Altera Advanced SEU Detection
(ALTERA_ADV_SEU_DETECTION) IP Core
User Guide
1. About the Altera Advanced SEU Detection IP Core

You can select and configure the Altera Advanced SEU Detection IP core through the IP Catalog and parameter editor in the Quartus® II software.

For more information about Altera IP cores and the IP Catalog, refer to Introduction to Altera IP Cores.

Features

The Altera Advanced SEU Detection IP core contains the following features:

- Hierarchy tagging—Enables tagging of logical hierarchies and specifying their criticality relative to SEU. This feature is available only for Arria® V, Cyclone® V, Stratix® V, and later device families.

- Sensitivity processing—Determines the criticality of an SEU detected and located by error detection cyclical redundancy check (EDCRC) hard IP. This feature includes on and off-chip sensitivity processing.

Device Family Support

The Altera Advanced SEU Detection IP core supports the following device families:

- Arria V
- Arria V GZ
- Cyclone V
- Stratix IV
- Stratix V
Stratix IV devices contain a 16-bit cyclic redundancy check (CRC) value per CRAM frame, and Arria V, Cyclone V, Stratix V, and later device families contain a 32-bit CRC value per CRAM frame. The CRC value allows the configuration engine to determine the SEU location. The Quartus II software can generate a Sensitivity Map Header File (.smh) of the configuration regions of your design that are sensitive to SEU.

You can instantiate the Altera Advanced SEU Detection IP core with the following configurations:

- "On-Chip Lookup Sensitivity Processing"—Error location reporting and lookup performed by the FPGA.
- "Off-Chip Lookup Sensitivity Processing"—Error location lookup determined by an external unit (such as a microprocessor).

**On-Chip Lookup Sensitivity Processing**

All device families that support SEU detection include a hard error detection block that detects soft errors and provides the location of single-bit errors, and double-bit adjacent errors for supported devices. The Altera Advanced SEU Detection IP core interprets the error detection register of the error detection block, and then compares single-bit error locations with a sensitivity map. This check determines whether or not the failure affects the device operation.

For more information about the hard EDCRC feature, refer to the SEU Mitigation chapter in the Arria V, Cyclone V, Stratix IV, or Stratix V device handbooks.

**Figure 2–1. System Overview for On-Chip Lookup Sensitivity Processing**

The Altera Advanced SEU Detection IP core accepts the content of the error message register (EMR) and issues a query to an external memory containing the sensitivity map. The system designer is responsible for the memory access logic and external memory.
Altera recommends that you implement an SEU detection circuit that tolerates a soft error in its logic by instantiating two instances of the Altera Advanced SEU Detection IP core in your design. In this case, one instance of the IP core flags errors that occur in the other instance of the IP core as “critical.”

**Figure 2–2. ALTERA Advanced SEU DETECTION IP Core Signals for On-Chip Processing**

Table 2–1. Altera Advanced SEU DETECTION IP Core Signals for On-Chip Processing

<table>
<thead>
<tr>
<th>Signals</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>clk</strong></td>
<td>Input</td>
<td>1</td>
<td>Clock input. Recommended frequency is 100 MHz or higher.</td>
</tr>
<tr>
<td>reset</td>
<td>Input</td>
<td>1</td>
<td>Active-high reset.</td>
</tr>
<tr>
<td><strong>cache_comparison_off</strong></td>
<td>Input</td>
<td>1</td>
<td>Static input signal. Commands the IP core to bypass cache comparison. You can use this signal with the internal scrubbing feature for custom design.</td>
</tr>
<tr>
<td><strong>emr[66:0]</strong></td>
<td>Input 67</td>
<td></td>
<td>Error Message Register data input from the Altera Error Message Register Unloader IP core.</td>
</tr>
<tr>
<td>emr_valid</td>
<td>Input</td>
<td>1</td>
<td>Indicates when emr data input is valid.</td>
</tr>
<tr>
<td><strong>emr_error</strong></td>
<td>Input</td>
<td>1</td>
<td>Indicates when emr data will be ignored due to an error. This may occur when there is a data overrun from the Altera Error Message Register Unloader IP core.</td>
</tr>
</tbody>
</table>
The Altera Advanced SEU Detection IP core interprets the content of the error detection block’s EMR and presents information to a system processor, which determines whether the failure affects the device operation. The system processor implements the algorithm to perform a lookup against the .smh.

