quartus Prime project.

With the Intel Quartus Prime Pin Planner GUI, you can identify I/O banks, V_{REF} groups, and differential pin pairings to help you through the I/O planning process. Right-click in the Pin Planner spreadsheet interface and click the **Pin Finder** to search for specific pins. If migration devices are selected, the Pin Migration view highlights pins that change function in the migration device when compared to the currently selected device.

You have the option of importing a Microsoft Excel spreadsheet into the Intel Quartus Prime software to start the I/O planning process if you normally use a spreadsheet in your design flow. You can also export a spreadsheet compatible (.csv) file containing your I/O assignments when all pins are assigned.

When you compile your design in the Intel Quartus Prime software, I/O Assignment Analysis in the Fitter validates that the assignments meet all the device requirements and generates messages if there are any problems.

Intel Quartus Prime designers can then pass the pin location information to PCB designers. Pin assignments between the Intel Quartus Prime software and your schematic and board layout tools must match to ensure the design works correctly on the board where it is placed, especially if changes to the pin-out must be made. The Pin Planner is integrated with certain PCB design EDA tools and can read pin location changes from these tools to check the suggested changes. When you compile your design, the Intel Quartus Prime software generates the .pin file. You can use this file to verify that each pin is correctly connected in the board schematics.

Related Information

- [Vertical Device Migration](#) on page 10
Provides more information about migrating to a different density Intel Stratix 10 devices.
- [Managing Device I/O Pins](#) chapter, [Design Constraints User Guide \(Intel Quartus Prime Pro Edition\)](#)
Provides more details about using the Pin Planner to make I/O assignments.
- [Cadence PCB Design Tools Support](#) chapter, [PCB Design Tools User Guide \(Intel Quartus Prime Pro Edition\)](#)
Provides more information about passing I/O information between the Intel Quartus Prime software and third-party EDA tools.

- [Mentor Graphics* PCB Design Tools Support chapter, PCB Design Tools User Guide \(Intel Quartus Prime Pro Edition\)](#)
Provides more information about passing I/O information between the Intel Quartus Prime software and third-party EDA tools.
- [Interface Planning chapter, Design Constraints User Guide \(Intel Quartus Prime Pro Edition\)](#)

Early Pin Planning and I/O Assignment Analysis

Table 48. Early Pin Planning and I/O Assignment Analysis Checklist

Number	Done?	Checklist Item
1		Use the Create Top-Level Design File command with I/O Assignment Analysis to check the I/O assignments before the design is complete.

In many design environments, FPGA designers want to plan top-level FPGA I/O pins early so that board designers can start developing the PCB design and layout. The FPGA device I/O capabilities and board layout guidelines influence pin locations and other types of assignments. In cases where the board design team specifies an FPGA pin-out, it is crucial that you verify pin locations in the FPGA place-and-route software as soon as possible to avoid board design changes.

Starting FPGA pin planning early improves the confidence in early board layouts, reduces the chance of error, and improves the overall time to market for the design. You can create a preliminary pin-out for an Intel FPGA using the Intel Quartus Prime Pin Planner before the source code is designed.

Early in the design process, the system architect typically has information about the standard I/O interfaces (such as memory and bus interfaces), IP cores to be used in the design, and any other I/O-related assignments defined by system requirements.

The Pin Planner Create/Import IP Core feature interfaces with the IP catalog, and enables you to create or import custom IP cores that use I/O interfaces. Enter PLL and LVDS blocks, including options such as dynamic phase alignment (DPA), because options affect the pin placement rules. When you have entered as much I/O-related information as possible, generate a top-level design netlist file using the **Create Top-Level Design File** command in the Pin Planner. You can use the I/O analysis results to change pin assignments or IP parameters and repeat the checking process until the I/O interface meets your design requirements and passes the pin checks in the Intel Quartus Prime software.

When planning is complete, the preliminary pin location information can be passed to PCB designers. When the design is complete, use the reports and messages generated by the Intel Quartus Prime Fitter for the final sign-off of the pin assignments.

Related Information

- [Making FPGA Pin Assignments](#) on page 34
Provides more information about using the Intel Quartus Prime Pin Planner for I/O pin assignment planning, assignment, and validation.
- [Managing Device I/O Pins chapter, Design Constraints User Guide \(Intel Quartus Prime Pro Edition\)](#)
Provides more information about I/O assignment and analysis.

I/O Features and Pin Connections

Intel Stratix 10 I/O pins are designed for ease of use and rapid system integration, while simultaneously providing high bandwidth. Independent modular I/O banks with a common bank structure for vertical migration lend efficiency and flexibility to the high speed I/O.

The following guidelines provide information pertaining to I/O features and pin connections.

Related Information

[Intel Stratix 10 Device Family Pin Connection Guidelines](#)

Provides a list of I/O pin locations and connection guidelines.

I/O Signaling Type

Table 49. I/O Signaling Type Checklist

Number	Done?	Checklist Item
1		Plan the I/O signaling type based on the system requirements.
2		Allow the software to assign locations for the negative pin in differential pin pairs.

Intel Stratix 10 devices support a wide range of industry I/O standards, including single-ended, voltage-referenced single-ended, and differential I/O standards. Follow these general guidelines when you select a signaling type.

Single-ended I/O signaling provides a simple rail-to-rail interface. Its speed is limited by the large voltage swing and noise. Single-ended I/Os do not require termination, unless reflection in the system causes undesirable effects.

Voltage-referenced signaling reduces the effects of simultaneous switching outputs (SSO) from pins changing voltage levels at the same time (for example, external memory interface data and address buses). Voltage-referenced signaling also provides an improved logic transition rate with a reduced voltage swing, and minimizes noise caused by reflection with a termination requirement. However, additional termination components are required for the reference voltage source (V_{TT}).

Differential signaling eliminates the interface performance barrier of single-ended and voltage-referenced signaling, with superior speed using an additional inverted closely-coupled data pair. Differential signaling also avoids the requirement for a clean reference voltage. This is possible because of a lower swing voltage and noise immunity with a common mode noise rejection capability. Considerations for this implementation include the requirements for a dedicated PLL to generate a sampling clock, and matched trace lengths to eliminate the phase difference between an inverted and non-inverted pair.

Intel Stratix 10 I/O pins are organized in pairs to support differential standards. Each I/O pin pair can support differential input or output operations, with the exception of certain clock pins that support differential input operations only. In your design source code, define just one pin to represent a differential pair, and make a pin assignment for this positive end of the pair. When you specify a differential I/O standard, the Intel Quartus Prime software automatically places the corresponding negative pin.

Selectable Standards and Flexible I/O Banks

Table 50. Selectable Standards and Flexible I/O Banks Checklist

Number	Done?	Checklist Item
1		Select a suitable signaling type and I/O standard for each I/O pin. The I/O banks are located in I/O columns. Each I/O bank contains its own PLL, DPA, and SERDES circuitries.
2		Ensure that the appropriate I/O standard support is supported in the targeted I/O bank.
3		Place I/O pins that share voltage levels in the same I/O bank.
4		Verify that all output signals in each I/O bank are intended to drive out at the bank V_{CCIO} voltage level.
5		Verify that all voltage-referenced signals in each I/O bank are intended to use the bank V_{REF} voltage level.
6		Check the I/O bank support for LVDS and transceiver features.
7		If you are using the Intel Stratix 10 TX 400 (1ST040E) devices, be aware that banks 3A and 3D have restrictions. I/O pins in these banks do not support LVDS SERDES or EMIF. In addition, LVDS, mini-LVDS, and RSDS I/O standards are only supported by the dedicated clock pins in these banks.
8		If you are using the Intel Stratix 10 GX 400 (1SG040HF35) and SX 400 (1SX040HF35) devices, be aware that banks 3A, 3C, and 3D have restrictions. I/O pins in these banks do not support LVDS SERDES or EMIF. In addition, bank 3D is limited to a maximum of 30 pins with 1.8 V I/O standard support only. Bank 3C is limited to 3.0 V and 3.3 V I/O support only. Finally, LVDS, mini-LVDS, and RSDS I/O standards are only supported by the dedicated clock pins in banks 3A and 3D.

