

SP601 Post Configuration CRC

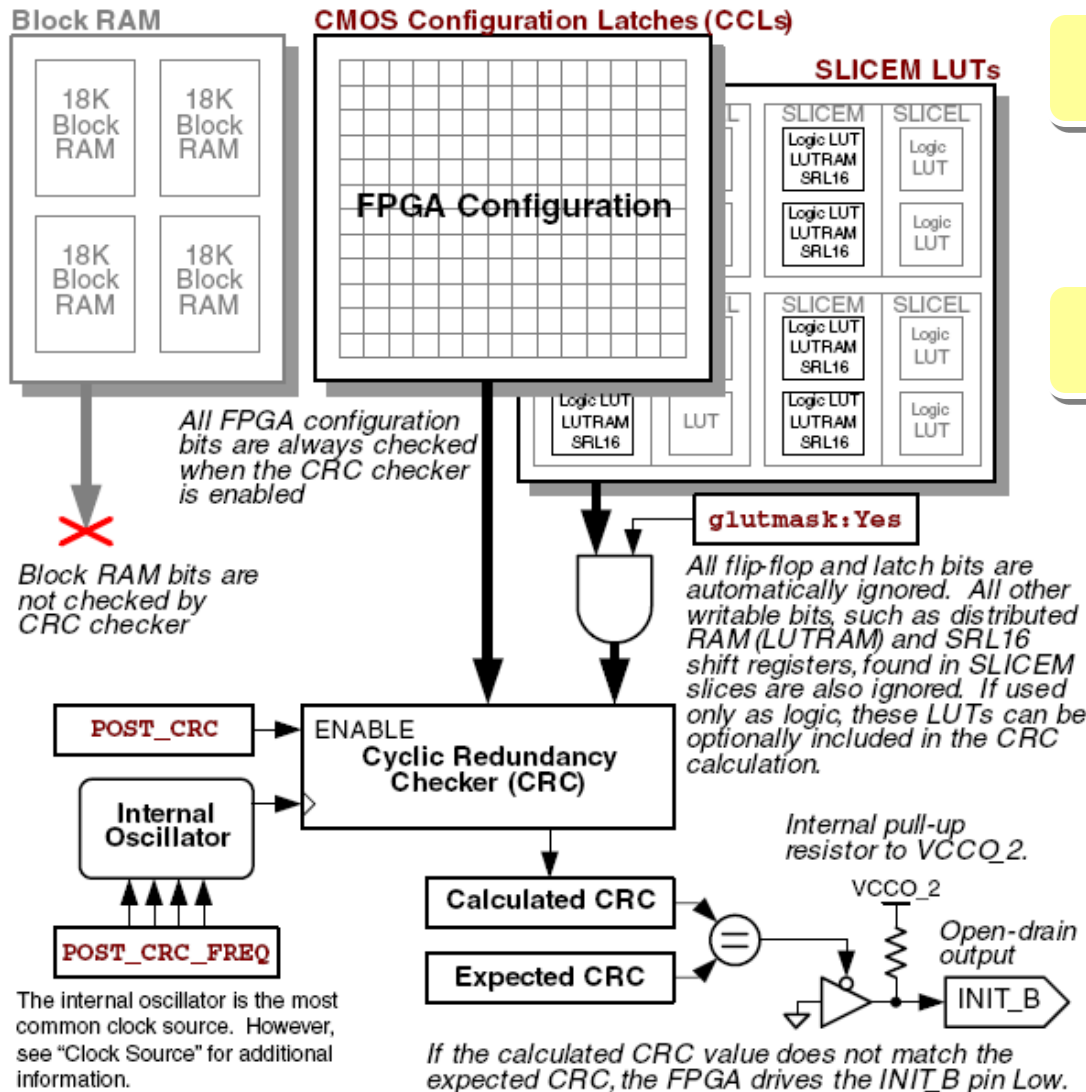
Overview

- **Introduction**
- **Hardware Implementation**
- **Constraints**
- **Reference Design**
- **References**

Introduction

- **Allows for continuous readback of configuration data**
- **Simple SEU (Single Event Upset) detection**
- **Enabled through CONFIG constraints or Bitgen settings**
- **Error detected on INIT_B pin and/or design primitive:
(POST_CRC_INTERNAL)**

