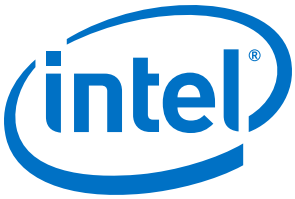


AN 704: FPGA-based Safety Separation Design Flow for Rapid Functional Safety Certification

AN-704 | 2018.09.01
Latest document on the web: PDF | HTML

Contents

FPGA-based Safety Separation Design Flow for Rapid Functional Safety Certification..... 3
About the Functional Safety Separation Flow.....3
Work Flows..... 5
Design Creation Flow.....5
Design Modification Flow.....6
Functional Safety Separation of a Motor Control Design Example..... 8
Design Considerations..... 9
Design Creation Flow.....11
Using the Design Modification Flow..... 22
Appendix A: Terminology.....25
Appendix B: Design Checklist 25
Document Revision History for AN 704: FPGA-based Safety Separation Design Flow for Rapid Functional Safety Certification.....27

AN-704 | 2018.09.01

Send Feedback

FPGA-based Safety Separation Design Flow for Rapid Functional Safety Certification

This design flow significantly reduces the certification efforts for the lifetime of an FPGA-based industrial system containing both safety critical and nonsafety critical components.

This application note describes how to use the design flow with a motor control system design example.

Industrial machinery manufacturers throughout the world experience the continuous pressure to reduce system cost, extend performance and efficiency, and deliver to ever reduced timescales. For products in safety critical environments, designers also strive to ensure safe behavior with compliance to *IEC 61508: Functional Safety of electrical/electronic/programmable electronic safety-related systems* and *ISO 26262: Road vehicles - Functional safety*.

FPGA-based systems provide designer with design flexibility performance scalability, and integration options. The TÜV Rheinland approved safety separation design flow retains these FPGA benefits and completely removes the need for a full design certification (when you don't change safety critical regions in a single FPGA device).

You can create safety and non-safety regions (or partitions) on a single Cyclone® IV, Cyclone V or Intel® MAX® 10 device. When you only change non-safety regions, the safety areas are fully preserved. The design flow provides the evidence that the placement and routing in the safety regions are identical to a previous hardware compilation. Then you should validate the FPGA design to ensure that the modified non-safety regions of the FPGA perform correctly with the safety regions. This validation may include functional tests of the safety regions.

About the Functional Safety Separation Flow

This flow extends the widely-adopted, proven Intel Quartus® Prime incremental compilation flow, which reduces compilation times by up to 70% through logic preservation.

The incremental compilation flow maps the design hierarchy to design partitions that the Intel Quartus Prime software treats separately during compilation. Intel defines a design partition as a logical partition. You use logical partitions with a physical placement constraint, a LogicLock region, to form the foundation for the safety flow.

In the functional safety separation flow, you categorize design partitions as either safety IP, which require complete preservation, or nonsafety IP. To configure a safety IP partition, set the partitions **Strict preservation** setting to **On**.

When you declare a design partition, every hierarchy within that partition becomes part of the same partition. When you create new partitions for hierarchies within an existing partition, the logic within the new lower-level partition is no longer part of the higher-level partition.

Figure 1. Partitions in a Design Hierarchy

B and F-G are design partitions. Partition B includes entity B which contains sub-entities D and E. Partition F-G includes entities F and G. The default partition, top, contains entities A and C which are not assigned to any other partition.

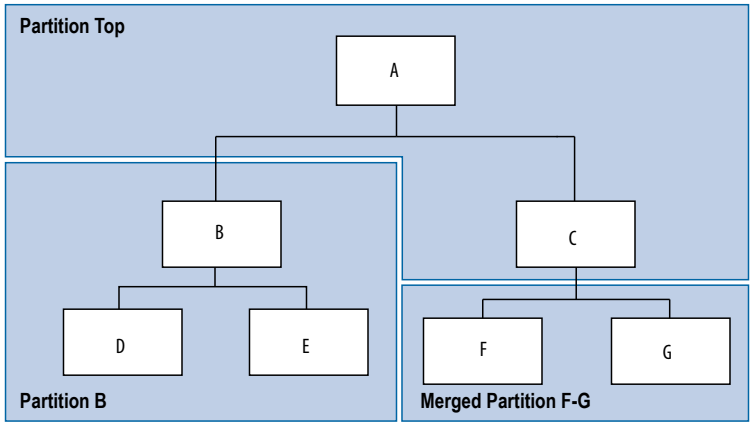
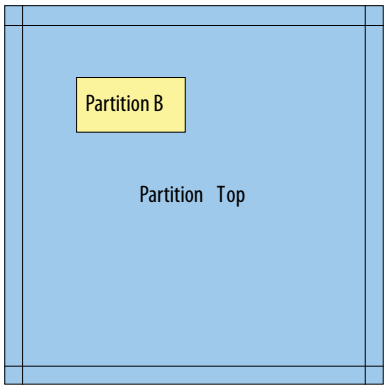


Figure 2. LogicLock Regions for Partitions in a Design Hierarchy

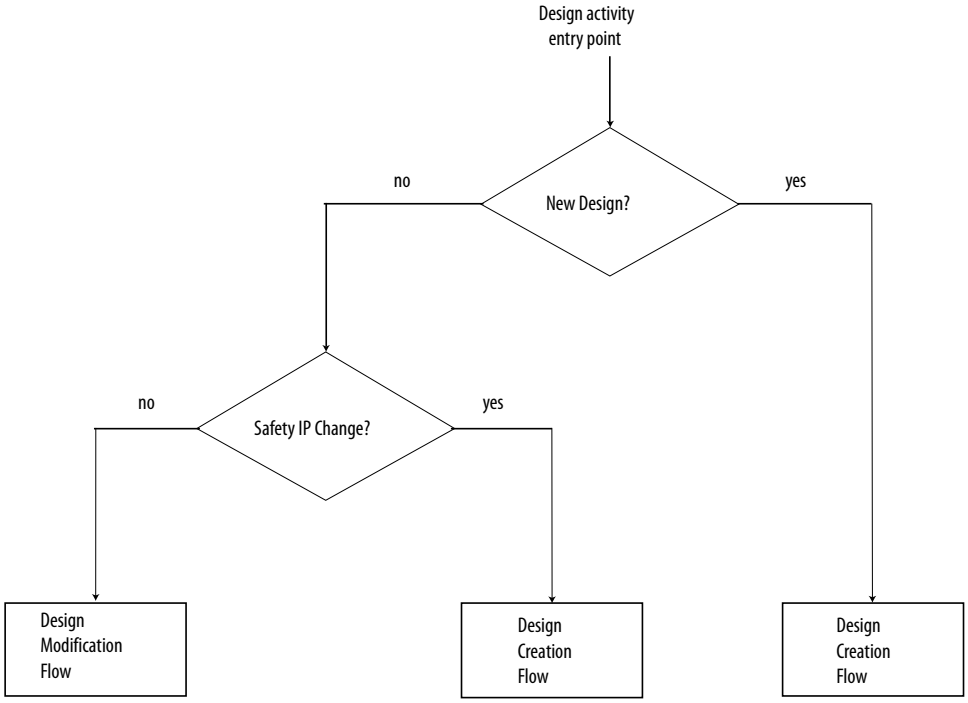
Use a LogicLock region to create a physical placement constraint for the logical partition B.