### Table 2–1. Altera Advanced SEU DETECTION IP Core Signals for On-Chip Processing

<table>
<thead>
<tr>
<th>Signals</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>noncritical_error</td>
<td>Output</td>
<td>1</td>
<td>Indicates that an SMH lookup determined that the EDCRC error is in a non-critical region.</td>
</tr>
<tr>
<td>critical_error</td>
<td>Output</td>
<td>1</td>
<td>Indicates that an SMH lookup determined that the EDCRC error is in a critical region.</td>
</tr>
</tbody>
</table>
| regions_report     | Output | 1     | - The ASD region for the error, as reported by the SMH lookup.  
                   |        |       | - The width of this port comes from the setting for the parameter “Largest ASD region ID used.” |

### External Memory Avalon-MM Master Interface Signals

<table>
<thead>
<tr>
<th>Signals</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| mem_addr     | Output | Output to the user logic.  
                   |       | Byte address of the 32-bit word to be read. |
| mem_rd       | Output | Output to the user logic.  
                   |       | Signals to the user logic to request a read operation. |
| mem_bytesel  | Output | Output to the user logic.  
                   |       | A four-bit signal that selects the bytes needed by the IP core. Use of this signal allows 16-bit or 8-bit memories to optimize the number of reads in cases where the IP does not need all 32 bits. If bit 0 of mem_bytesel is 0, then the IP core ignores bits 0 to 7 of mem_data, and similarly for bits 1 to 3 of mem_bytesel. |
| mem_wait     | Input  | Input from the user logic.  
                   |       | Signals to the memory interface that the read operation is still running. Must be high by the first rising clock after mem_rd is asserted to hold the IP core in a wait state. |
| mem_data     | Input  | Input from the user logic.  
                   |       | 32-bit data bus. Data must be present if mem_wait goes high and if mem_rd returns low. |
| mem_datavalid| Input  | Input from the user logic.  
                   |       | Signals that the mem_data signal contains valid data in response to a previous mem_rd request. |

Note (1): The Avalon (ST) Streaming Sink Interface should be connected to the corresponding Avalon-ST Source Interface of the EMR Uploader IP Core.
The off-chip lookup sensitivity processing consists of two components:

- Design logic to interpret content of the EMR of the CRC block and present the information to a processor interface.
- Cache to store off-loaded content of the EMR.

For information about the .smh, refer to “SMH Lookup” on page 2–7.

For more information about the EMR, refer to the SEU Mitigation chapter in the Arria V, Cyclone V, Stratix IV, or Stratix V device handbooks.

The EMR processing unit interprets the content of EMR offloaded from the CRC block by the EMR Uploader IP core upon an SEU. The EMR processing unit writes each unique EMR value into cache, until the cache is full. After the cache is full, it asserts a cache overflow flag to the system interface.

For each new value written into cache, the EMR processing unit asserts an interrupt to the processor. The system processor reads the EMR value and performs a lookup against the .smh to determine the criticality of a CRAM location. After the system processor services the interrupt, the EMR processing unit advances the cache line and generates additional interrupt assertions, provided that there is an EMR value in cache that has not been processed.

After SMH lookup, the system processor determines the required corrective response.
Off-chip sensitivity processing has similar signals with on-chip sensitivity processing, with the exception of the external memory interface; the off-chip sensitivity processing has EMR cache interface instead.

