pok\_gen

Integration Guide

IP Rev. 0.3

Intel Restricted Secret

Copyright © 2014, Intel Corporation. All rights reserved.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

\* Other names and brands may be claimed as the property of others.

This document contains information on products in the design phase of development.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED OR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked “reserved” or “undefined.” Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your Intel account manager or distributor to obtain the latest specifications and before placing your product order.

Copies of documents that have an order number and are referenced in this document or in other Intel literature can be obtained from your Intel account manager or distributor.

Contents

[1 Introduction 4](#_Toc500774102)

[1.1 Audience 4](#_Toc500774103)

[1.2 Supported Projects 4](#_Toc500774104)

[1.3 Terminology 4](#_Toc500774105)

[1.4 Related Documents 4](#_Toc500774106)

[1.5 Opens, Risks, and Assumptions 4](#_Toc500774107)

[1.6 Contact Information 4](#_Toc500774108)

[1.7 Document Revision History 4](#_Toc500774109)

[2 Overview 5](#_Toc500774110)

[2.1 Goal 5](#_Toc500774111)

[2.2 I/O 5](#_Toc500774112)

[2.3 Parameters 6](#_Toc500774113)

[2.4 State Machine 6](#_Toc500774114)

[2.5 Timing diagrams 7](#_Toc500774115)

[2.6 Integration details 8](#_Toc500774116)

[2.7 IOSF requirements for the deassertion of side\_pok 8](#_Toc500774117)

[3 SS impact of implementing this module in an IP 10](#_Toc500774118)

[4 User Registry 12](#_Toc500774119)

# Introduction

## Audience

The information in this document is intended for an integration or design team that is using this reference design.

## Supported Projects

## Terminology

## Related Documents

If you need more information on this IP, you may find these documents helpful.

|  |  |
| --- | --- |
| Document Title | Location |
| IOSF spec 1.2 | <https://sharepoint.amr.ith.intel.com/sites/MDGArchMain/Converged/SIGA/IOSF%20Specs/Spec%20Release/1.2%20Spec%20Release/IOSF%20spec%201.2.pdf> |
| Widget block diagrams.vsdx | In widget repo doc/widget block diagrams.vsdx |
|  |  |

## Opens, Risks, and Assumptions

|  |  |  |  |
| --- | --- | --- | --- |
| Item # | Description | Comment | Status (Open or Closed)/Date |
|  |  |  |  |

## Contact Information

If you need additional help, use the contact information below.

|  |  |  |
| --- | --- | --- |
| Function | Name | Email |
| IP Integration | Ken Correll | [ken.correll@intel.com](mailto:ken.correll@intel.com) |
| Doc Template Owner | Susann Flowers | [susann.flowers@intel.com](mailto:susann.flowers@intel.com) |

## Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision Number | Description of Change | Date | Revised By |
| 0.1 | Initial Release |  | Ken Correll |
| 0.2 | Modified equation generating fpgpok\_req\_held to only look at bit[0] of the Type field, since that bit indicates whether pok should be deasserted or not | 5/17/18 | Ken Correll |
| 0.3 | Corrected typo in section 3 where I incorrectly had ADL, not SBBRIDGE | 1/7/19 | Ken Correll |

# Overview

## Goal

The pok\_gen.sv file was created as a reference design for adoption by IP’s that include an IOSF\_SB interface and do not currently support generation of side\_pok. It is hoped that many such chassis IP’s adopt this reference design

## I/O

|  |  |  |
| --- | --- | --- |
| Signal Name | Direction | Discription |
| side\_clk | input | Side\_clk that is connected to the local iosf sideband endpoint |
| side\_rst\_b | input | Side\_rst\_b that is connected to the local iosf sideband endpoint with it’s deasserting edge synchronized to side\_clk |
| ism\_agent[2:0] | input | Side\_ism\_agent produced by the local iosf sideband endpoint  This must come directly from the endpoint with a single cycle path from ism\_agent to ism\_lock\_b back to endpoint |
| wake | Input | Optional input, from the local pma, must be sync’ed to side\_clk outside this module. If not used tie to ‘0. If used, the parameter ACCESSIBLE\_ON\_WAKE must be set to ‘1.  It is recommended to not use this input. |
| fpgpok\_req[1:0] | input | The user must decode the ForcePwrGatePOK message and output the type field to these inputs. These should be reset to 2’b00 with side\_rst\_b. They must be held by the decoding logic for a minimum of 1 side\_clk cycle.  If bit 0 is set, then POK de-assertion is required.  If bit 1 is set, then PGReq assertion is required.  This circuit only acts on type field 2’b01 |
| sbe\_idle | Input | Connect to the sideband endpoints output of the same name. |
| fpgpok\_holdoff | input | To be connected to the IP’s NP queue status so that the endpoint remains in the Accessible state until all NP completions have been received.  Also can be used (with external OR’ing with above) as an optional input that will delay sequencing to ism\_lock assertion until local IP conditions are meet outside the status of the sideband endpoint. |
| ism\_lock\_b | output  (not flopped) | Use as the ism\_lock\_b input to the local iosf sideband ep  This must go directly to the endpoint with a single cycle path from ism\_agent to ism\_lock\_b back to endpoint |
| pok | output  (flopped) | Use as the pok output, part of the iosf sb standard interface, to be connected to the router and to the local PMA on ip\_pm\_pok – a vectored input (non-pmrc related). |

## Parameters

|  |  |  |
| --- | --- | --- |
| PARAMETER NAME | DEFAULT VALUE | DESCRIPTION |
| ACCESSIBLE\_ON\_WAKE | 0 | Determines whether a pm\_ip\_wake signal must be used to to transition from INACCESSIBLE state |