Intel Stratix 10 I/O pins are arranged in groups called modular I/O banks. Be sure to use the correct dedicated pin inputs for signals such as clocks and global control signals.

The board must supply each bank with one V_{CCIO} voltage level for every V_{CCIO} pin in a bank. Each I/O bank is powered by the V_{CCIO} pins of that particular bank, and is independent of the V_{CCIO} pins of other I/O banks. A single I/O bank supports output signals that are driving at the same voltage as the V_{CCIO} . An I/O bank can simultaneously support any number of input signals with different I/O standards.

To accommodate voltage-referenced I/O standards, each I/O bank supports multiple V_{REF} pins feeding a common V_{REF} bus. Set the V_{REF} pins to the correct voltage for the I/O standards in the bank. Each I/O bank can only have a single V_{CCIO} voltage level and a single V_{REF} voltage level at a given time. If the V_{REF} pins are not used as voltage references, they cannot be used as generic I/O pins and should be tied to V_{CCIO} of that same bank or GND.

An I/O bank including single-ended or differential standards can support voltage-referenced standards as long as all voltage-referenced standards use the same V_{REF} setting. Voltage-referenced bi-directional and output signals must drive out at the I/O bank V_{CCIO} voltage level.

Different I/O banks include different support for LVDS signaling, and the Intel Stratix 10 transceiver banks include additional support. There are two types of I/O banks, LVDS and 3 V.

The LVDS I/O bank supports differential and single-ended I/O standards up to 1.8 V. The LVDS I/O pins form pairs of true differential LVDS channels. Each pair supports a parallel input/output termination between the two pins. You can use each LVDS channel as transmitter or receiver.

The 3 V I/O bank supports only single-ended I/O standards up to 3 V. Each adjacent I/O pair also supports Differential SSTL and Differential HSTL I/O standards. The single-ended output of the 3 V I/O has the same set of features as the single-ended output of the DDR I/O IP, except the programmable pre-emphasis feature.

Related Information

- [Clock and PLL Selection](#) on page 40
- [I/O and Differential I/O Buffers, Intel Stratix 10 General Purpose I/O User Guide](#)
Provides more information about I/Os and the number of channels available for the LVDS I/O standard.
- [Guideline: I/O Standards Limitation for Intel Stratix 10 TX 400, Intel Stratix 10 General Purpose I/O User Guide](#)
Provides the limitation for I/O banks 3A and 3D for 1ST040E device.
- [Intel Stratix 10 Transceiver PHY Architecture chapter, Intel Stratix 10 Transceiver PHY User Guide](#)
Provides more information about the transceiver-bank-related features.
- [I/O Standard Specifications, Intel Stratix 10 Device Datasheet](#)
Provides the electrical characteristics of each I/O standard.

Memory Interfaces

Table 51. Memory Interfaces Checklist

Number	Done?	Checklist Item
1		Use the External Memory Interfaces Intel Stratix 10 FPGA IP core for each memory interface, and follow connection guidelines/restrictions in the appropriate documentation.
2		Always check the pin table for the actual locations of the DQS and DQ pins and the EMIF pin table for locations of address and control pins.
3		I/O pins in banks 3A and 3D for Intel Stratix 10 TX 400 (1ST040E) do not support external memory interfaces. Do not use these I/O banks in this device to implement your external memory interface.

Intel Stratix 10 devices provide an efficient architecture to quickly and easily fit wide external memory interfaces with their small modular I/O banks. Any I/O banks that do not support transceiver operations in Intel Stratix 10 devices support external memory interfaces. However, DQS (data strobe or data clock) and DQ (data) pins are listed for EMIF supported banks in the device pin tables and are fixed at specific locations in the device. You must adhere to these pin locations to optimize routing, minimize skew, and maximize margins. Always check the pin table for the actual locations of the DQS and DQ pins and the EMIF pin table for locations of address and control pins.

Note: Maximum interface width varies from device to device depending on the number of I/O pins and DQS or DQ groups available. Achievable interface width also depends on the number of address and command pins that the design requires. To ensure adequate PLL, clock, and device routing resources are available, you should always test fit any IP in the Intel Quartus Prime software before PCB sign-off.

The self-calibrating External Memory Interfaces IP core is optimized to take advantage of the Intel Stratix 10 I/O structure. The External Memory Interfaces IP core allows you to set external memory interface features and helps set up the physical interface (PHY) best suited for your system. When you use the Intel memory controller Intel FPGA IP functions, the External Memory Interfaces IP core is instantiated

automatically. If you design multiple memory interfaces into the device using Intel FPGA IP core, generate a unique interface for each instance to ensure good results instead of designing it once and instantiating it multiple times.

The data strobe DQS and data DQ pin locations are fixed in Intel Stratix 10 devices. Before you design your device pin-out, refer to the memory interface guidelines for details and important restrictions related to the connections for these and other memory-related signals.

You can implement a protocol that is not supported by External Memory Interfaces IP core by using the PHY Lite for Parallel Interfaces Intel Stratix 10 FPGA IP core.

Address and command pins within the address/command bank must follow a fixed pin-out scheme, as defined in the `<variation_name>_readme.txt` file generated with your IP core. The pin-out scheme varies according to the topology of the memory interface. The pin-out scheme is a hardware requirement that you must follow. Some schemes require three lanes to implement address and command pins, while others require four lanes.

Related Information

- [External Memory Interfaces IP - Support Center page](#)
Provides more information about the external memory interface (EMIF) support on designing and implementing an EMIF.
- [PHY Lite for Parallel Interfaces Intel FPGA IP Core User Guide](#)
Provides more information about the PHY Lite for Parallel Interfaces Intel Stratix 10 FPGA IP core.
- [Interface Pins, External Memory Interfaces Intel Stratix 10 FPGA IP User Guide](#)
Provides more information about the DQS and DQ pins.
- [Intel Stratix 10 Device Family Pin Connection Guidelines](#)
- [Pin-Out Files for Intel FPGA Devices](#)

Dual-Purpose and Special Pin Connections

Table 52. Dual-Purpose and Special Pin Connections Checklist

Number	Done?	Checklist Item
1		Make dual-purpose pin settings and check for any restrictions when using these pins as regular I/O.

Intel Stratix 10 devices allow I/O flexibility with dual-purpose configuration pins. You can use dual-purpose configuration pins as general I/Os after device configuration is complete. Select the desired setting for each of the dual-purpose pins on the **Dual-Purpose Pins** category of the **Device and Pin Options** dialog box. Depending on the configuration scheme, these pins can be reserved as regular I/O pins, as inputs that are tri-stated, as outputs that drive ground, or as outputs that drive an unspecified signal.

You can also use dedicated clock inputs, which drive the programmable clock routing networks, as general-purpose input pins if they are not used as clock pins. When you use the clock inputs as general inputs, I/O registers use ALM-based registers because the clock input pins do not include dedicated I/O registers.