**Figure 2–4. ALTERA Advanced SEU DETECTION IP Core Signals for Off-Chip Sensitivity Processing**

![Diagram of ALTERA Advanced SEU DETECTION IP Core Signals](image)

**Table 2–2. Altera Advanced SEU Detection IP Core Signals for Off-Chip Processing**

<table>
<thead>
<tr>
<th>Signals</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clk</td>
<td>Input</td>
<td>1</td>
<td>Clock input. Recommended frequency is 100 MHz or higher.</td>
</tr>
<tr>
<td>reset</td>
<td>Input</td>
<td>1</td>
<td>Active high reset.</td>
</tr>
<tr>
<td>cache_comparison_off</td>
<td>Input</td>
<td>1</td>
<td>Static input signal. Commands the IP to bypass cache comparison.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>You can use this signal with the internal scrubbing feature for custom design.</td>
</tr>
<tr>
<td>emr[66:0]</td>
<td>Input</td>
<td>67</td>
<td>Error Message Register data input from the Altera Error Message Register Unloader IP core.</td>
</tr>
<tr>
<td>emr_valid</td>
<td>Input</td>
<td>1</td>
<td>Indicates when emr data input is valid.</td>
</tr>
<tr>
<td>emr_error</td>
<td>Input</td>
<td>1</td>
<td>Indicates when emr data will be ignored due to an error.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This may occur when there is a data overrun from the Altera Error Message Register Unloader IP core.</td>
</tr>
<tr>
<td>emr</td>
<td>Input</td>
<td>67</td>
<td>Error Message Register data from the Altera Error Message Register Unloader IP core.</td>
</tr>
</tbody>
</table>
Table 2–2. Altera Advanced SEU Detection IP Core Signals for Off-Chip Processing

<table>
<thead>
<tr>
<th>Signals</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>emr_valid</td>
<td>Input</td>
<td>1</td>
<td>Indicates when emr data input is valid.</td>
</tr>
<tr>
<td>emr_error</td>
<td>Input</td>
<td>1</td>
<td>Indicates when emr data input will be ignored due to an error.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This may occur when there is a data overrun from the Altera Error Message</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Register Unloader IP core.</td>
</tr>
</tbody>
</table>

**EMR Cache Avalon-ST Source Interface Signals**

| cache_data         | Output     | 34    | Error cache data.                                                           |
|                   |            |       | This is the location information for an EMR cache entry.                    |
| cache_valid        | Output     | 1     | Indicates when the cache_data contents are valid.                           |
| cache_ready        | Input      | 1     | Indicates that the reader of the Avalon Stream interface is ready.          |
| cache_error        | Output     | 1     | This Avalon stream control signal indicates the current transfer is in error|
|                    |            |       | and should be ignored.                                                     |

**Cache Status Signals**

| cache_fill_level   | Output     | 4     | Indicates how many entries are in the cache.                               |

**Errors Output Interface Signals**

| critical_error     | Output     | 1     | Indicates that an SMH lookup determined that the EDCRC error is in a critical region. |

Note (1): The Avalon (ST) Streaming Sink Interface should be connected to the corresponding Avalon-ST Source Interface of the EMR Uploader IP Core.
SMH Lookup

The `.smh` file represents a hash of the CRAM bit settings on a design. Related groups of CRAM are mapped to a signal bit in the sensitivity array. During an SEU event, a design can perform a lookup against the `.smh` to determine if a bit is used. By using the information about the location of a bit, you can reduce the effective soft error rate in a running system.

The following criteria determine the criticality of a CRAM location in your design:

- **Routing**—All bits that control a utilized routing line
- **Adaptive logic modules (ALMs)**—If you configure an ALM, then the megafuction considers all CRAM bits related to that ALM sensitive.
- Logic array block (LAB) control lines—If you use an ALM in a LAB, then the megafunction considers all bits related to the control signals feeding that LAB sensitive.

- M20K memory blocks and digital signal processing (DSP) blocks—If you use a block, then the megafunction considers all CRAM bits related to that block sensitive.