Work Flows

The functional safety separation flow consists of the design creation flow and the design modification flow. Both use incremental compilation, but the two flows have different use-case scenarios.

Figure 3. Work Flows



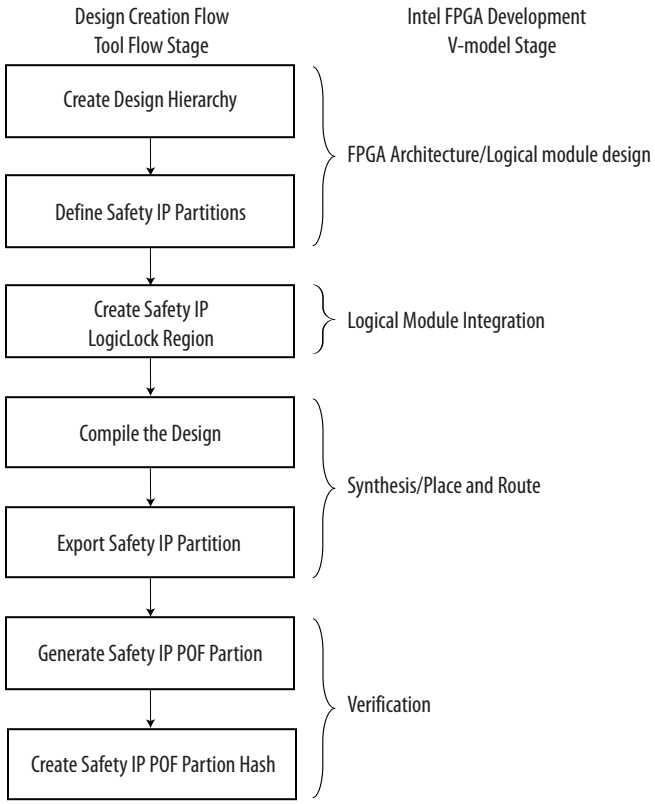
Design Creation Flow

This flow defines the necessary steps for initial design creation in a way that allows you to modify the nonsafety IP in your design without recertifying the safety IP. Some of the steps are architectural constraints. You need to perform the remaining steps in the Intel Quartus Prime software. You use the design creation flow for the first pass certification of your product.

The Intel FPGA development V-Model stage refers to the V-Model stages described in the Intel FPGA V-flow that Intel's Functional Safety Data Pack includes

Caution: When you make modifications to the safety IP in your design, you must use the design creation flow.

Figure 4. Design Creation Flow



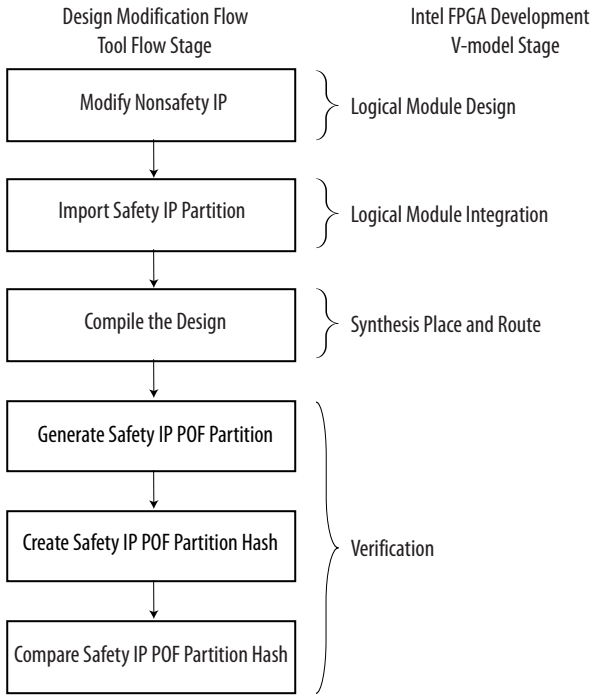
Design Modification Flow

This flow describes the necessary steps for you to modify the nonsafety IP in your design. This flow ensures that the previously compiled safety IP that the project uses remains unchanged when you change or compile nonsafety IP.

Caution: Use the design modification flow only after you qualify your design in the design creation flow.

For a general description of the global assignments required to enable this flow refer to the Intel Quartus PrimeSoftware Handbook.

Figure 5. Design Modification Flow



Note: The hash uses the MD5 algorithm.

Note: If your safety IP is a sub-block in a Qsys system, every time you regenerate HDL for the Qsys system, the timestamp for the safety IP HDL changes. When you change any HDL source file that belongs to a safety IP partition, by default the Intel Quartus Prime software resynthesises the partition and performs a clean place and route for that partition. For a clean place and route, the design creation flow is active for the safety IP. To change the default so that HDL changes do not cause resynthesis, and to keep the design modification flow active, you can either:

1. Use the partition export and import flow
2. Use the design partition window menu to modify the design partition properties and turn on **Ignore changes in source files and strictly use the specified netlist, if available**.

As the design modification flow preserves the placement and routing from the design creation flow compilation, Intel recommends that you try the design modification flow with representative changes to ensure that the FPGA placement and routing is not adversely affected by the design creation flow place and route. Adjust the safety partition LogicLock region size and/or location, clock routing and pin placement as necessary. If you have specific pin placement and or logic placement requirements for the non-safe logic ensure these resources are reserved during the design creation flow.

To check the Intel Quartus Prime software acheives the expected strict preservation, for each safety IP partition check the Intel Quartus Prime Fitter report sub-section Incremental Compilation Placement Preservation and Incremental Compilation Routing Preservation. In the design modification flow you see entries showing that the Intel Quartus Prime software preserves placement and routing for the safety IP partitions.

For more information, refer to the *Intel Quartus PrimeSoftware Handbook, chapter 3, Incremental Compilation for Hierarchical and Team based Design*

Functional Safety Separation of a Motor Control Design Example

A simplified motor control system demonstrates the functional safety separation flow. Intel derived this system from the Intel Drive-On-Chip Reference Design. The Drive-On-Chip design demonstrates concurrent multi-axis control of up to 4 three phase AC 400V permanent magnet synchronous motors (PMSMs) or brushless DC (BLDC) motors, and supports many combinations of device, development board, power board, and design parameterizations.