The device-wide reset and clear pins are available as design I/Os if they are not enabled.

Intel Stratix 10 I/O Features

Table 53. Intel Stratix 10 I/O Features Checklist

Number	Done?	Checklist Item
1		Check available device I/O features that can help I/O interfaces: current strength, slew rate, I/O delays, open-drain, bus hold, programmable pull-up resistors, PCI* clamping diodes, programmable pre-emphasis, and V _{OD} .
2		Consider on-chip termination (OCT) features to save board space.
3		Verify that the required termination scheme is supported for all pin locations.
4		Choose the appropriate mode of DPA, non-DPA, or soft-CDR for high-speed LVDS interfaces.

The Intel Stratix 10 bi-directional I/O element (IOE) features support rapid system integration while simultaneously providing the high bandwidth required to maximize internal logic capabilities and system-level performance. Advanced features for device interfaces assist in high-speed data transfer into and out of the device and reduce the complexity and cost of the PCB.

Intel recommends performing an IBIS or SPICE simulations to optimize your design settings.

Related Information

- [Intel Stratix 10 I/O Architecture and Features, Intel Stratix 10 General Purpose I/O User Guide](#)
Provides more information about the Intel Stratix 10 I/O features and usage.
- [Intel Stratix 10 Device Datasheet](#)
Provides more information about the I/O Standard Specifications and I/O Timing.

Clock and PLL Selection

Table 54. Clock and PLL Selection Checklist

Number	Done?	Checklist Item
1		Use the correct dedicated clock pins and routing signals for clock and global control signals.
2		Use the device PLLs for clock management.
3		Analyze input and output routing connections for each PLL and clock pin. Ensure PLL inputs come from the dedicated clock pins or from another PLL.

The first stage in planning your clocking scheme is to determine your system clock requirements. Understand your device's available clock resources and correspondingly plan the design clocking scheme. Consider your requirements for timing performance, and how much logic is driven by a particular clock.

Intel Stratix 10 devices provide dedicated low-skew and high fan-out routing networks.

The dedicated clock pins drive the clock network directly, ensuring lower skew than other I/O pins. Use the dedicated routing network to have a predictable delay with less skew for high fan-out signals. You can also use the clock pins and clock network to drive control signals like asynchronous reset.

Connect clock inputs to specific PLLs to drive specific low-skew routing networks. Analyze the global resource availability for each PLL and the PLL availability for each clock input pin.

Intel Stratix 10 devices contain dedicated resources for distributing signals throughout the fabric with balanced delay. These resources are typically used for clock signals. You can also use these resources for other signals with low-skew requirements. In Intel Stratix 10 devices, these resources are implemented as a programmable clock routing, which allows for the implementation of low-skew clock networks of variable size.

If your system requires more clock or control signals than are available in the target device, consider cases where the dedicated clock resource could be spared, particularly low fan-out and low-frequency signals where clock delay and clock skew do not have a significant impact on the design performance. Use the **Global Signal** assignment in the Intel Quartus Prime Assignment Editor to select the type of global routing, or set the assignment to **Off** to specify that the signal should not use any global routing resources.

Related Information

[Intel Stratix 10 Clocking and PLL Architecture and Features chapter, Intel Stratix 10 Clocking and PLL User Guide](#)

Provides more information about clock resources, and clock and PLL features.

PLL Feature Guidelines

Table 55. PLL Feature Guidelines Checklist

Number	Done?	Checklist Item
1		Enable PLL features and check settings in the parameter editor.

Based on your system requirements, define the required clock frequencies for your FPGA design, and the input frequencies available to the FPGA. Use these specifications to determine your PLL scheme. Use the Intel Quartus Prime parameter editor to enter your settings in IOPLL Intel FPGA IP core, and check the results to verify whether particular features and input/output frequencies can be implemented in a particular PLL.

Intel Stratix 10 devices contain fractional PLLs in addition to I/O PLLs. You can configure fractional PLLs as integers or as enhanced fractional PLLs.

You can use I/O PLLs and fractional PLLs to reduce the number of oscillators required on the board, as well as to reduce the clock pins used in the FPGA by synthesizing multiple clock frequencies from a single reference clock source. In addition, you can use fractional PLLs for transmit clocking for transceivers.

Intel Stratix 10 device PLLs are feature rich, and support advanced capabilities such as clock feedback modes, switchover, and dynamic phase shifting.

Related Information

[PLLs Architecture and Features, Intel Stratix 10 Clocking and PLL User Guide](#)

Provides more information about the PLL features.

Clock Feedback Mode

Table 56. Clock Feedback Mode Checklist

Number	Done?	Checklist Item
1		Ensure you select the correct PLL feedback compensation mode.

Intel Stratix 10 PLLs support six different clock feedback modes.

Related Information

[Clock Feedback Modes, Intel Stratix 10 Clocking and PLL User Guide](#)
Provides more information about the clock feedback modes.

Clock Outputs

Table 57. Clock Outputs Checklist

Number	Done?	Checklist Item
1		Check that the PLL offers the required number of clock outputs and use dedicated clock output pins.

You can connect clock outputs to dedicated clock output pins or dedicated clock networks. There is no dedicated clock out pin for fractional PLL. I/O PLL can connect to a clock network or a dedicated clock pin.

Clock Control Features

Table 58. Clock Control Features Checklist

Number	Done?	Checklist Item
1		Use the clock control block for clock selection and power-down.

Intel Stratix 10 devices uses these clock control features: clock gating and clock divider. The clock from the I/O PLL output can be gated dynamically. These clock signals along with other clock sources go to the periphery distributed clock multiplexer (DCM). In the periphery DCM, the clock signal can either pass straight through, be gated by the root clock gate, or be divided by the clock divider.

Related Information

[Clock Control Features, Intel Stratix 10 Clocking and PLL User Guide](#)
Provides more information about the clock control features.

I/O Simultaneous Switching Noise

Table 59. I/O Simultaneous Switching Noise Checklist

Number	Done?	Checklist Item
1		Reduce the number of pins that switch the voltage level at exactly the same time whenever possible.
2		Use differential I/O standards and lower-voltage standards for high-switching I/Os.
3		Use lower drive strengths for high-switching I/Os. The default drive strength setting might be higher than your design requires.
<i>continued...</i>		

Number	Done?	Checklist Item
4		Reduce the number of simultaneously switching output pins within each bank. Spread output pins across multiple banks if possible.
5		Spread switching I/Os evenly throughout the bank to reduce the number of aggressors in a given area to reduce SSN (when bank usage is substantially below 100%).
6		Separate simultaneously switching pins from input pins that are susceptible to SSN.
7		Place important clock and asynchronous control signals near ground signals and away from large switching buses.
8		Avoid using I/O pins one or two pins away from PLL power supply pins for high-switching or high-drive strength pins.
9		Use staggered output delays to shift the output signals through time, or use adjustable slew rate settings.
10		Limit the number of unterminated SSO pins within the I/O bank where the PLL output clock resides to achieve the Intel Stratix 10 I/O PLL clock output jitter performance specification.

SSN is a concern when too many I/Os (in close proximity) change voltage levels at the same time. Plan the I/O and clock connections according to the recommendations.

Related Information

- [Programmable IOE Features in Intel Stratix 10 Devices, Intel Stratix 10 General Purpose I/O User Guide](#)
Provides more details about the programmable IOE features.
- [Intel Stratix 10 Clocking and PLL User Guide](#)
Provides the guidelines on the I/O PLL jitter performance.