**Types of SMH Files**

The .smh is an Intel-format Hexadecimal file. You can generate two revisions of .smh files:

- Revision 1—Generated for Stratix IV family devices. This revision does not support hierarchy tagging, and does not contain tag size or region map information.

- Revision 2—Generated for Arria V, Cyclone V, and Stratix V, and later device families. The generated .smh contains tag size and region map information.

**Figure 2–6. Revision 1 Sensitivity Map Header**

In revision 1 files, the sensitivity map header starts from 160-bit header information that provides basic information about the .smh format. This includes the base addresses for the frame information, offset maps, and the sensitivity data array, and length of the single offset map.
Frame information array:
The frame information array contains a 32-bit string for each frame in the device. The frame number serves as the index for the frame information string. Each frame information string provides the following information:
- The index for the offset map array that this frame uses. (offset_map_array_index) (length = 8 bits)
- A 24-bit address offset into the sensitivity array for this frame. (frame_info_data_offset) (length = 24 bits)
- Bits 7:0 of the frame information string form the offset map array index.
- Bits 31:8 of the frame information string form the offset into the sensitivity array.

Offset map array—The offset map information array is a set of arrays containing 16-bit offset maps. Each offset map value represents an additional offset into the sensitivity array for a frame group. Each offset map value is 16 bits. The offset_map_length string in the header information defines the size of each offset map array.

Sensitivity data array—The sensitivity data array is a flat-bit vector where 1 specifies a sensitive bit and 0 specifies an insensitive bit.
In revision 2 files, the sensitivity map header is an extension of revision 1 header format. The header information provides basic information about the .smh revision 2, and includes all the revision 1 header information fields. The additional fields include size of the sensitivity data tag size in bits, base addresses for the region map, and 32-bit CRC signature of the corresponding .sof file.

The 32-bits ID of the sensitivity map header revision 2 is defined as follows:

- Bits 23:0 is Altera sensitivity map header ID 0x445341
- Bits 24:31 is a bit mask for the header information with bit 24 reserved
- Bit 25 indicating presence of sensitivity tag information in the .smh
- Bits 27:26 reserved
- Bit 28 indicating presence of 32-bit CRC signature of corresponding .sof
- Bits 29:31 reserved

Frame Information Array

- The frame information array contains a 32-bit string for each frame in the device. The frame number serves as the index for the frame’s information string. Each Frame Information string provides the following information:
  - The index for the offset map array that this frame uses
    (offset_map_array_index) (length = 8 bits)
  - A 24 bit address offset into the sensitivity array for this frame.
    (frame_info_data_offset) (length = 24 bits)
  - Bits 7:0 of the frame information string form the offset map array index.
  - Bits 31:8 of the frame information string form the offset into the sensitivity array.

Offset map information array—The offset map information array is a set of arrays containing 16 bit offset maps. Each offset map value represents an additional offset into the sensitivity array for a frame group. Each offset map value is 16 bits. The size of each offset map array is defined by the offset_map_length string contained in the header information.

Sensitivity data array—The size of the single sensitivity data entry or tag (sensitivity_data_tag_size) is in bits and aligned to power of 2. The sensitivity data array is a flat sensitivity tag vector where a sensitive tag of 0 specifies a bit insensitive for all regions, and non-zero tag specifies an offset into region map.

Region map information array—The region map information array contains a 16-bit string for each non-zero sensitivity tag. The sensitivity data tag serves as the index-1 for the region map array. The string is a bitmask of the regions, the bit is sensitive for. Each region can be identified in the bitmask by mask 1 << (Region ID - 1).
3. Getting Started with Altera IP Cores

Installing and Licensing IP Cores

The Quartus II software includes the Altera IP Library. The library provides many useful IP core functions for production use without additional license. You can fully evaluate any licensed Altera IP core in simulation and in hardware until you are satisfied with its functionality and performance.