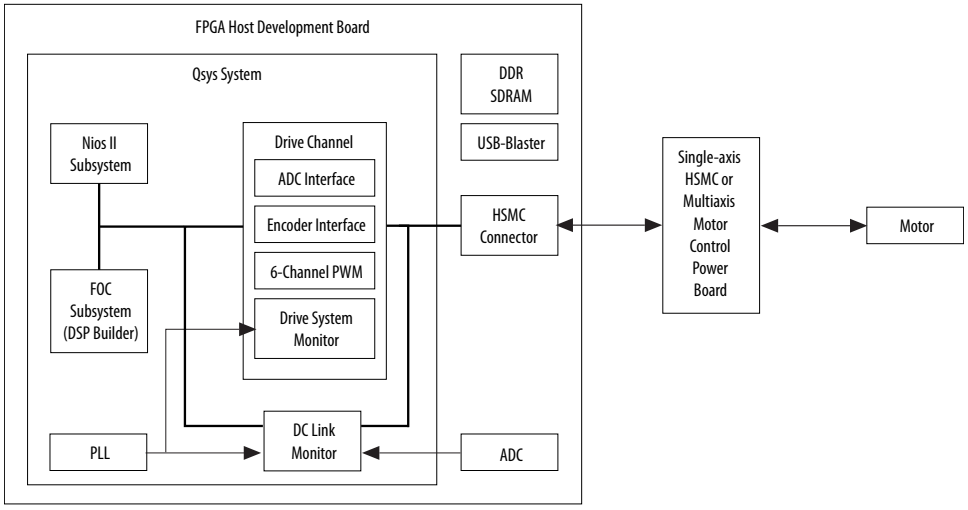
For the functional safety IP partition flow, Intel removes the following items from the Drive-On-Chip reference design for the simplified motor control system:

- Intel Quartus Prime project revisions
- *# defines* in the top level design file leaving support for one development platform configuration
- OpenCore Plus licensed IP (enDAT and BISS encoder interface components). You require full licenses to use the functional safety separation flow

Caution: This design example only demonstrates the functional safety separation flow. The design example is not fully functional and does not run on a hardware platform

Figure 6. Motor Control Design Example Block Diagram

The PLL provides clocks to all blocks in the Qsys system and the external ADC.



Related Information

Drive On-Chip Reference Design

Design Considerations

The architecture and design hierarchy options for an industrial system with functional safety requirements are vast, and coupled closely to the application. In this example, the safety IP partitions demonstrate the design flow in the context of multiple partitions and hierarchical partitions with associated clocking structures, PLLs, and IO pins.

This application note does not recommend which IP you should categorize as safety IP and nonsafety IP.

About Safety IP Partitions and LogicLock Regions

Intel recommends that you do not overlap any reserved LogicLock regions. The Intel Quartus Prime Fitter does not use overlapping areas. The chip planner checks for overlapping regions.

You must not create safety IP partitions as sub-partitions of another safety partition. If you create a safety IP partition and then place a safety IP partition inside that partition, the Intel Quartus Prime software does not achieve strict preservation. The design modification flow detects this preservation failure.

Assigning I/O Pins

You should consider all I/O pins that connect to a safety IP to be included in the safety IP partition.

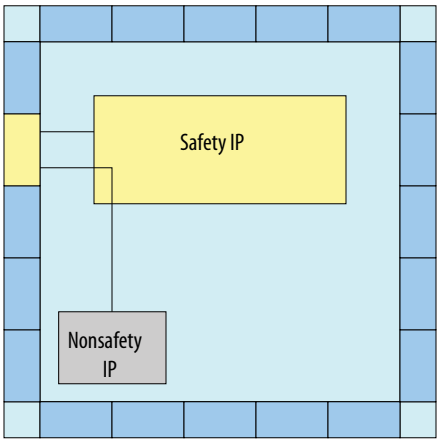
The functional safety separation flow preserves the pin locations and routing to and from the safety IP partition logic. The partially preserved bitstream contains information that IOs are still connected and configured the same as some of the programming bits. If an IO_REG group contains a pin that you assign to a safety IP, the Intel Quartus Prime software exclusively reserves all pins in the IO_REG group for this safety IP. Because an IO register bank contains 16 pins, this restriction can potentially waste pins.

1. If this restriction causes pin assignment problems with other partitions because of unused pins in safety IP IO_REG groups, preallocate unused I/O pins in the safety IP IO_REG group, so other partitions can use them.
2. Connect unused pins to the safety IP, which pass through to internal ports.

   To enable a signal to go through a Safety IP partition, add input and output ports to the HDL for the partition to allow the non-safe signal to pass through and ensure some logic, which the synthesis tool does not optimize away. The simplest logic that you can add is a wire-lut, by using a synthesis keep command on the signal that passes through the Safety IP. You can connect these pins to nonsafety IP partitions without requiring changes to the safety IP.

Figure 7. Tunnel Nonsafety IP Signal Through Safety IP Region

Sharing unused pins in a safe IO register bank with a non-safety IP partition

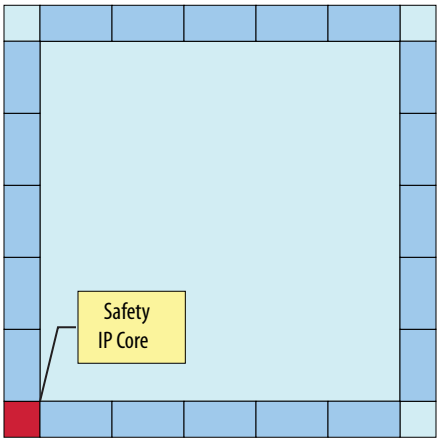


Clocks, PLLs and Resets

Consider carefully clocks and their associated resets for a safety IP. PLLs often generate clocks in the FPGA using an external reference clock source. To preserve the PLL configuration and clock routing in the safety IP, optionally include the PLL in the safety IP partition. However, clocks and resets often require additional safety checking measures to ensure the safety IP is operating to specification. These measures may include clock checking functionality.

Figure 8. PLL and Safety IP Placement (Red)

Note: The Intel Quartus Prime software preserves the routing to and from the PLL in the partially preserved bitstream



After a design creation flow compliation, the Intel Quartus Prime software fixes all routing for the safety IP. Ensure that the design's clock networks allow the flexibility the design requires for the design modification flow. If necessary, assign clock networks manually and consider manually adding `altclkctrl` buffers to the safety IP to increase PLL placement flexibility.

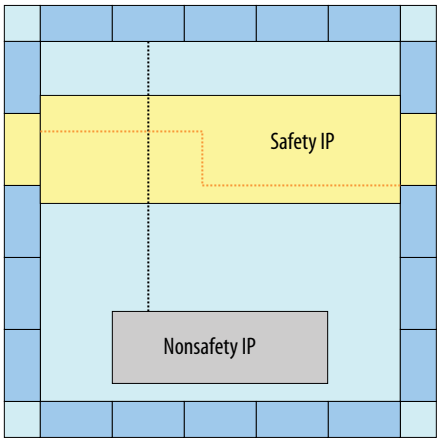
Routing Across Safety IP Partitions

The Intel Quartus Prime software strictly preserves all logic and routing configuration in safety IP.

However, you can route across a safety IP if long routing lines are available that cross over the safety IP without requiring changes to any programming bits in the safety IP area.

1. To route across a safety IP if pin and LogicLock placement restrict the routing from your nonsafety IP to the pins, adjust the pins or LogicLock region locations to free up the necessary routing.