Security Considerations

Table 60. Security Considerations Checklist

Number	Done?	Checklist Item
1		Consider whether your design requires device security features to be enabled. If so, you must provide power to the VCCFUSEWR_SDM rail for authentication fuse management.
2		Consider whether your design requires bitstream encryption, and whether the encryption keys are stored in Battery-Backed RAM (BBRAM). If so, plan to provide power to the VCCBAT pin using a battery on the board.
3		Consider licensing terms that best suit your requirements for the available device variants.

Intel Stratix 10 devices provide flexible and robust security features to help protect sensitive data, intellectual property, and the device itself under both remote and physical attacks. Intel Stratix 10 devices provide two main categories of security features:

- **Authentication**—Authentication helps ensure that the device firmware and optionally the configuration bitstream are from a trusted source. Device firmware authentication is always performed. Owner bitstream authentication must be enabled to use any other security features available on Intel Stratix 10 devices.
- **Encryption**—Encryption helps protect confidential information in the owner configuration bitstream and reduces the threat of intellectual property theft.

When designing a system with an Intel Stratix 10 device that utilizes device security features, you must consider provisions for enabling and managing the features throughout the expected operating lifetime of the device. To enable owner bitstream authentication, you must program an owner root key hash into eFuses. For devices with design security features enabled, Intel strongly recommends updating to the latest available device firmware and canceling old firmware IDs as necessary. Cancellation of firmware and device design IDs are managed in eFuses. Therefore, you must provide appropriate power to the VCCFUSEWR_SDM pin if you enable device security features. Devices with design security features enabled are not able to respond to security vulnerabilities if they cannot blow fuses. For more information about powering on VCCFUSEWR_SDM, refer to the *Intel Stratix 10 Device Family Pin Connection Guidelines*. For more information about cancellation of firmware IDs, refer to the *Intel Stratix 10 Device Security User Guide*.

You may also need to consider encryption key storage and management. If bitstream encryption is enabled on the Intel Stratix 10 device, you need to store the encryption key on the device. The encryption key may be stored in Battery-Backed RAM (BBRAM) or eFuses. Storing the encryption key in eFuses is permanent, while storing the encryption key in BBRAM allows for key wipe or reprovisioning. If the design requires encryption key storage in BBRAM, a non-volatile battery must be connected to the VCCBAT pin. For more information about connecting a battery to the VCCBAT pin, refer to the *Intel Stratix 10 Device Family Pin Connection Guidelines*. Intel Stratix 10 devices with -BK suffix support black key provisioning which helps protect the confidentiality of the Advanced Encryption Standard (AES) root key during the provisioning process.

Table 61. Authentication and Advanced Security Features Support for Intel Stratix 10 Devices

Security features are available in Intel Stratix 10 devices that support advanced security. Devices with advanced security enabled can only load firmware using the Intel Quartus Prime Pro Edition software.

Contact your Intel Sales representatives for more information about Intel Stratix 10 device security features.

Intel Stratix 10 Device Variant	Authentication	Advanced Security (Includes Encryption)
GX	Yes	-AS suffix devices
GX 10M	Yes	No
SX	Yes	-AS suffix devices
MX	Yes	-AS suffix devices
TX	Yes	-AS suffix devices
DX	Yes	Yes

Related Information

- [Intel Stratix 10 Device Family Pin Connection Guidelines](#)
- [Intel Stratix 10 Device Security User Guide](#)
Provides more information about cancellation of firmware IDs.

Design Entry

In complex FPGA design development, design practices, coding styles, and IP cores use have an enormous impact on your device's timing performance, logic utilization, compilation time, and system reliability. In addition, while planning and creating the design, plan for a hierarchical or team-based design to improve design productivity.

Design Recommendations

Table 62. Design Recommendations Checklist

Number	Done?	Checklist Item
1		Use synchronous design practices. Pay attention to clock signals.

In a synchronous design, a clock signal triggers all events. When all of the registers' timing requirements are met, a synchronous design behaves in a predictable and reliable manner for all process, voltage, and temperature (PVT) conditions. You can easily target synchronous designs to different device families or speed grades.

Problems with asynchronous design techniques include reliance on propagation delays in a device, incomplete timing analysis, and possible glitches. Pay particular attention to your clock signals, because they have a large effect on your design's timing accuracy, performance, and reliability. Problems with clock signals can cause functional and timing problems in your design. Use dedicated clock pins and clock routing for best results. For clock inversion, multiplication, and division, use the device PLLs. For clock multiplexing and gating, use the dedicated clock control block or PLL clock switchover feature instead of combinational logic. If you must use internally generated clock signals, register the output of any combinational logic used as a clock signal to reduce glitches. For example, if you divide a clock using combinational logic, clock the final stage with the clock signal used to clock the divider circuit.

Related Information

- [PLL Board Design Guidelines](#) on page 26
Provides more information about designing clock and PLL.
- [Recommended Design Practices chapter, Design Recommendations User Guide \(Intel Quartus Prime Pro Edition\)](#)
Provides more information about design recommendations.

Using IP Cores

Table 63. Using IP Cores Checklist

Number	Done?	Checklist Item
1		Use IP cores with the parameter editor.

Intel provides parameterizable IP cores that are optimized for Intel device architectures. You can save design time by using IP cores instead of coding your own logic. Additionally, the Intel-provided IP cores can offer more efficient logic synthesis and device implementation. You can scale the IP core's size and set various options with parameters. IP cores include the library of parameterized modules (LPM) and Intel device-specific IP cores. You can also take advantage of Intel and third-party IP cores and reference designs to save design time. The Intel Quartus Prime IP catalog

provides a user interface to customize IP cores. You should build or change IP core parameters using the parameter editor to ensure you set all ports and parameters correctly.

Related Information

- [IP Selection](#) on page 6
- [Documentation: User Guides](#)
Provides detailed information about the specific IP cores.

Reconfiguration

Table 64. Reconfiguration Checklist

Number	Done?	Checklist Item
1		Consider the reconfiguration feature for your board development.

Intel Stratix 10 devices allow you to easily modify your transceivers and FPGA-core while other portions of your design are still running by using dynamic reconfiguration and partial reconfiguration, respectively.

Intel Stratix 10 devices allow you to dynamically reconfigure different portions of the transceivers for different protocols, data rates, and PMA settings without powering down any part of the device or interrupting adjacent transceiver channels. This feature will be available in a future release of the Intel Quartus Prime software.

If you are interested in using partial reconfiguration, contact your local Intel representatives for support.

Related Information

[Reconfiguration Interface and Dynamic Reconfiguration chapter, Intel Stratix 10 Transceiver PHY User Guide](#)

Provides more information about dynamic reconfiguration.

Recommended HDL Coding Styles

Table 65. Recommended HDL Coding Styles Checklist

Number	Done?	Checklist Item
1		Follow recommended coding styles, especially for inferring device dedicated logic such as memory and DSP blocks.

HDL coding styles can have a significant effect on the quality of results for programmable logic designs. Use Intel’s recommended coding styles to achieve optimal synthesis results. When designing memory and digital system processing (DSP) functions, understand the device architecture so you can take advantage of the dedicated logic block sizes and configurations.

Related Information

- [Recommended HDL Coding Styles chapter, Design Recommendations User Guide \(Intel Quartus Prime Pro Edition\)](#)
Provides the specific HDL coding examples and recommendations. Refer to your synthesis tool's documentation for any additional tool-specific guidelines. In the Intel Quartus Prime software, you can use the HDL examples in the Language Templates available from the right-click menu in the text editor.
- [Intel Stratix 10 High-Performance Design Handbook](#)
Provides more information about the Intel Hyperflex™ feature.