Some Altera IP cores, such as MegaCore® functions, require that you purchase a separate license for production use. After you purchase a license, visit the Self Service Licensing Center to obtain a license number for any Altera product. For additional information, refer to Altera Software Installation and Licensing.

Figure 3–1. IP core Installation Path

The default installation directory on Windows is \altera\version number; on Linux it is <home directory>/altera/version number.

Customizing and Generating IP Cores

You can customize IP cores to support a wide variety of applications. The Quartus II IP Catalog displays IP cores available for the current target device. The parameter editor guides you to set parameter values for optional ports, features, and output files.

To customize and generate a custom IP core variation, follow these steps:

1. In the IP Catalog (Tools > IP Catalog), locate and double-click the name of the IP core to customize. The parameter editor appears.

2. Specify a top-level name for your custom IP variation. This name identifies the IP core variation files in your project. If prompted, also specify the target Altera device family and output file HDL preference. Click OK.

3. Specify the desired parameters, output, and options for your IP core variation:
   - Optionally select preset parameter values. Presets specify all initial parameter values for specific applications (where provided).
   - Specify parameters defining the IP core functionality, port configuration, and device-specific features.
   - Specify options for generation of a timing netlist, simulation model, testbench, or example design (where applicable).
   - Specify options for processing the IP core files in other EDA tools.
4. Click **Finish** or **Generate** to generate synthesis and other optional files matching your IP variation specifications. The parameter editor generates the top-level .qip or .qsys IP variation file and HDL files for synthesis and simulation. Some IP cores also simultaneously generate a testbench or example design for hardware testing.

When you generate the IP variation with a Quartus II project open, the parameter editor automatically adds the IP variation to the project. Alternatively, click **Project > Add/Remove Files in Project** to manually add a top-level .qip or .qsys IP variation file to a Quartus II project. To fully integrate the IP into the design, make appropriate pin assignments to connect ports. You can define a virtual pin to avoid making specific pin assignments to top-level signals.

For more information about IP cores, refer to **Integrating IP Cores** in **Managing Quartus II Projects**.

The **Altera Advanced SEU Detection** IP core must be used along with the **Altera Error Message Register Unloader** IP core. The **Altera Error Message Register Unloader** IP core provides Error Message Register (EMR) contents whenever it detects an EDCRC error. Connect the EMR, EMR_valid and EMR_error signals from your **Altera Error Message Register Unloader** IP variation to the corresponding inputs of your **Altera Advanced SEU Detection** IP variation.

### Files Generated for Altera IP Cores

The Quartus II software generates the following files during generation of your IP core variation.

#### Figure 3–2. IP Core Generated Files

```
<Project Directory>
  <your_ip>.qip - Quartus II IP integration file
  <your_ip>.v or .vhd - Top-level IP synthesis file
  <your_ip>_inst.v or .vhd - Sample instantiation template
  <your_ip>.bsf - Block symbol schematic file
  <your_ip>.vo or .vho - IP functional simulation model
  <your_ip>_syn.v or .vhd - Timing & resource estimation netlist
  <your_ip>_bb.v - Verilog HDL black box EDA synthesis file
  <your_ip>.qip - Quartus II IP integration file
  greybox_tmp
```

Notes:
1. If supported and enabled for your IP variation
2. If functional simulation models are generated
3. Ignore this directory
4. Parameter Settings

## Altera Advanced SEU Detection IP Core Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CRC error cache depth</strong></td>
<td>Specifies how many non-critical cyclic redundancy check (CRC) error to ignore. Default value is 8.</td>
</tr>
<tr>
<td>2, 4, 8, 16, 32, 64</td>
<td></td>
</tr>
<tr>
<td><strong>Largest ASD region ID</strong></td>
<td>Indicates the largest ASD SEU detection region ID in your design. Configures the width of the regions_report port. Default value is 1.</td>
</tr>
<tr>
<td>1 to 255</td>
<td></td>
</tr>
</tbody>
</table>