Figure 9. Routing Across a Safety IP Partition



Design Creation Flow

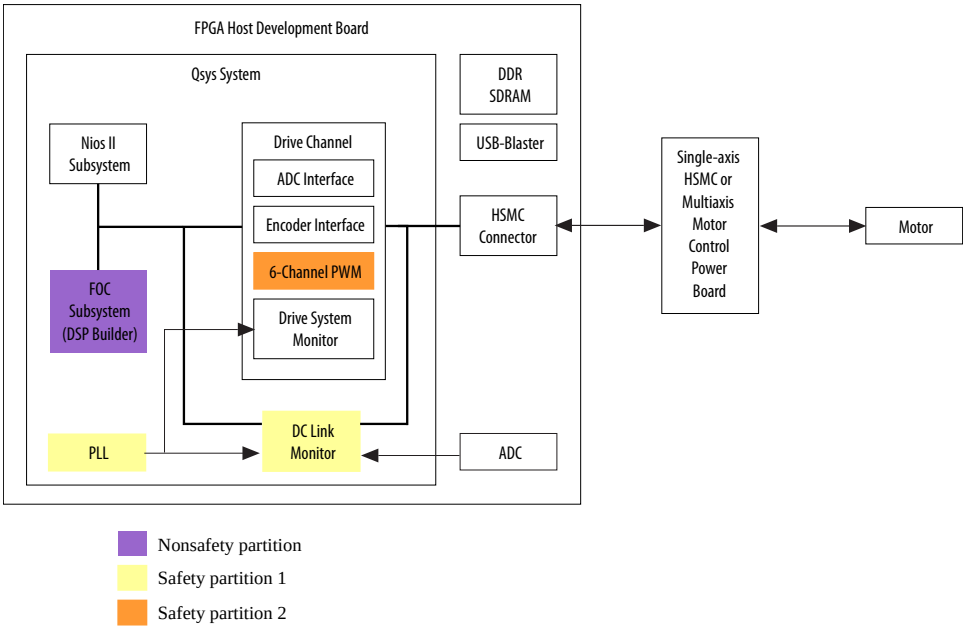
Design Hierarchy and safety IP partitions

The first stage of the design creation flow is to define the design hierarchy and safety IP partitions. You should implement every safety related IP in your design in a partition(s) so the safety IP is protected from recompilation.

To demonstrate the design flow for the motor control system example, Intel considers the following IP is safety related:

- DC Link Monitor, PLL, associated clocking structure and IO pins
- The six-channel PWM in the drive channel 0 subsystem
- FOC subsystem (nonsafety IP)

Figure 10. The Main Entities in the Design Hierarchy and the Logical Design Partitions



The design hierarchy allows design flexibility and is suitable for logical separation of the safety components in this example. Intel groups together the drive channel peripherals , independent of the FOC algorithm and the Nios II processor. This design can support many drive axes, with only a single instance of the FOC algorithm or Nios II processor.

This design specifies two safety related IP as design partitions, and one nonsafety IP partition. Using nonsafety IP partitions in the functional safety separation flow is optional.

Intel achieves strict preservation for safety IP partitions using the global `.qsf` assignment `PARTITION_ENABLE_STRICT_PRESERVATION`.

```
set_global_assignment -name PARTITION_ENABLE_STRICT_PRESERVATION
<ON/OFF> - section_id < partition_name >
```

The assignment exhibits the following properties

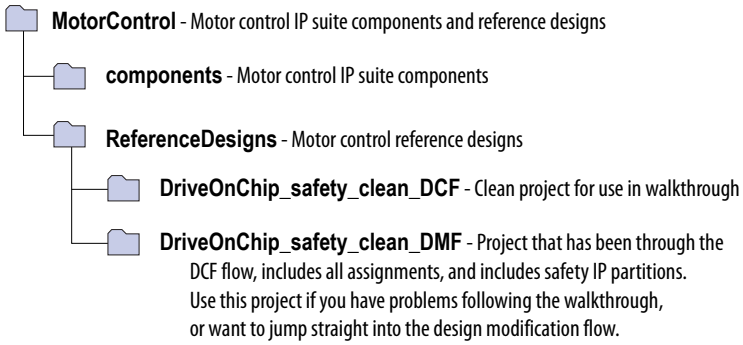
- When any partition has the assignment value **ON**, the Intel Quartus Prime software enables the safety and nonsafety separation flow
- A partition default assignment value is **OFF** (nonsafety IP). In other words, when a partition has the assignment value **OFF**, it is equivalent to not having the `PARTITION_ENABLE_STRICT_PRESERVATION` assignment and specifies that the partition is defined as nonsafety IP
- You may only assign partitions and I/O pins to safety IP.
- A partition assigned to safety IP may contain safe logic only. If you assign a parent partition to a safety IP, consider all its child partitions as part of the same safety IP.
- A design may contain several safety IP paritions. All the partitions containing logic that implements a single safety IP functionality should belong with the same top-level parent partition.

Preparing the Design Example in the Intel Quartus Prime Software

Install the Intel Quartus Prime Standard Edition software v17.0.2.

1. Obtain `an704.zip` from the Intel Functional Safety webpage and extract the files to your PC.

Figure 11. Directory Structure



2. In the Intel Quartus Prime software, open the `DOC_top.qpf` project file from the project directory
3. On the Tools menu, click **Qsys**.
4. Open the Qsys System `DOC_Single_Axis_FE2H_CVE.qsys` .
5. Click **Generate ➤ Generate HDL ➤ .**
6. In Intel Quartus Prime, before you specify any partition settings, compile the complete design: click **Processing ➤ Start ➤ Start Analysis and Elaboration**
7. View the design hierarchy in the Project Navigator.

Related Information

- Appendix B: Design Checklist on page 25
- Intel Functional Safety Web Page

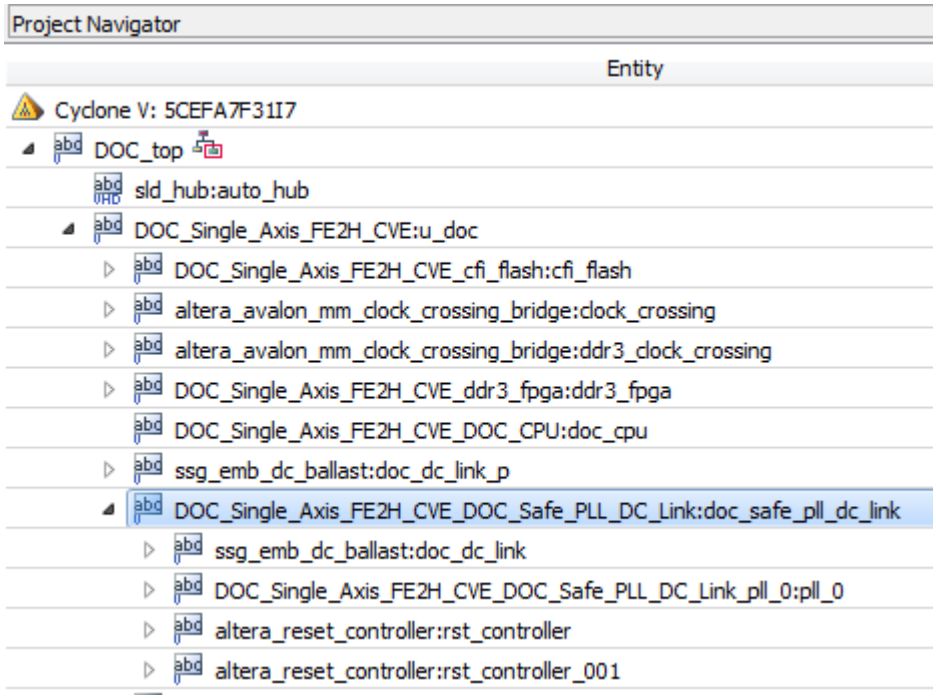
DC Link Monitor safety IP partition

A safety IP partition may contain safety logic only. The DC link monitor measures the DC Link voltage in the drive system and requests a shutdown of the system if it is out of tolerance. The block only has one data input from the sigma-delta ADC on the power board. However the ADC also requires a 20MHz clock output, generated from the FPGA PLL, to operate. Therefore, the safety IP partition includes both the DC link monitor and the PLL. In this design example, a separate Qsys subsystem contains the PLL and DC link monitor and has a safety IP partition for the created subsystem. The Qsys system exports the PLL generated clocks from the safety IP partition so other partitions may use them.