Design Implementation, Analysis, Optimization, and Verification

After you create your design source code and apply constraints including the device selection and timing requirements, your synthesis tool processes the code and maps it to elements of the device architecture. The Intel Quartus Prime Fitter then performs placement and routing to implement the design elements in specific device resources. If required, you can use the Intel Quartus Prime software to optimize the design's resource utilization and achieve timing closure, preserve the performance of unchanged design blocks, and reduce compilation time for future iterations. You can also verify the design functionality with simulation. This section provides guidelines for these stages of the compilation flow.

Selecting a Synthesis Tool

Table 66. Selecting a Synthesis Tool Checklist

Number	Done?	Checklist Item
1		Specify your synthesis tool and use the correct supported version.

The Intel Quartus Prime software includes advanced and easy-to-use integrated synthesis that fully supports Verilog HDL and VHDL, as well as the Intel hardware description language (AHDL) and schematic design entry. You can also use industry-leading third-party EDA synthesis tools to synthesize your Verilog HDL or VHDL design, and then compile the resulting output netlist file in the Intel Quartus Prime software. Specify a third-party synthesis tool in the New Project Wizard or the **EDA Tools Settings** page of the **Settings** dialog box to use the correct Library Mapping File (.lmf) for your synthesis netlist.

Intel recommends using the most recent version of third-party synthesis tools, because tool vendors are continuously adding new features, fixing tool issues, and enhancing performance for Intel devices.

Different synthesis tools can give different results. If you want to select the best-performing tool for your application, you can experiment by synthesizing typical designs for your application and coding style and comparing the results. Be sure to perform placement and routing in the Intel Quartus Prime software to get accurate timing analysis and logic utilization results.

Your synthesis tool might offer the capability to create a Intel Quartus Prime project and pass constraints such as the EDA tool setting, device selection, and timing requirements that you specified in your synthesis project. You can use this capability to save time when setting up your Intel Quartus Prime project for placement and routing.

Related Information

- [Synthesis Tool, Getting Started User Guide \(Intel Quartus Prime Pro Edition\)](#)
Provides more information about the supported synthesis tools.
- [Intel Quartus Prime Pro Edition Software and Device Support Release Notes](#)
Lists the version of each synthesis tool that is officially supported by that version of the Intel Quartus Prime software.

Device Resource Utilization Reports

Table 67. Device Resource Utilization Reports Checklist

Number	Done?	Checklist Item
1		Review resource utilization reports after compilation.

After compilation in the Intel Quartus Prime software, review the device resource utilization information to determine whether the future addition of extra logic or other design changes introduce fitting difficulties. If your compilation results in a no-fit error, resource utilization information is important so you can analyze the fitting problems in your design.

To determine resource usage, refer to the **Flow Summary** section of the Compilation Report for a percentage representing the total logic utilization, which includes an estimation of resources that cannot be used due to existing connections or logic use.

For Intel Stratix 10 devices, low logic utilization does not have the lowest ALM utilization possible. In addition, a design that is reported as close to 100% full might still have space for extra logic. The Fitter uses ALUTs in different ALMs, even when the logic can be placed within one ALM, so that it can achieve the best timing and routability results. Logic might be spread throughout the device when achieving these results. As the device fills up, the Fitter automatically searches for logic that can be placed together in one ALM.

More detailed resource information is available by viewing the reports under **Fitter > Place** section of the Compilation Report. The Fitter **Resource Usage Summary** report breaks down the logic utilization information and indicates the number of fully and partially used ALMs, and provides other resource information including the number of bits in each type of memory block. There are also reports that describe some of the optimizations that occurred during compilation. For example, if you use the Intel Quartus Prime integrated synthesis, the reports under **Analysis & Synthesis > Partition <partition_name> > Optimization Results** provide information, including registers that were removed during synthesis. Use this report to estimate device resource utilization for a partial design to ensure that registers were not removed due to missing connections with other parts of the design.

Intel Quartus Prime Messages

Table 68. Intel Quartus Prime Messages Checklist

Number	Done?	Checklist Item
1		Review all Intel Quartus Prime messages, especially warning or error messages.

Each stage of the compilation flow generates messages, including informational notes, warnings, and critical warnings. Review these messages to check for any design problems. Ensure that you understand the significance of any warning messages, and

make changes to the design or settings if required. In the Intel Quartus Prime user interface, you can use the **Message** window tabs to look at only certain types of messages, and you can suppress messages if you have determined that they do not require any action from you.

Related Information

[Viewing Project Messages, Getting Started User Guide \(Intel Quartus Prime Pro Edition\)](#)

Provides more information about messages and message suppression.

Timing Constraints and Analysis

Table 69. Design Specifications Checklist

Number	Done?	Checklist Item
1		Ensure timing constraints are complete and accurate, including all clock signals and I/O delays.
2		Review the Timing Analyzer reports after compilation to ensure there are no timing violations.
3		Ensure that the input I/O times are not violated when data is provided to the Intel Stratix 10 device.

In an FPGA design flow, accurate timing constraints allow timing-driven synthesis software and place-and-route software to obtain optimal results. Timing constraints are critical to ensure designs meet their timing requirements, which represent actual design requirements that must be met for the device to operate correctly. The Intel Quartus Prime software optimizes and analyzes your design using different timing models for each device speed grade, so you must perform timing analysis for the correct speed grade. The final programmed device might not operate as expected if the timing paths are not fully constrained, analyzed, and verified to meet requirements.

The Intel Quartus Prime software includes the Intel Quartus Prime Timing Analyzer, a powerful ASIC-style timing analysis tool that validates the timing performance of all logic in your design. It supports the industry standard Synopsys* Design Constraints (SDC) format timing constraints, and has an easy-to-use GUI with interactive timing reports. It is ideal for constraining high-speed source-synchronous interfaces and clock multiplexing design structures.

A comprehensive static timing analysis includes analysis of register to register, I/O, and asynchronous reset paths. It is important to specify the frequencies and relationships for all clocks in your design. Use input and output delay constraints to specify external device or board timing parameters. Specify accurate timing requirements for external interfacing components to reflect the exact system intent.

The Timing Analyzer performs static timing analysis on the entire system, using data required times, data arrival times, and clock arrival times to verify circuit performance and detect possible timing violations. It determines the timing relationships that must be met for the design to correctly function.

You can use the `report_datasheet` command to generate a datasheet report that summarizes the I/O timing characteristics of the entire design.

Related Information

The Intel Quartus Prime Timing Analyzer chapter, [Timing Analyzer User Guide \(Intel Quartus Prime Pro Edition\)](#)
Provides more information about timing analysis.

Recommended Timing Optimization and Analysis Assignments

Table 70. Recommended Timing Optimization and Analysis Assignments Checklist

Number	Done?	Checklist Item
1		Turn on Optimize multi-corner timing on the Fitter Settings page in the Settings dialog box.
2		Use <code>create_clock</code> and <code>create_generated_clock</code> to specify the frequencies and relationships for all clocks in your design.
3		Use <code>set_input_delay</code> and <code>set_output_delay</code> to specify the external device or board timing parameters.
4		Use <code>derive_clock_uncertainty</code> to automatically apply inter-clock, intra-clock, and I/O interface uncertainties.
5		Use <code>check_timing</code> to generate a report on any problem with the design or applied constraints, including missing constraints.
6		Use <code>set_false_path</code> or <code>set_clock_groups</code> for asynchronous paths.