### Sensitivity Data Access

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use on-chip sensitivity processing</td>
<td>Configures the IP core to use on-chip sensitivity processing or off-chip sensitivity processing. When enabled, implements an external memory interface in the IP.</td>
</tr>
<tr>
<td>ON, OFF</td>
<td></td>
</tr>
<tr>
<td>Memory interface address width</td>
<td>Specifies width of the address bus connected to the external memory interface. Default value is 32.</td>
</tr>
<tr>
<td>___</td>
<td></td>
</tr>
<tr>
<td>Sensitivity data start address</td>
<td>Specifies the offset added to all addresses the external memory interface generates. Default value is 0x0.</td>
</tr>
<tr>
<td>___</td>
<td></td>
</tr>
</tbody>
</table>
Critical applications require an SEU recovery strategy. The Quartus II software provides SEU detection, and allows you to design a recovery response to reduce SEU disruption.

### Enabling the Advanced SEU Detection Feature in the Quartus II Software

To enable the Advanced SEU Detection feature in the Quartus II software, and generate a .smh, turn on **Generate SEU sensitivity map file (.smh)** in the **Device and Pin Options** dialog box (Assignments > Device > Device and Pin Options).

**Figure 5–1. Generate SEU Sensitivity Map File (.smh)**
Hierarchy Tagging

The Quartus II hierarchy tagging feature enables customized soft error classification by indicating design logic susceptible to soft errors. Hierarchy tagging improves design-effective FIT rate by tagging only the critical logic for device operation. You also define the system recovery procedure based on knowledge of logic impaired by SEU. This technique reduces downtime for the FPGA and the system in which the FPGA resides. Hierarchy tagging is available only for Arria V, Cyclone V, Stratix V, and later device families.

The .smh contains a mask for design sensitive bits in a compressed format. The sensitivity mask is generated for the entire design. Hierarchy tagging provides the following benefits:

- Reduces design soft error rate even further by specifying only the critical logic in design.
- Increases system stability by avoiding disruptive recovery procedures for inconsequential errors.
- Allows diverse corrective action for different design logic.

Using Partitions to Specify Logic Sensitivity ID

In the Quartus II software, you can designate a design block as a design partition. You can then assign a sensitivity value to the partition.

The PARTITION_ASD_REGION_ID global assignment specifies the numeric value from 0 to 255. The value represents the sensitivity tag associated with this partition:

```
set_global_assignment -name PARTITION_ASD_REGION_ID <asd_id> -section_id <partition_name>
```

A sensitivity tag of 1 is the same as no PARTITION_ASD_REGION_ID assignment, specifying basic sensitivity level: "region used in design". If a soft error occurs in this partition, the error is reported back as a critical error in the sensitivity region 1.

A sensitivity tag of 0 is reserved, as indication that CRAM bits are not used in your design. You can explicitly set it to indicate that partition is not-critical, and force the partition to be completely excluded from the sensitivity mapping.

You can create multiple partitions with the same sensitivity tag in a design.
Design Partitions Properties

Specify the sensitivity ID assigned to the partition in the ASD Region column in the Design Partition window.

Figure 5–2. ASD Region Column in the Design Partition Window

Sensitivity Map Header File Lookup

The .smh contains critical bit information about the design. The sensitivity data is generated as a standard Intel hex (big-endian) .smh file during .sof generation.

Programming a Sensitivity Map Header File into a Memory

You can program a .smh into any type of memory. For example, to use CFI flash memory, follow these steps:

1. Rename the .smh to <file_name>.hex, or convert it to little-endian <file_name>.hex if required.

2. In the Quartus II software, click File > Convert Programming Files.
3. Under **Output programming file**, select the desired options.