Hardware Implementation



Readback CRC checking is enabled



CRC checker circuit continuously scans bitstream

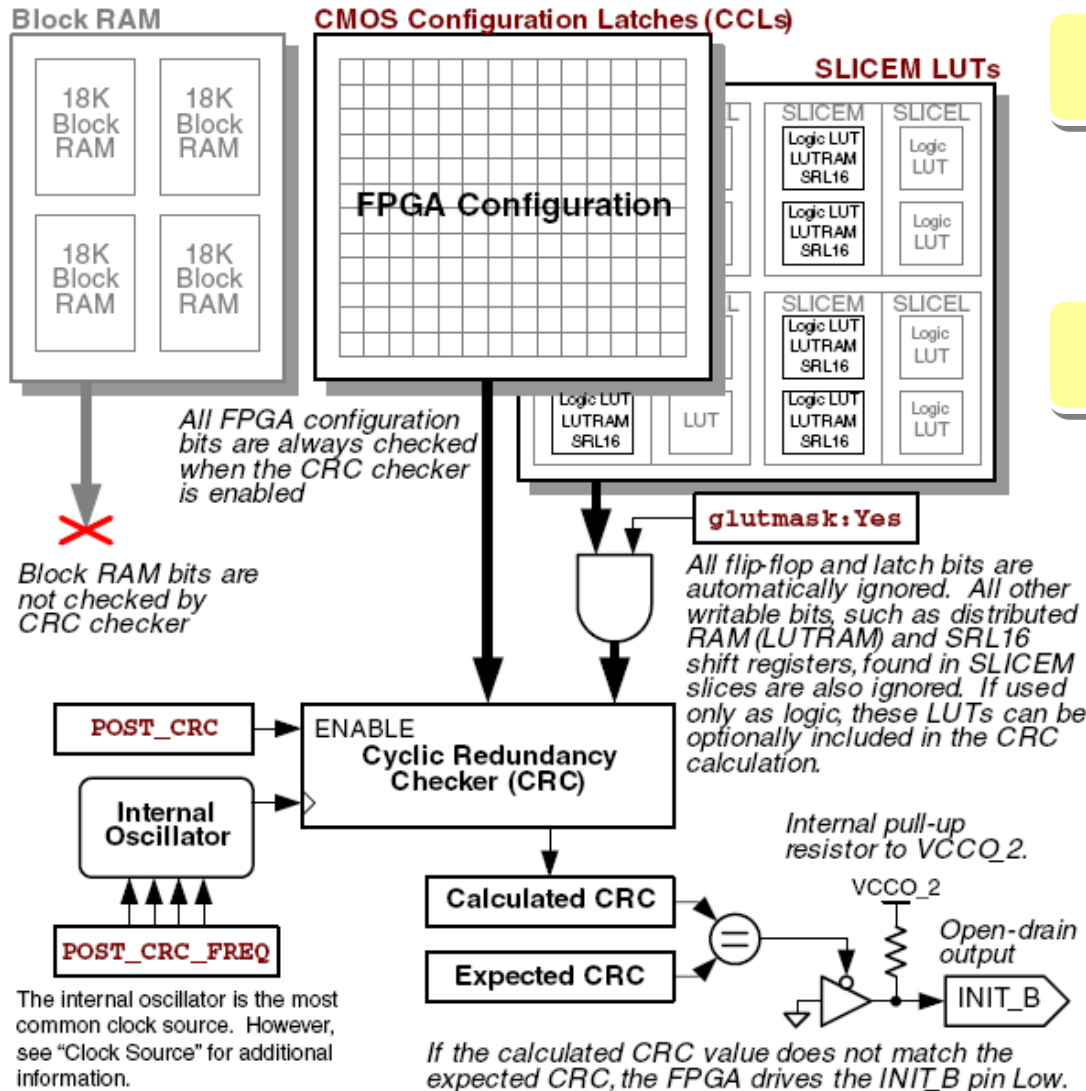


Calculates CRC value



Compares value against previously calculated, expected (golden) CRC value




Hardware Implementation



Difference in two CRC check values



CRC checker flags the error

-  CRCERROR pin of POST_CRC_INTERNAL primitive is driven High
-  INIT_B driven low
-  DONE remains high

NOTE: POST_CRC_INIT_FLAG can be set to DISABLE to disable INIT_B as a Readback CRC flag