These assignments and settings are important for large designs such as those in Intel Stratix 10 devices.

When you turn on the **Optimize multi-corner timing** option, the design is optimized to meet its timing requirements at all timing process corners and operating conditions. Therefore, turning on this option helps create a design implementation that is more robust across PVT variations.

In your Timing Analyzer `.sdc` constraints file, apply the recommended constraints to your design.

Related Information

[Timing Constraints, Timing Analyzer User Guide \(Intel Quartus Prime Pro Edition\)](#)
Provides more guidelines about timing constraints.

Area and Timing Optimization

Table 71. Area and Timing Optimization Checklist

Number	Done?	Checklist Item
1		Run Fitter (Plan) if you want timing estimates before running a full compilation.
2		Use Intel Quartus Prime optimization features to achieve timing closure or improve the resource utilization.
3		Use the Timing Optimization Advisors to suggest optimization settings.

This section highlights some of the features offered in the Intel Quartus Prime software to help optimize area (or resource utilization) and timing performance. If the timing analysis reports that your design requirements were not met, you must make

changes to your design or settings and recompile the design to achieve timing closure. If your compilation results in no-fit messages, you must make changes to get successful placement and routing.

You can run Fitter (Plan) to estimate your design’s timing results before the software performs full placement and routing. Click **Processing > Start > Start Fitter (Plan)** to generate initial compilation results after you have run analysis and synthesis.

Physical synthesis optimizations make placement-specific changes to the netlist that improve results for a specific Intel device. You can optimize for performance by selecting **High Performance Effort** or **Superior Performance** Optimization Mode in the **Compiler Settings**. These optimization modes turn on the **Advanced Physical Synthesis** option under the **Advanced Fitter Settings**. If you turn on these options, ensure that they do improve the results for your design. If you do not require these options to meet your design timing requirements, turn off the options to reduce the compilation time.

The Design Space Explorer II (DSE II) is a utility that automates the process to find optimal project settings for resource, performance, or power optimization goals. DSE II attempts multiple seeds to identify one that meets your requirements. The **Exploration Panel > Exploration mode** allows you a predefine exploration space to target design performance, area of improvements, or power reduction with multiple compilations.

The Optimization Advisors provide guidance in making settings that optimize your design. On the Tools menu, click **Advisor > Timing Optimization Advisor**. Evaluate the options and choose settings that suit your requirements.

Related Information

- [Area Optimization chapter, Design Optimization User Guide \(Intel Quartus Prime Pro Edition\)](#)
Provides more information about additional optimization features.
- [Timing Closure and Optimization chapter, Design Optimization User Guide \(Intel Quartus Prime Pro Edition\)](#)
Provides more information about additional optimization features.
- [Netlist Optimizations and Physical Synthesis chapter, Design Optimization User Guide \(Intel Quartus Prime Pro Edition\)](#)
Provides more information about additional optimization features.
- [Power Optimization](#) on page 54
Provides more information about the optimization for power usage.
- [Intel Quartus Prime Help](#)
Provides more information about the Design Space Explorer.

Preserving Performance and Reducing Compilation Time

Table 72. Preserving Performance and Reducing Compilation Time Checklist

Number	Done?	Checklist Item
1		Use incremental compilation to preserve performance for unchanged blocks in your design and to reduce compilation times.
2		Ensure parallel compilation is enabled if you have multiple processors available for compilation.
3		Use the Compilation Time Advisor to suggest settings that reduce compilation time.

Use the incremental compilation feature to preserve logic in unchanged parts of your design, preserve timing performance, and reach timing closure more efficiently. You can speed up design iteration time by an average of 60% when making changes to the design with the incremental compilation feature.

The Intel Quartus Prime software can run some algorithms in parallel to take advantage of multiple processors and reduce compilation time when more than one processor is available to compile the design. Set the **Parallel compilation** option on the **Compilation Process Settings** page of the **Settings** dialog box, or change the default setting in the **Options** dialog box in the **Processing** page from the Tools menu.

The Compilation Time Advisor provides guidance in making settings that reduce your design compilation time. On the Tools menu, point to **Advisors** and click **Compilation Time Advisor**. Using some of these techniques to reduce compilation time can reduce the overall quality of results.

Designing with Intel Hyperflex™

Table 73. Designing with Intel Hyperflex™ Checklist

Number	Done?	Checklist Item
1		Use Intel Hyperflex™ feature to optimize your design and achieve enhanced performance.

Intel Hyperflex core architecture adds registers to both the interconnect routing and the inputs of all major functional blocks in the FPGA. These added registers, called Hyper-Registers, are different from conventional registers. Conventional registers are present only in the adaptive logic modules (ALMs). Hyper-Registers can help to achieve significant core performance improvement.

To achieve this enhanced performance, you must optimize your designs using the following steps:

1. Hyper-Retiming
2. Hyper-Pipelining
3. Hyper-Optimization

Related Information

[Intel Stratix 10 High-Performance Design Handbook](#)

Provides more information about Hyper-Retiming, Hyper-Pipelining, and Hyper-Optimization.

Simulation

Table 74. Simulation Checklist

Number	Done?	Checklist Item
1		Specify your simulation tool, and use the correct supported version and simulation models.

The Intel Quartus Prime software supports both RTL and gate level functional simulations. Perform functional simulation at the beginning of your design flow to check the design functionality or logical behavior of each design block. You do not have to fully compile your design; you can generate a functional simulation netlist that does not contain timing information.

Intel provides the ModelSim* - Intel FPGA Starter Edition and offers the higher performance ModelSim - Intel FPGA Edition, which enable you to take advantage of advanced testbench capabilities and other features. In addition, the Intel Quartus Prime EDA Netlist Writer can generate timing netlist files to support other third-party simulation tools such as Synopsys VCS, Cadence NC-Sim, and Aldec Active-HDL. Specify your simulation tool in the **EDA Tools Settings** page of the **Settings** dialog box to generate the appropriate output simulation netlist.

If you use a third-party simulation tool, use the software version that is supported with your Intel Quartus Prime software version. The Intel Quartus Prime Software Release Notes list the version of each simulation tool that is officially supported with that particular version of the Intel Quartus Prime software. Use the model libraries provided with your Intel Quartus Prime software version, because libraries can change between versions, which might cause a mismatch with your simulation netlist. To create a testbench, on the Processing menu, point to **Start** and click **Start Test Bench Template Writer**.

Related Information

- [Simulating Intel FPGA Designs chapter, Third-party Simulation User Guide \(Intel Quartus Prime Pro Edition\)](#)
Provides more information about the simulation tool flows.
- [Intel Quartus Prime Pro Edition Software and Device Support Release Notes](#)
Lists the version of each simulation tool that is officially supported with that particular version of the Intel Quartus Prime software.

Power Analysis

Table 75. Power Analysis Checklist

Number	Done?	Checklist Item
1		After compilation, analyze power consumption and heat dissipation in the Power Analyzer.
2		Provide accurate signal activities, preferably with a gate-level simulation .vcd, to get accurate power analysis results.
3		Specify the correct operating conditions for power analysis.

Before design completion, estimate power consumption using the PTC. After compiling your design, analyze the power consumption and heat dissipation with the Intel Quartus Prime Power Analyzer to ensure the design has not violated power supply and thermal budgets.