**Figure 5–3. Convert Programming File Dialog Box**

4. To add hex data, follow these steps:
   a. Click **Add Hex Data**.
   b. In the **Add Hex Data** dialog box, turn on **Set start address** and enter a start address.
   c. In the **Hex file** box, click browse to select the `.hex` file, and click **OK**.
Chapter 5: SEU Mitigation on CRAM Array

5–5

Sensitivity Map Header File Lookup

5. Click **Generate**.

   **Figure 5–4. Add Hex Data Dialog Box**

   ![Add Hex Data Dialog Box](image)

---

**Performing a Lookup for SMH Revision 1**

To perform a lookup into the sensitivity map header data using a bit, byte, and frame number from an EMR (Arria V, Cyclone V, Stratix V, and later devices only):

1. Read the 32-bit frame information string for the frame number:
   - Address = 0x14 + (frame*4)
   - Return value = (frame_info_data_offset, offset_map_array_index)

2. Read the offset map information for a frame. The return value for the offset map information is 16 bits:
   - Address = offset_map_base_address + offset array for current frame + offset data value for current byte and bit
     Where,
     - Offset array for current frame = offset_map_array_index * offset_map_length
     - Offset data value for current byte and bit = [(byte * 8) + bit] * 2
     - Return value = offset_map_value

3. Read the 8-bit sensitivity value:
   - Address = (offset_map_value/8) + sensitivity_base_address + frame_info_data_offset
   - Return value = sensitive_bit_word[7:0]

4. Read the sensitive bit. The offset map value provides the sensitive bit index. A value of 1 indicates a critical bit, and a value of 0 indicates a non-critical bit.
   - Sensitive bit = sensitive_bit_word[bit_index]
     Where,
     - bit_index = offset_map_value[2:0]
Performing a Lookup for SMH Revision 2

To perform a lookup into the sensitivity map header data using a bit, byte and frame number from an EMR (Stratix IV devices only):

1. Read the 32-bit frame information string for the frame number:
   - Address = 0x1C + (frame*4)
   - Return value = (frame_info_data_offset, offset_map_array_index)

2. Read the offset map information for a frame. The return value for the offset map information is 16-bits.
   - Address = offset_map_base_address + offset array for current frame + offset data value for current byte and bit
   
   Where,
   - Offset array for current frame = offset_map_array_index*single_offset_map_length
   - Offset data value for current byte and bit = [(byte * 8) + bit] * 2
   - Return value = offset_map_value

3. Read the 8-bit sensitivity value
   - Address = (offset_map_value * sensitivity_data_tag_size / 8) + sensitivity_base_address + frame_info_data_offset
   - Return value = sensitive_data_word[7:0]

4. Read sensitivity data tag. The offset map value provides the sensitive bit index. The return value for the sensitivity tag is sensitivity_data_tag_size bit length. A zero tag indicates that bit is not critical for any region while a non-zero tag indicates offset in region map.
   - sensitive_tag = (sensitive_data_word >> tag_shift) and tag_mask
   
   Where,
   - tag_shift = (offset_map_value * sensitivity_data_tag_size)[2:0]
   - tag_mask = (0x1 << sensitivity_data_tag_size) - 1;

5. Read the region mask for a non-zero sensitivity tag. The return value for the region mask is 16 bits.
   - region_mask = region_map_base_address + (sensitivity_data_tag - 1) * 2
This document provides additional information about the document and Altera.

**Document Revision History**

The following table lists the revision history for this document.

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 30 2014</td>
<td>2014.06.30</td>
<td>■ Updated supported devices. ■ Replaced information about the MegaWizard Plug-in Manager with the IP Catalog.</td>
</tr>
<tr>
<td>December 2012</td>
<td>1.0</td>
<td>Initial release for MOLSON.</td>
</tr>
</tbody>
</table>

**How to Contact Altera**

To locate the most up-to-date information about Altera products, refer to the following table.