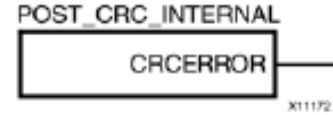
Hardware Implementation

- **Dynamic content will be automatically excluded from CRC checking.**
 - Component types masked during CRC checking:
 - LUTRAM, SRL16, and Block RAM
 - Masking manually controlled with bitgen –g glutmask option
- **Component types checked during CRC checking:**
 - PLL DRP
 - Any change to PLL results in a CRC error PLL DRP
 - I/O interface DRP at top and bottom
 - Any changes to these interfaces result in a CRC error
 - GTP DRPs
 - Any changes here will result in a CRC error

Hardware Implementation

- **POST_CRC_INTERNAL primitive for user access to Error Flag**

- Provides fabric access to post configuration CRC error
- If POST_CRC_INIT_FLAG is set to Disable INIT will not flag errors
 - See Spartan-6 Libraries Guide for more information



Port	Type	Width	Function
CRCERROR	Output	1	Post-configuration CRC error

VHDL Instantiation Template

```
-- POST_CRC_INTERNAL: Post-configuration CRC error detection
--                      Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

POST_CRC_INTERNAL_inst : POST_CRC_INTERNAL
generic map (
)
port map (
  CRCERROR => CRCERROR -- 1-bit Post-configuration CRC error
);

-- End of POST_CRC_INTERNAL_inst instantiation
```

Verilog Instantiation Template

```
// POST_CRC_INTERNAL: Post-configuration CRC error detection
//                      Spartan-6
// Xilinx HDL Language Template, version 11.1

POST_CRC_INTERNAL POST_CRC_INTERNAL_inst (
  .CRCERROR(CRCERROR) // 1-bit Post-configuration CRC error
);

// End of POST_CRC_INTERNAL_inst instantiation
```

Constraints

▪ POST_CRC

- Enable of Post Configuration CRC
- Settings:
 - ENABLE = SEU detection is enabled
 - DISABLE = SEU detection is disabled
- Constraints:

POST_CRC

NCF Syntax Example

```
CONFIG POST_CRC = [ENABLE|DISABLE]
```

UCF Syntax Example

```
CONFIG POST_CRC = [ENABLE|DISABLE]
```

- Equivalent BitGen Options:
 - -g post_crc_en:yes
 - -g glutmask:yes

Constraints

▪ POST_CRC_INIT_FLAG

- Determines whether INIT_B pin is used as SEU error signal
- Settings:
 - ENABLE = INIT_B pin is used as indicator (default)
 - DISABLE = INIT_B **NOT** used as an indicator
- If INIT_B is not set as error status pin, POST_CRC_INTERNAL can be used to note the error condition
- Constraints:

POST_CRC_INIT_FLAG

NCF Syntax Example

```
CONFIG POST_CRC_INIT_FLAG = [ENABLE|DISABLE]
```

UCF Syntax Example

```
CONFIG POST_CRC_INIT_FLAG = [ENABLE|DISABLE]
```

- No equivalent BitGen option

Constraints

▪ **POST_CRC_ACTION**

- Determines action of CRC engine after CRC error is detected
- Settings:
 - HALT – If a CRC error is detected, no further Readback CRC checking is performed. After error is cleared, CRC testing resumes (default)
 - CONTINUE – If a CRC error is detected, the error flag is issued and checking continues to be performed
- Constraints:

POST_CRC_ACTION

NCF Syntax Example

```
CONFIG POST_CRC_ACTION = [HALT | CONTINUE]
```

UCF Syntax Example

```
CONFIG POST_CRC_ACTION = [HALT | CONTINUE]
```

- Equivalent BitGen Option
 - -g post_crc_keep:no

Constraints

▪ **POST_CRC_FREQ**

- Determines the frequency of the internally generated clock to Readback CRC logic in MHz
- Settings:
 - 2, 4, 6, 10, 12, 16, 22, 26, 33, 40, 50, and 66.
- Constraints:

POST_CRC_FREQ

NCF Syntax Example

```
CONFIG POST_CRC_FREQ = [2|4|6|10|12|16|22|26|33|40|50|66]
```

UCF Syntax Example

```
CONFIG POST_CRC_FREQ = [2|4|6|10|12|16|22|26|33|40|50|66]
```