You must compile a design (to provide information about design resources, placement and routing, and I/O standards) and provide signal activity data (toggle rates and static probabilities) to use the Power Analyzer. You can derive signal activity data from simulation results or a user-defined default toggle rate and vectorless estimation. The signal activities used for analysis must be representative of the actual operating behavior. For the most accurate power estimation, use gate-level simulation results with a .vcd output file from a third-party simulation tool. The simulation activity should include typical input vectors over a realistic time period and not the corner cases often used during functional verification. Use the recommended simulator settings (such as glitch filtering) to ensure good results.

You must also specify operating conditions, including the core voltage, device power characteristics, ambient and junction temperature, cooling solution, and the board thermal model. Select the appropriate settings on the **Operating Settings and Conditions** page in the **Settings** dialog box.

To calculate the dynamic, static, and I/O thermal power consumption, on the Processing menu, click **Power Analyzer Tool**. The tool also provides a summary of the signal activities used for analysis and a confidence metric that reflects the overall quality of the data sources for signal activities.

The report is a power estimate based on the data provided, and is not a power specification. Always refer to the datasheet for the power specification of your device.

Related Information

- [Power and Thermal Estimation](#) on page 11
- [Simulation](#) on page 52
- [Absolute Maximum Ratings, Intel Stratix 10 Device Datasheet](#)
Provides the power specifications.
- [Power Analysis chapter, Power Analysis and Optimization User Guide \(Intel Quartus Prime Pro Edition\)](#)
Provides more information about power analysis and recommendations for simulation settings for creating signal activity information.

Power Optimization

Intel Stratix 10 devices utilize advanced process and circuit techniques, along with major circuit and architecture innovations, to minimize power and deliver high performance.

To reduce dynamic power consumption in Intel Stratix 10 devices, you can use various design and software techniques to optimize your design.

Power optimization in the Intel Quartus Prime software depends on accurate power analysis results. Use the guidelines in the previous section to ensure the software optimizes the power utilization correctly for the design’s operating behavior and conditions.

Device and Design Power Optimization Techniques

Table 76. Device and Design Power Optimization Techniques Checklist

Number	Done?	Checklist Item
1		Use recommended design techniques and Intel Quartus Prime options to optimize your design for power consumption, if required.
2		Use the Power Optimization Advisor to suggest optimization settings.

Related Information

- [Power Reduction Techniques, Intel Stratix 10 Power Management User Guide](#)
Provides more information about the power savings techniques.
- [Power Optimization chapter, Power Analysis and Optimization User Guide \(Intel Quartus Prime Pro Edition\)](#)
Provides more details and additional design techniques to reduce power consumption.

Device Speed Grade

Table 77. Device Speed Grade Checklist

Number	Done?	Checklist Item
1		Consider using a faster speed grade device.

If your design includes many critical timing paths that require the high-performance mode, you might be able to reduce power consumption by using a faster speed grade device if available.

Clock Power Management

Table 78. Clock Power Management Checklist

Number	Done?	Checklist Item
1		Optimize the clock power management.

Clocks represent a significant portion of dynamic power consumption, because of their high switching activity and long paths. The Intel Quartus Prime software automatically optimizes clock routing power by enabling only the portions of a clock network that are required to feed downstream registers. You can also use clock control features to dynamically enable or disable the clock network. When a clock network is powered down, all the logic fed by that clock network does not toggle, thereby reducing the overall power consumption of the device.

To reduce LAB-wide clock power consumption without disabling the entire clock tree, use the LAB-wide clock enable signal to gate the LAB-wide clock. The Intel Quartus Prime software automatically promotes register-level clock enable signals to the LAB level.

Related Information

[Clock Control Intel Stratix 10 FPGA IP Core, Intel Stratix 10 Clocking and PLL User Guide](#)

Provides more information about using the clock control features.

Memory Power Reduction

Table 79. Memory Power Reduction Checklist

Number	Done?	Checklist Item
1		Reduce the number of memory clocking events.

Reduce the number of memory clocking events to reduce memory power consumption. You can use clock gating or the clock enable signals in the memory ports.

Related Information

[Clock Gating, Intel Stratix 10 Clocking and PLL User Guide](#)

Provides more information about clock gating.

I/O Power Guidelines

Table 80. I/O Power Guidelines Checklist

Number	Done?	Checklist Item
1		Review the I/O power guidelines.

The dynamic power consumed in the I/O buffer is proportional to the total load capacitance; therefore, lower capacitance reduces power consumption.

Non-terminated I/O standards such as LVTTTL and LVCMOS have a rail-to-rail output swing equal to the V_{CCIO} supply voltage. Because dynamic power is proportional to the square of the voltage, use lower voltage I/O standards to reduce dynamic power. These I/O standards consume little static power.

Because dynamic power is also proportional to the output transition frequency, use resistively-terminated I/O standards such as SSTL for high-frequency applications. The output load voltage swings by an amount smaller than the V_{CCIO} around a bias point; therefore, dynamic power is lower than for non-terminated I/O under similar conditions.

Resistively-terminated I/O standards dissipate significant static power because current is constantly driven into the termination network. Use the lowest drive strength that meets your speed and waveform requirements to minimize static power when using resistively terminated I/O standards.

The power used by external devices is not included in the PTC calculations, so be sure to include it separately in your system power calculations.

Intel Quartus Prime Power Optimization Techniques

Table 81. Intel Quartus Prime Power Optimization Techniques Checklist

Number	Done?	Checklist Item
1		Review recommended design techniques and Intel Quartus Prime options to optimize power consumption.

The Intel Quartus Prime software offers power-optimized synthesis and fitting to reduce core dynamic power.

Optimizing your design for area also saves power because fewer logic blocks are used; therefore, there is typically less switching activity. Improving your design source code to optimize for performance can also reduce power usage. You can use the DSE and Power Optimization Advisor to provide additional suggestions to reduce power.

Related Information

[Power Optimization chapter, Power Analysis and Optimization User Guide \(Intel Quartus Prime Pro Edition\)](#)

Provides more information about power-driven compilation and power optimization.

Power Optimization Advisor

Table 82. Power Optimization Advisor Checklist

Number	Done?	Checklist Item
1		Use the Power Optimization Advisor to suggest optimization settings.

The Intel Quartus Prime software includes the Power Optimization Advisor, which provides specific power optimization advice and recommendations based on the current design project settings and assignments. On the Tools menu, point to **Advisors** and click **Power Optimization Advisor**. After making any of the recommended changes, recompile your design and run the Power Analyzer to check the change in your power results.