<table>
<thead>
<tr>
<th>Contact (1)</th>
<th>Contact Method</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical support</td>
<td>Website</td>
<td><a href="http://www.altera.com/support">www.altera.com/support</a></td>
</tr>
<tr>
<td>Technical training</td>
<td>Website</td>
<td><a href="http://www.altera.com/training">www.altera.com/training</a></td>
</tr>
<tr>
<td>Email</td>
<td></td>
<td><a href="mailto:custrain@altera.com">custrain@altera.com</a></td>
</tr>
<tr>
<td>Product literature</td>
<td>Website</td>
<td><a href="http://www.altera.com/literature">www.altera.com/literature</a></td>
</tr>
<tr>
<td>(software licensing)</td>
<td>Email</td>
<td><a href="mailto:nacomp@altera.com">nacomp@altera.com</a></td>
</tr>
<tr>
<td>(nontechnical support (general))</td>
<td>Email</td>
<td><a href="mailto:authorization@altera.com">authorization@altera.com</a></td>
</tr>
</tbody>
</table>

*Note to Table:*

(1) You can also contact your local Altera sales office or sales representative.

**Typographic Conventions**

The following table shows the typographic conventions this document uses.

<table>
<thead>
<tr>
<th>Visual Cue</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bold Type with Initial Capital Letters</strong></td>
<td>Indicate command names, dialog box titles, dialog box options, and other GUI labels. For example, Save As dialog box. For GUI elements, capitalization matches the GUI.</td>
</tr>
<tr>
<td><strong>bold type</strong></td>
<td>Indicates directory names, project names, disk drive names, file names, file name extensions, software utility names, and GUI labels. For example, \qdesigns directory, D: drive, and chiptrip.pdf file.</td>
</tr>
<tr>
<td><strong>Italic Type with Initial Capital Letters</strong></td>
<td>Indicate document titles. For example, Stratix IV Design Guidelines.</td>
</tr>
<tr>
<td><strong>italic type</strong></td>
<td>Indicates variables. For example, ( n + 1 ). Variable names are enclosed in angle brackets (‹ ›). For example, &lt;file name&gt; and &lt;project name&gt;.pof file.</td>
</tr>
<tr>
<td>Visual Cue</td>
<td>Meaning</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Initial Capital Letters</td>
<td>Indicate keyboard keys and menu names. For example, the Delete key and the Options menu.</td>
</tr>
<tr>
<td>“Subheading Title”</td>
<td>Quotation marks indicate references to sections in a document and titles of Quartus II Help topics. For example, “Typographic Conventions.”</td>
</tr>
<tr>
<td>Courier type</td>
<td>Indicates signal, port, register, bit, block, and primitive names. For example, data1, tdi, and input. The suffix n denotes an active-low signal. For example, resetn. Indicates command line commands and anything that must be typed exactly as it appears. For example, c:\qdesigns\tutorial\chiptrip.gif. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword SUBDESIGN), and logic function names (for example, TRI).</td>
</tr>
<tr>
<td>➤</td>
<td>An angled arrow instructs you to press the Enter key.</td>
</tr>
<tr>
<td>1., 2., 3., and a., b., c., and so on</td>
<td>Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.</td>
</tr>
<tr>
<td>■ ■ ■</td>
<td>Bullets indicate a list of items when the sequence of the items is not important.</td>
</tr>
<tr>
<td>🖐</td>
<td>The hand points to information that requires special attention.</td>
</tr>
<tr>
<td>🤔</td>
<td>The question mark directs you to a software help system with related information.</td>
</tr>
<tr>
<td>🔗</td>
<td>The feet direct you to another document or website with related information.</td>
</tr>
<tr>
<td>🎥</td>
<td>The multimedia icon directs you to a related multimedia presentation.</td>
</tr>
<tr>
<td>☢</td>
<td>A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.</td>
</tr>
<tr>
<td>⚠</td>
<td>A warning calls attention to a condition or possible situation that can cause you injury.</td>
</tr>
<tr>
<td>💌</td>
<td>The envelope links to the Email Subscription Management Center page of the Altera website, where you can sign up to receive update notifications for Altera documents.</td>
</tr>
</tbody>
</table>