Creating a Safety IP partition for the DC Link Monitor and PLL Subsystem Component

1. In the Project Navigator, expand the **DOC_Single_Axis_FE2H_CVE:u_doc** hierarchy
2. Select the entity **DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link:doc_safe_pll_dc_link**
3. Right-click and select **Design Partition ➤ Set as Design Partition.**

Figure 12. Selecting the Entity



Note: A safety IP partition must include all IO pins that are directly connected to the partition.

4. Confirm a partition icon appears next to the **DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link:doc_safe_pll_dc_link** entity.
5. Open the Design Partition Window:

a. In the Project Navigator, select the entity **DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link:doc_safe_pll_dc_link**

b. Right-click and select **Design Partition ➤ Design Partition Window**.

6. Ensure you strictly preserve the design partition, as required by the safety separation flow:

a. In the Design Partition Window, right-click the **DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link:doc_safe_pll_dc_link** partition and select **Design Partition Properties...**

b. Set the partition netlist type to **Post-Fit**.

c. On the **Strict Preservation** tab, turn on **Allow partition to be strictly preserved for safety**.

d. On the **Advanced** tab set the **Fitter Preservation Level** to **Placement and Routing**.

e. Click **Apply** and click **OK**.

7. Assign the safety partition I/O pins to the design partition.

Note: A safety IP partition must include all IO pins that are directly connected to the partition.

a. Identify the I/O pins that are directly connected. Assign them to fixed pin locations and add these named pins to the partition by adding assignments to the project settings file (`DOC_top_FE2H_CVE.qsf`) using a text editor.

```
#Reference clock to PLL
set_instance_assignment -name ENABLE_STRICT_PRESERVATION ON -to clk_50 -
section_id
"DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link:doc_safe_pll_dc_link"

set_instance_assignment -name ENABLE_STRICT_PRESERVATION ON -to
dc_link_Sync_Dat_VBUS -section_id
"DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link:doc_safe_pll_dc_link"

set_instance_assignment -name ENABLE_STRICT_PRESERVATION ON -to
"dc_link_Sync_Dat_VBUS(n)" -section_id
"DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link:doc_safe_pll_dc_link"

#Output clocks from PLL to ADC (3 LVDS pairs)
set_instance_assignment -name ENABLE_STRICT_PRESERVATION ON -to VBus_Clk -
section_id
"DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link:doc_safe_pll_dc_link"

set_instance_assignment -name ENABLE_STRICT_PRESERVATION ON -to
"VBus_Clk(n)" -section_id
"DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link:doc_safe_pll_dc_link"

set_instance_assignment -name ENABLE_STRICT_PRESERVATION ON -to IU_Clk -
section_id
"DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link:doc_safe_pll_dc_link"

set_instance_assignment -name ENABLE_STRICT_PRESERVATION ON -to
"IU_Clk(n)" -section_id
"DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link:doc_safe_pll_dc_link"

set_instance_assignment -name ENABLE_STRICT_PRESERVATION ON -to IW_Clk -
section_id
"DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link:doc_safe_pll_dc_link"
```

```
set_instance_assignment -name ENABLE_STRICT_PRESERVATION ON -to
"IW_Clk(n)" -section_id
"DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link:doc_safe_pll_dc_link"
```

Note: Several DC link monitor block signals are connected to pins of LVDS IO standard and therefore have two pins associated with each signal e.g. `dc_link_Sync_Dat_VBUS` and `dc_link_Sync_Dat_VBUS(n)`.

Caution: All the I/O pins that connect up to a safety IP should have an explicit assignment. The Intel Quartus Prime software reports an error if:

- a pin that connects to the safety IP does not have an assignment
- a pin with an assignment does not connect to the specified safety IP.

Caution: If an `IO_REG` group contains a pin that is assigned to a safety IP, the Intel Quartus Prime software reserves all the pins in the `IO_REG` group for this safety IP. You must assign all pins in the `IO_REG` group to the same safety IP; assign none of the pins in the group to nonsafety signals.

Creating a Safety IP partition for the PWM Interface Component

1. In the Project Navigator, expand the **DOC_Single_Axis_FE2H_CVE:u_doc** hierarchy
2. Select the entity **DOC_Single_Axis_FE2H_CVE_drive0:drive0 > ssg_emb_pwm:doc_pwm**
3. Right-click and select **Design Partition ➤ Set as Design Partition.**
4. Confirm a partition icon appears next to the **doc_pwm** entity.
5. Open the **Design Partition Window**:
   a. In the Project Navigator, select the entity **DOC_Single_Axis_FE2H_CVE_drive0:drive0 > ssg_emb_pwm:doc_pwm**
   b. Right-click and select **Design Partition ➤ Design Partition Window**.
6. Ensure you strictly preserve the design partition, as required by the safety separation flow
   a. In the Design Partition Window, right-click the **ssg_emb_pwm:doc_pwm** partition and select **Design Partition Properties...**
   b. Set the partition netlist type to **Post-Fit**.
   c. On the **Strict Preservation** tab, turn on **Allow partition to be strictly preserved for safety**.
   d. On the **Advanced** tab set the "Fitter Preservation Level" to "Placement and Routing"
7. Add the following assignments to the `.qsf` file, to set the pin assignment.

```
set_instance_assignment -name ENABLE_STRICT_PRESERVATION ON -to
drive0_pwm_u_h -section_id "ssg_emb_pwm:doc_pwm"

set_instance_assignment -name ENABLE_STRICT_PRESERVATION ON -to
drive0_pwm_u_l -section_id "ssg_emb_pwm:doc_pwm"

set_instance_assignment -name ENABLE_STRICT_PRESERVATION ON -to
drive0_pwm_v_h -section_id "ssg_emb_pwm:doc_pwm"

set_instance_assignment -name ENABLE_STRICT_PRESERVATION ON -to
```

```
drive0_pwm_v_l -section_id "ssg_emb_pwm:doc_pwm"

set_instance_assignment -name ENABLE_STRICT_PRESERVATION ON -to
drive0_pwm_w_h -section_id "ssg_emb_pwm:doc_pwm"

set_instance_assignment -name ENABLE_STRICT_PRESERVATION ON -to
drive0_pwm_w_l -section_id "ssg_emb_pwm:doc_pwm"
```

Creating a Safety IP LogicLock Region for the DC Link Monitor

To fix the safety IP into specific areas of the device, define LogicLock regions. By using preserved LogicLock regions, the Intel Quartus Prime software reserves device placement for the safety IP. LogicLock regions prevent the Intel Quartus Prime software from placing nonsafety IP into the unused resources of the safety IP region. You establish a fixed size and origin to ensure location preservation.