- Equivelent BitGen Option

- -g post_crc_freq:2

Constraints

▪ BitGen options will also enable Post Configuration CRC

BitGen Option	Setting (<i>default</i>)	Description
post_crc_en	<u>No</u>	Default. Disable the post-configuration CRC checker.
	Yes	Enable the post-configuration CRC checker.
post_crc_freq	2, 4, 6, 10, 12, 16, 22, 26, 33, 40, 50, 66	Sets the clock frequency used for the post-configuration CRC checker.
post_crc_keep	<u>No</u>	Default. Stop checking when error detected. Allows CRC signature to be read back.
	Yes	Continue to check for CRC errors after an error was detected.
glutmask	<u>Yes</u>	Default. Mask out the Look-Up Table (LUT) bits from the SLICEM logic slices. SLICEMs support writable functions such as distributed RAM and SRL16 shift registers, which generate CRC errors when bit locations are modified.
	No	Include the Look-Up Table (LUT) bits from SLICEM logic slices. Use this option only if the application does not include any distributed RAM or SRL16 shift registers.

▪ Preferred to use CONFIG Constraints

- Only way to disable INIT flag is with constraints

Reference Design

- **Software:** 11.2 (L.46)
- **Design:**
 - 2 SRL16E components connected with a counter
 - -g glutmask:no is set to “trigger” the Readback CRC error
- **Display:**
 - Phase Width Modulation (PWM) type setup is used to vary LED intensity for 4 LEDs on the SP601 board
- **Activity:**
 - SRL16 Clock Enable and Reset connected to two switches.
 - Shifting data through SRL16 will trigger error with glutmask set

Reference Design

▪ UCF Setup

- CRC Enabled with the UCF constraint:
 - CONFIG POST_CRC = ENABLE
- Set the oscillator frequency using:
 - CONFIG POST_CRC_FREQ = 2
- Set up for continuous or halted checking on an error:
 - CONFIG POST_CRC_ACTION = HAULT

▪ Triggering CRC Errors

- Dynamic elements (SRL16) are used without masking function
- BitGen -g glutmask:no will override default setting
 - Dynamic elements will be included in CRC check and if they change CRC will fail.
- CRC errors can be manually triggered when the SRL16 changes state.

Reference Design

- When the `-g glutmask BitGen` option is set to No, the dynamic elements of the design are not masked out -> a CRC error is seen
- When a `POST_CRC` error is detected, and the `CONFIG POST_CRC_INIT_FLAG` constraint is enabled, `INIT_B` will go low
- The red `INIT_B` LED will turn ON on the SP601
- This is a status LED on the SP601 – during normal post-configuration operation, this `INIT_B` LED will be turned off, meaning there is no CRC error

Reference Design

- The reference design can be loaded into the Spartan-6 LX16 on the SP601 using iMPACT
- With `-g glutmask:yes` and `-g post_crc_en:yes` design runs through implementation, bit file is generated, downloaded to FPGA, design starts up as expected
- Initialise the chain, assign `test_sp6.bit`, and configure the FPGA
- Observe the board behaviour
- Note `DONE` is high, `INIT` is high (`INIT LED` is not lit on the board)

Reference Design

- **Set `-g glutmask: No`**

- Note INIT goes low when design is downloaded to board
- This is because dynamic SRL16 values are not masked
- A Readback CRC error is triggered

- **With `-g glutmask: No` still set**

- Add the `CONFIG POST_CRC_ACTION = CONTINUE` constraint to the UCF file
- This means that when the initial CRC error is detected, the CRC circuit continues to check for additional Readback CRC errors
- `INIT_B` pin stays low after the initial error

Options to change design behaviour

- **With `-g glutmask: No` still set**

- Add the `CONFIG POST_CRC_ACTION = HALT` constraint to the design UCF file
- This means that the CRC circuit stops calculating a new CRC value, and stops checking for additional CRC errors, if an initial Readback CRC error occurs

- **Remove `-g post_crc_en: Yes` from BitGen options**

- Replace this with the `CONFIG POST_CRC=ENABLE` configuration constraint in the design UCF file
- Re-run design and note this gives the same behaviour as setting the `-g post_crc_en: Yes` BitGen option

References

- **UG380**

- Spartan-6 FPGA Configuration User Guide

- http://www.xilinx.com/support/documentation/user_guides/ug380.pdf

- **ISE 11 Software Manuals**

- Xilinx Command Line Tools User Guide

- http://www.xilinx.com/support/documentation/sw_manufact/xilinx11/devref.pdf