Document Revision History for Intel Stratix 10 Device Design Guidelines

Document Version	Changes
2021.04.22	<ul style="list-style-type: none"> Added information about Intel Stratix 10 devices with -BK suffix that support black key provisioning in the <i>Security Considerations</i> section. Updated the <i>Authentication and Advanced Security Features Support for Intel Stratix 10 Devices</i> table. <ul style="list-style-type: none"> Added description about security features in Intel Stratix 10 devices that support advanced security. Added Intel Stratix 10 GX 10M device.
2020.12.14	Added checklist item on Quad SPI flash operations in the <i>Planning for Device Configuration Checklist</i> table.
2020.06.30	<ul style="list-style-type: none"> Updated the <i>Planning for Device Configuration Checklist</i> table. <ul style="list-style-type: none"> Added checklist item on Intel Stratix 10 GX 10M devices configuration. Added checklist items on nCONFIG operation. Added <i>Device Power Cycling and Reconfiguration</i> section. Updated checklist item to add guidelines on V_{CCIO} when using the Avalon-ST ×32 or ×16 configuration scheme in the <i>Configuration Pin Voltage Level Checklist</i> table. Removed all instances of EPCQ-L. This flash device is obsolete. Updated the serial flash information in the following sections: <ul style="list-style-type: none"> <i>Serial Configuration Devices</i> <i>Configuration Pin Voltage Level</i> Removed guidelines related to SD/MMC device configuration in the following sections: <ul style="list-style-type: none"> Removed SD/MMC flash memories support in the <i>Configuration Pin Voltage Level</i> section. Removed S_{DMMC_CFG_CCLK} pin in the <i>Clock Trace Signal Integrity</i> section. Changed Early Power Estimator (EPE) to Intel FPGA Power and Thermal Calculator (PTC).
2019.12.16	<ul style="list-style-type: none"> Added checklist items on I/O pins support limitation for banks 3A, 3C, and 3D in the <i>Selectable Standards and Flexible I/O Banks Checklist</i> table. Added <i>Security Considerations</i> section.
2019.10.10	<ul style="list-style-type: none"> Updated checklist item on configuration guidelines and additional clock requirements in the <i>Planning for Device Configuration Checklist</i> table. Added checklist items on I/O pins support limitation for LVDS SERDES in the <i>Selectable Standards and Flexible I/O Banks Checklist</i> table. Updated checklist item on DQS and DQ pins and added checklist item on I/O pins support limitations for EMIF in the <i>Memory Interfaces Checklist</i> table. Updated the description on DQS and DQ pins for EMIF supported banks.
<i>continued...</i>	

Document Version	Changes
2019.06.24	<ul style="list-style-type: none"> Added a checklist item on PMBus-compliant voltage regulator for SmartVID devices in the <i>Power Pin Connections and Power Supplies Checklist</i> table. Added a link to the PDN website in the <i>Decoupling Capacitors</i> section. Added a checklist item on unterminated SSO pins in the <i>I/O Simultaneous Switching Noise Checklist</i> table.
2019.04.02	<ul style="list-style-type: none"> Added a checklist item on Intel Stratix 10 Reset Release IP in the <i>Planning for Device Configuration Checklist</i> table. Updated the guidelines on physical synthesis optimizations in the <i>Area and Timing Optimization</i> section.
2018.09.24	<ul style="list-style-type: none"> Added a checklist item on configuration guidelines and additional clock requirements for designs using PCIe, transceiver channels, HPS, High Bandwidth Memory (HBM2) IP core, or SmartVID in the <i>Planning for Device Configuration Checklist</i> table. Added a checklist item on SmartVID connection and the VCC voltage regulator in the <i>Other Configuration Pins Checklist</i> table.
2018.05.07	<ul style="list-style-type: none"> Updated checklist item in the <i>Device Variant Checklist</i> table. Added links to transceiver documents in the following sections: <ul style="list-style-type: none"> Speed Grade Vertical Device Migration Transceiver Board Design Guidelines Removed NAND configuration scheme. Renamed the following IP cores as per Intel rebranding: <ul style="list-style-type: none"> Renamed Intel FPGA S10 Temperature Sensor IP core to Temperature Sensor Intel Stratix 10 FPGA IP core. Renamed Virtual JTAG IP core to Virtual JTAG Intel FPGA IP core. Renamed SLD_VIRTUAL_JTAG IP core to SLD_VIRTUAL_JTAG Intel FPGA IP core. Renamed Stratix 10 External Memory Interfaces IP core to External Memory Interfaces Intel Stratix 10 FPGA IP core. Renamed Stratix 10 Intel FPGA PHYLite for Parallel Interfaces IP core to PHYLite for Parallel Interfaces Intel Stratix 10 FPGA IP core. Renamed Intel FPGA IOPLL IP core to IOPLL Intel FPGA IP core. Renamed Stratix 10 Clock Control IP core to Clock Control Intel Stratix 10 FPGA IP core. Removed the LPM_CONSTANT IP core. Not supported in the Intel Stratix 10 devices.

Date	Version	Changes
December 2017	2017.12.12	<ul style="list-style-type: none"> Updated checklist item in the <i>Design Specifications Checklist</i> table. Updated checklist item in the <i>IP Selection Checklist</i> table. Updated checklist item in the <i>PLLs and Clock Routing Checklist</i> table. Updated the <i>Logic, Memory, and Multiplier Density</i> section. Added checklist item in the <i>Vertical Device Migration Checklist</i> table. Added external TSD information in the <i>Temperature Sensing for Thermal Management</i> section. Added thermal information in the following sections: <ul style="list-style-type: none"> Early Power Estimation Thermal Management and Design Temperature Sensing for Thermal Management Changed the section title from <i>Data Compression</i> to <i>Configuration Bitstream Compression</i>. Updated the <i>Optional Configuration Pins</i> section. Added description for the Signal Tap Embedded Logic Analyzer in the <i>On-Chip Debugging Tools</i> section. Added checklist item to the <i>Power Pin Connections and Power Supplies Checklist</i> table. Updated checklist item in the <i>Board-Related Intel Quartus Prime Settings Checklist</i> table.

continued...

Date	Version	Changes
		<ul style="list-style-type: none"> • Updated the links in the <i>Memory Interfaces</i> section. • Updated the <i>Dual-Purpose and Special Pin Connections</i> section. • Updated the <i>Design Entry</i> section. • Removed information on Design Assistant in the <i>Design Recommendations</i> section. • Updated the <i>Reconfiguration</i> section. • Removed information on formal verification in the <i>Design Implementation, Analysis, Optimization, and Verification</i> section. • Updated report locations in the <i>Device Resource Utilization Reports</i> section. • Updated the <i>Timing Constraints and Analysis</i> section. • Updated the <i>Recommended Timing Optimization and Analysis Assignments Checklist</i> table. <ul style="list-style-type: none"> — Removed checklist item on <code>derive_pll_clocks</code>. — Added checklist item on <code>set_false_path</code> and <code>set_clock_groups</code>. • Updated the <i>Area and Timing Optimization</i> section. • Updated description on Hyper-Registers in the <i>Designing with Intel Hyperflex</i> section. • Removed information on NativeLink in the <i>Simulation</i> section. • Removed information on programmable power tiles in the following section: <ul style="list-style-type: none"> — <i>Power Optimization</i> — <i>Device Speed Grade</i> — <i>Intel Quartus Prime Power Optimization Techniques</i> • Removed the following sections: <ul style="list-style-type: none"> — <i>Device-Wide Output Enable Pin</i> — <i>Register Power-Up Levels and Control Signals</i> — <i>Formal Verification</i> • Updated the following terms: <ul style="list-style-type: none"> — Changed Qsys to Platform Designer — Changed OpenCore Plus to Intel FPGA IP Evaluation Mode — Changed TimeQuest Timing Analyzer to Timing Analyzer — Changed BluePrint Platform Designer to Interface Planner • Updated IP name from Altera PHYLite for Parallel Interfaces to PHY Lite for Parallel Interfaces. • Rebranded as Intel.
February 2017	2017.02.13	<ul style="list-style-type: none"> • Removed Start I/O Assignment Analysis command in Early Pin Planning and I/O Assignment Analysis section. • Removed incremental compilation feature. Removed the following topics: <ul style="list-style-type: none"> — Planning for Hierarchical and Team-Based Design — Planning Design Partitions — Planning in Bottom-Up and Team-Based Flows — Creating a Design Floorplan • Updated feature names. <ul style="list-style-type: none"> — Changed SignalProbe to Signal Probe — Removed PowerPlay text from tool name
October 2016	2016.10.31	Initial release.