1. Create a new LogicLock region.
   a. In the Project Navigator, right-click on the entity **DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link:doc_safe_pll_dc_link**
   b. Click **LogicLock Region ➤ Create New LogicLock Region**
2. Open the LogicLock Regions window:
   a. In the Project Navigator, select the entity `DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link:doc_safe_pll_dc_link`
   b. Right-click and select **LogicLock Region ➤ LogicLock Regions Window**.
3. Set a fixed size and origin for the safety IP partition:
   a. In the **LogicLock Regions** window, right-click the region `DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link:doc_safe_pll_dc_link`
   b. Select **LogicLock Regions Properties**
   c. Turn on **Reserved (Prevent Fitter from placing non-member logic in region)**.
   d. On the **Size and Origin** tab, choose a fixed size and origin for the region. You may select your own values, however width = 7, height = 5, origin = X7_Y34 are suitable values.

Creating a LogicLock Region for the PWM Interface

1. Create a new LogicLock region.
   a. In the Project Navigator, right-click on the entity **DOC_Single_Axis_FE2H_CVE_ drive0:drive0 > ssg_emb_pwm:doc_pwm**
   b. Click **LogicLock Region ➤ Create New LogicLock Region**
2. Open the LogicLock Regions window:
   a. In the Project Navigator, select the entity `DOC_Single_Axis_FE2H_CVE_drive0:drive0 > ssg_emb_pwm:doc_pwm`
   b. Right-click and select **LogicLock Region ➤ LogicLock Regions Window**.
3. Set a fixed size and origin for the safety IP partition:

a. In the **LogicLock Regions** window, right-click the region `DOC_Single_Axis_FE2H_CVE_ drive0:drive0 > ssg_emb_pwm:doc_pwm`

b. Select **LogicLock Regions Properties**

c. Turn on **Reserved (Prevent Fitter from placing non-member logic in region)**.

d. On the **Size and Origin** tab, choose a fixed size and origin for the region. You may select your own values, however width = 7, height = 5, origin = X74_Y45 are suitable values.

Creating a Fixed Size and Origin for a LogicLock Region

The LogicLock size and origin for every component is different. A safety IP partition must have a fixed size and origin for its LogicLock region, otherwise the fitter gives an error. You can run a trial Intel Quartus Prime compilation with the LogicLock region set to Auto size and floating origin and then use this result to fix the region:

1. Temporarily turn off **Allow partition to be strictly preserved for safety** in the **Design Partitions Properties...** window
2. Set the LogicLock region to **Auto size and floating origin**.
3. Compile the design.
4. Open the **LogicLock Regions Properties** window again and then select **Set Size and Origin to Previous Fitter Results**.
5. Turn on **Allow partition to be strictly preserved for safety** in the **Design Partitions Properties...** window

Removing Precomiled Netlists

If you are using the design creation flow, you should remove any previously compiled netlists for safety IP partitions before recompiling the design. You must remove precompiled netlists, otherwise:

- If the source code changes for the safety IP and you turn on the **Ignore changes in source files and strictly use the specified netlist, if available** design partition property, the Intel Quartus Primesoftware uses the previous netlist and does not include the changes in the safety IP.
- If the source code does not change for the safety IP, you may see a no-fit error. The safety IP partition uses the design modification flow and restricts the place and route unnecessarily.

1. Right-click on the safety IP partition in the **Design Partitions** window,
2. Select **Advanced ➤ Delete Netlists**.

The Intel Quartus Prime software then recompiles the safety IP partitions from the source files and the fitter can reroute and place the safety IP partition.

Using the Intel Quartus Prime Incremental Compilation

Using this feature on the nonsafety IP partitons in the design does not interfere with strictly preserved safety IP partitions.

Creating a Partition for the FOC Fixed-Point Component

1. In the Project Navigator, expand the `DOC_Single_Axis_FE2H_CVE:u_doc` hierarchy.
2. Select the entity `DOC_Single_Axis_FE2H_CVE_FOC_fixed_point:foc_fixed_point`.
   a. Right-click and select **Design Partition ➤ Set as Design Partition**.
   b. Confirm a partition icon appears next to the `foc_fixed_point` entity.

Creating a LogicLock Region for the FOC Fixed-Point Component

1. Create a New LogicLock Region:
   a. In the Project Navigator, right-click on the entity `DOC_Single_Axis_FE2H_CVE_FOC_fixed_point:foc_fixed_point`
   b. Select **LogicLock Region ➤ Create New LogicLock Region**.

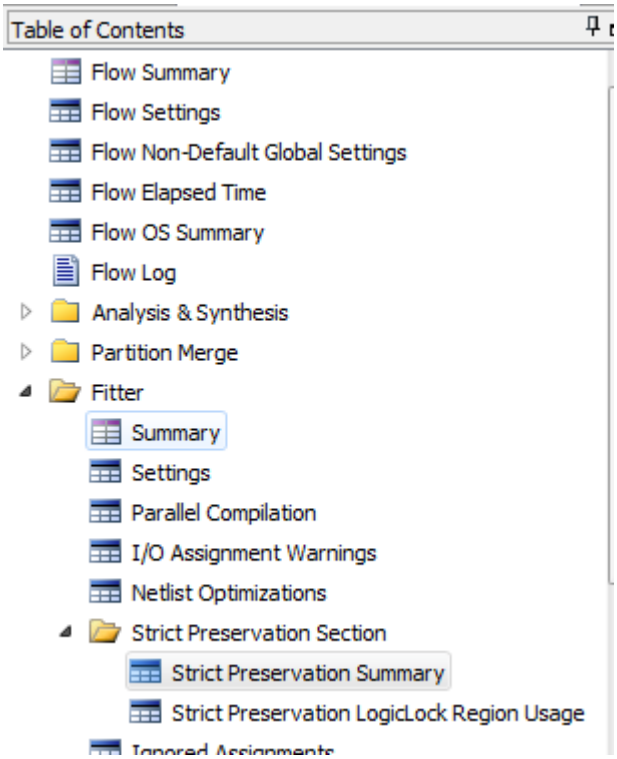
Nonsafety IP partitions have no special restrictions on the partition and LogicLock settings. In the example, the partition netlist type is set to **Source File** and the LogicLock region is set to **Auto**.

Compiling the Design

1. In the Intel Quartus Prime software, click **Processing ➤ Start Compilation**

When the design successfully compiles, the Intel Quartus Prime software displays the safety and nonsafety IP information in the fitter section of the Compilation Report under the **Strict Preservation Section**.

Figure 13. Strict Preservation Section



The Fitter Report

The Fitter report includes information for each safety IP and the respective partition and I/O usage.

The report contains the following information:

- Partition name (with the name of the top level safety IP partition used as the safety IP name)
- Effective design flow in use, which indicates either design **creation** flow or design **modification** flow.
- Number of safety or nonsafety inputs to the partitions
- Number of safety or non-safety outputs to the partitions
- LogicLock region names along with size and locations for the regions
- I/O pins used for the respective safety IP in your design
- Safety related error messages

Exporting Safety IP Partition

Save the safety IP partition placement and routing information for use in any subsequent design modification flow. Saving the partition information enables you to import the project to a clean Intel Quartus Prime project where no previous compilation results exist.

1. Right click on the partition `DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link:doc_safe_pll_dc_link` in the **Design Partitions Window**
2. Select **Export Design Partition...**

   Note: Ensure that you turn on **Post-fit netlist** and **Export routing** and you turn off **Post-synthesis netlist**. Attempting to export a synthesis netlist for a safety IP partition gives an error.
3. Click **OK** to export the partition, generating a Intel Quartus Prime exported partition file (`.qxp`).
4. Repeat for the safety IP partition, `ssg_emb_pwm:doc_pwm`.

Generating Safety IP Bitstream Files

The design modification flow requires a safety IP bitstream file, known as a partially preserved bitstream. The separate safety IP partitioning verification tool reads the partially preserved bitstream file to verify that no change occurs to the state of safety IP regions (i.e. whether the safety region is unchanged) or other relevant device level configuration options.

Note: Run any command-lines below from a **Nios II 17.0 Command Shell** with the current directory set to the `Quartus Prime project output_files` directory.

1. Post process the bitstream file (`.sof`) generated by the Intel Quartus Prime assembler, to create the partially preserved bitstream file using the following command

   ```
   quartus_cpf --genppb <partitionname>.psm  <projectname>.sof
   <partitonname>.rbf.ppb
   quartus_cpf -c <partitionname>.psm  <partitonname>.rbf.ppb
   ```

   The following commands generate the partially preserved bitstream for two safety IP partitions.

   ```
   quartus_cpf --genppb DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link-
   doc_safe_pll_dc_link.psm  DOC_top_FE2H_CVE.sof
   DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link-doc_safe_pll_dc_link.rbf.ppb

   quartus_cpf -c DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link-
   doc_safe_pll_dc_link.psm   DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link-
   doc_safe_pll_dc_link.rbf.ppb

   quartus_cpf --genppb ssg_emb_pwm-doc_pwm.psm  DOC_top_FE2H_CVE.sof
   ssg_emb_pwm-doc_pwm.rbf.ppb

   quartus_cpf -c ssg_emb_pwm-doc_pwm.psm  ssg_emb_pwm-doc_pwm.rbf.ppb
   ```

   During partially preserved bitstream file generation, the Intel Quartus Prime software generates an additional checksum file `<partitionname>.md5.sign`.
2. Archive the generated `.psm`, `.ppb` and `.md5.sign` files for use later in the design modification flow. For this example, create a `design creation flow` directory in the `output_files` directory and copy the following files to it:

   — `DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link-doc_safe_pll_dc_link.md5.sign`

- `DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link-doc_safe_pll_dc_link.psm`
- `DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link-doc_safe_pll_dc_link.rbf.ppb`
- `ssg_emb_pwm-doc_pwm.md5.sign`
- `ssg_emb_pwm-doc_pwm.psm`
- `ssg_emb_pwm-doc_pwm.rbf.ppb`
- Quartus settings (`.qpf`, `.qsf`)
- HDL Source code, IP, Qsys project (design specific)
- Exported partition netlists (for safe and other post-fit partitions) (`.qxp`)
- Programming file (`.sof`)
- Additional safety IP bitstream files (`.psm`, `.ppb`, `,md5.sign`)

Note: The Intel Quartus Prime archiver does not include all these file types by default. You must ensure all necessary files are archived.

Note: When unarchiving, use a commonly available MD5 checksum utility (e.g. md5sum shipped with Cygwin in the) to regerenate the MD5 checksum of the `.rbf.ppb` and `.psm` files and compare against those stored in the `md5.sign` file to check the files for any corruption.

Generating Complete FPGA Bitstream File

1. When generating the final FPGA programming bitstream with quartus_cpf, use the **generate_signature** option to generate an `MD5.sign` file containing the MD5 checksum. Store the checksum as evidence of the flow

   .e.g. `quartus_cpf -c -o generate_signature=on <design>.sof <fpgabitstream>.rbf`
2. The final FPGA programming bitstream generated by quartus_cpf may be encrypted or unencrypted. The partialy preserved bitstream files (`.rbf.ppb`) generated by quartus_cpf are always unencrypted. To encrypt a bitstream use the quartus_cpf `--key` option.

   e.g. `quartus_cpf -c -o generate_signature=on --key test.key:key1 <design>.sof <encryptedfpgabitstream>.rbf`

Using the Design Modification Flow

Use this flow if you improve an algorithm, or fix a bug, or add a new feature in the nonsafety IP. For the design modification flow, you may change any of the nonsafety IP partitions of the design if the safety IP partitions remain unchanged. If you need to make any changes to a safety IP partition, you must use the design creation flow.

For example, use this flow to:

- Change parameters for the FOC algorithm component (algorithm improvement or bug fix).
- Add a timer component to the Nios II system (adding a new feature).

1. Change the nonsafety IP partitions of the design.
2. Import safety IP partitions. Turn on **Design Partition Window** to ensure the Intel Quartus Prime software preserves the the safety IP place and route that it saves in the design creation flow stage.
3. Recompile the complete design.

   Note: If the Intel Quartus Prime assembler gives an internal error message, there may be a mismatch between the safety IP bitstream in the previously generated `.sof` file and the `.sof` from the current compilation. If the error is because of a mismatch, move or rename the `.sof` file to allow you to rerun the assembler without the internal error. You may then continue to run the partial bitstream comparison where the comparison is expected to fail with an explicit message.
4. Post process the `.sof` file and `.psm` files that the assembler generates to create the partially preserved bitstream file for the safety IP regions.
5. Verify that the Intel Quartus Prime software preserves the strictly preserved partition successfully by using the functional safety POF partition verification tool to compare the `.ppb` file created in the design creation flow and design modification flows for each safety IP partition.

   ```
   sppv --device=<device name from qsf> [<options>] <design creation flow
   directory>/<partition-name>.rbf.ppb <design modification flow directory>/
   <partitionname>.rbf.ppb
   ```

   Intel includes the functional safety POF partition verification tool in the functional safety data pack but not with the Intel Quartus Prime Design Suite

   The verification tool generates a report file named <*partition-name*>.rbf.ppb.rpt
6. Alternatively, if you do not have the functional safety POF partition verification tool, compare the `MD5.sign` checksum files.

   Note: The checksum also covers some device option bits that are legal to change without compromising strict preservation. If these bits change the checksums do not match. In this case, use the functional safety POF partition verification tool to detect the location of the mismatch. Commonly, this mismatch is the Intel Quartus Prime auto usercode feature (enabled by default). Use the following `.qsf` setting to disable the auto usercode feature:

   ```
   set_global_assignment -name USE_CHECKSUM_AS_USERCODE OFF
   ```

   The safety assessor may archive the `md5.sign` checksum file for each safety IP when they initially assess a design creation flow design. To verify that the design modification flow uses the `.ppb` file that the design creation flow generates, check for a match between the checksum that the functional safety POF partition verification tool reports and the checksum in the `.md5`.sign file both match The functional safety POF partition verification tool then reports the comparison results between the design creation flow and design modification flow as evidence of strict preservation.

Related Information

Generating IP Bitstreams on page 21

Changing Nonsafety IP - Changing FOC Algorithm Parameter

To demonstrate modifying a nonsafety IP, change a parameter in the FOC algorithm block in Qsys.

1. Open Qsys.
2. In the **System Contents** tab, scroll to the **FOC_fixed_point** component in the component list, double click the component to open the **Parameters** window, and change the **Number of Channels** to **1**.
3. Regenerate the Qsys system.

Related Information

Design Modification Flow on page 6

Adding New Features to Nonsafety IP

To demonstrate adding a new feature to a nonsafety IP, add a timer function to the Qsys system. The timer function connects up to Nios processor in the nonsafety IP partition and to clocks created from the PLL in the safety IP partition.

1. Open Qsys.
2. In the **System Contents** tab, scroll to the bottom of the component list and turn on **timer_0** component (select the checkbox in the Use column).
3. Regenerate the Qsys system.

Related Information

Design Modification Flow on page 6

Importing Safety IP Partition

1. Right click on the partition `DOC_Single_Axis_FE2H_CVE_DOC_Safe_PLL_DC_Link:doc_safe_pll_dc_link` in the **Design Partitions Window**.
2. Select **Import Design Partition...**.
3. Select the Intel Quartus Prime exported partition file, `DOC_Single_Axis_FE2H_CVE_DOC_Saf.qxp`, which you exported in step 1.
4. Click **OK** to import the partition.
5. Repeat for the safety IP partition, `ssg_emb_pwm:doc_pwm`.

By default the Intel Quartus Prime software keeps the previous compilation partition data in its database. Only perform the partition import step when you perform a clean Intel Quartus Prime compilation or upgrade to a newer release of the Intel Quartus Prime software.

Appendix A: Terminology

Table 1. Terminology

Term	Description
LogicLock region	A LogicLock region is a physical partition or type of placement constraint in the Intel Quartus Prime software. You can define any arbitrary region of physical resources on the target device as a LogicLock region. A LogicLock region can have the following size and location settings: • Fixed size, locked location • Fixed size, floating location • Auto-size, floating location
Validation	The DUT is performing the correct operation versus the high-level requirements.
Verification	The DUT operation is correct versus the module design and test specification.

Appendix B: Design Checklist

Design Phase	Flow		Actions
	Design Creation Flow	Design Modification Flow	
General	Yes	Yes	Have you installed ? All tool log messages should include "`Info: Version 17.0.2 Build 602 07/19/2017 SJ Standard Edition`".
General	Yes	Yes	If you use a JTAG Master in the safety partition, have you added `sci_use_legacy_sld_flow=on` to `quartus.ini` as described in Knowledge Base solution ID rd07012015_904.
Place and Route	Yes	Yes	Have you assigned a strict preserved partition and reserved LogicLock region with fixed size and position to the safety IP?
Place and Route	Yes	Yes	Do not use the `LL_MEMBER_EXCEPTIONS` assignment on a safety IP partition, otherwise all the logic is not inside the LogicLock region. The Intel Quartus Prime software should give an error in this condition. In the , you must ensure the project does not use this assignment. Searching the project `.qsf` file for the following assignment or look at the Members column in the LogicLock **Regions** window. `set_instance_assignment -name LL_MEMBER_EXCEPTIONS <exception-list> -to <safe-partition-name>`
Place and Route	Yes	Yes	Have you manually reviewed all `ENABLE_STRICT_PRESERVATION` assignments in the Intel Quartus Prime project `.qsf` file to ensure they are all set correctly? • An `ENABLE_STRICT_PRESERVATION ON` assignment is made for every safe IO pin connected to the safety IP • An `ENABLE_STRICT_PRESERVATION OFF` assignment is made for every non-safe IO pin connected to the safety IP • An `ENABLE_STRICT_PRESERVATION` assignment is not made to an IO pin connected to a non-safety IP. The Intel Quartus Prime software checks that all IO pins connected to a safety IP have explicit `ENABLE_STRICT_PRESERVATION` assignments. If the assignment is missing the software gives an error message. The Intel Quartus Primesoftware ignores any `ENABLE_STRICT_PRESERVATION` assignments that you make to IO pins that are not connected to a safety IP.
Place and Route	Yes	Yes	Did the Intel Quartus Prime software apply the expected strict preservation settings to the safety IP partition and execute the expected strict preservation flow?

continued...

Design Phase	Flow		Actions
	Design Creation Flow	Design Modification Flow	
			Have you checked the Intel Quartus Prime Fitter report strict preservation sub-section to confirm that LogicLock regions, partitions, and I/O assignments are correct?
Place and Route	Yes	No	Have you exported the post-fit netlist and routing information for the safety IP to `.qxp` file using the Intel Quartus Prime export parition feature? For example, check the datestamp on the <*safety IP partition name*>`.qxp` file is correct.
Place and Route	No	Yes	Have you imported the post-fit netlist and routing information for the safety IP using the Intel Quartus Prime import parition feature from the `.qxp` file genertated during the design control flow phase ?
Place and route	Yes	No	Before running the design creation flow, have you removed all previous compilation netlists and `.sof` files? Otherwise the design modification flow runs.
Bitstream Generation	Yes	Yes	After successful compilation, has the Intel Quartus Prime software generated a `.psm` file for the safety IP? Check the datestamp on the <*safety IP partition name*>`.psm` file is correct.
Bitstream Generation	Yes	Yes	Have you generated the `.rbf`, `.ppb`, and `.md5.sign` files for each safety IP using the `quartus_cpf` utlity?
Bitstream Generation	No	Yes	Have you run the functional safety POF verification tool to compare the safety IP bitstreams between design creation and design modification flows? • Check the .rbf.ppb.rpt file for correct tool versions, file datestamps, and error or warnings messages. • Any warnings detected by the functional safety POF verification tool should be reviewed to check they are expected and do not impact the design. Refer to the *Functional Safety POF Comparison Tool User Guide* for information about functional safety POF verification tool usage and messages. • Check that there is no inadvertent file corruption of the .psm and .rbf.ppb files. You should run an MD5 checksum on those files and compare the checksums against those stored in the .md5.sign file. You may use the 'md5sum' (or equivalent) utility shipped with ACDS Cygwin to recreate the checksum.
Bitstream Generation	No	Yes	Does the Intel Quartus Prime assembler give an internal error message, which may indicate a mismatch in the safety IP bitstream?
Verification	Yes	Yes	For each safety IP partition, have you prepared the following evidence for both design creation and design modification flows? • `<design-name>.sof` • `<partition-name>.psm` • `<partition-name>.rbf.ppb` • `<partition-name>.rbf.ppb.rpt` • `<partition-name>.md5.sign`
Verification	Yes	No	Have you archived the `.qxp`, `.rbf.ppb`, `.psm`, `.sof` and `.md5.sign` files to prevent the Intel Quartus Prime software overwriting them when you subsequently run a design modification flow compilation?
Verification	Yes	Yes	Have you archived the project on sucessful completion and verification of design creation flow and design modification flow compilations?

Related Information

Knowledge Base solution ID rd07012015_904

Document Revision History for AN 704: FPGA-based Safety Separation Design Flow for Rapid Functional Safety Certification

Version	Changes
2018.09.01	• Added support for Intel MAX 10 devices • Updated to Intel Quartus Prime Standard Edition v17.0 update 2
2015.12.01	Updated for Altera Complete Design Suite v14.1 update 1.
2015.04.15	Updated for Altera Complete Design Suite v13.1 update 4.
2014.06.24	Initial release.