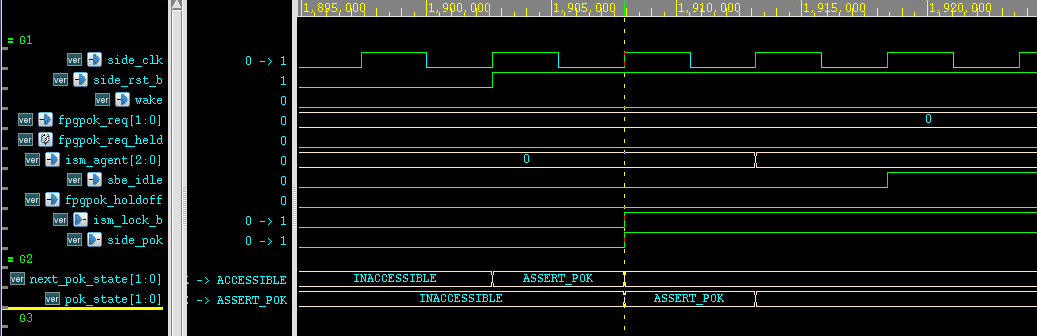
## State Machine



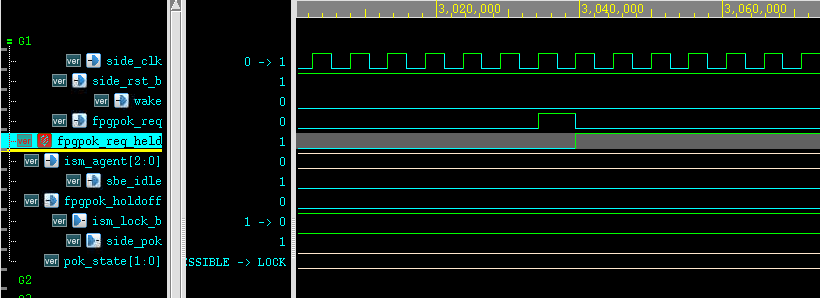
|  |  |  |  |
| --- | --- | --- | --- |
| pok\_state | next\_pok\_state | Condition to move to next state | Outputs |
| INACCESSIBLE | ASSERT\_POK | Deassertion of side\_rst\_b or  Assertion of wake if side\_rst\_b is asserted | ism\_lock\_b = 0  pok = 0 |
| ASSERT\_POK | ASSESSIBLE | automatic | ism\_lock\_b = 0  pok = 1 |
| ASSESSIBLE | LOCK | fpgpok\_req &&  ism\_agent[2:0] == IDLE &&  !fpgpok\_holdoff &&  sbe\_idle | ism\_lock\_b = 1  pok = 1 |
| LOCK | INACCESSIBLE | !side\_rst\_b | ism\_lock\_b = 0  pok = (pok\_state == ACCESSIBLE) |
| ASSERT\_POK  Or  ACCESSIBLE | INACCESSIBLE | !side\_rst\_b | Ism\_lock\_b = 0  Pok = 0  Both transition one clock cycle after side\_rst\_b asserts |

## Timing diagrams

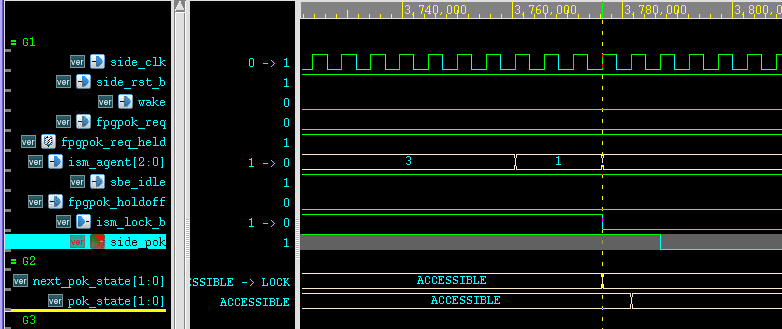
Coming out of reset with no wake:



Receipt of ForcePwrGatePOK message:



Transition to LOCK:



Transition to INACCESSIBLE:

## Integration details

Copy the Verilog file from /nfs/sc/disks/sdg74\_0279/kcorrell/pok\_gen/pok\_gen.sv

Edit the Verilog filename and module name to uniquify it for your application.

The pok\_gen\_Integration\_Guide.docx is located at:

Checkout/edit/checkin: Register as a user in the pok\_gen registery at the end of this document so that you can be notified of any findings.

Note that for the IOSF\_SB\_SBRIDGE, the ForcePwrGatePOK message is already decoded with the type field exported as outputs. This message is not addressed to the SBBridge, it is forwarded to endpoints on the local gpsb bus.

If the SBBridge incorporates this design, it should only forward the type field to the pok\_gen if the ForcePwrGatePOK message is addressed to the SBBridge.

## IOSF requirements for the deassertion of side\_pok

Requirements in the IOSF spec for deasserting pok are:

Rule 5. An agent may only deassert pok when:  
a) The agent ISM is in the idle state.  
b) The agent is not waiting for a completion to an outstanding NP request.

c) The agent has returned all credits.[RA490]

Items a & c are met by monitoring the agent\_ism bits directly and by connecting sbe\_idle.

Item b) refers to incoming and outgoing NP requests.

Incoming NP requests are satisfied by setting the endpoints ISM\_COMPLETION\_FENCING and EXPECTED\_COMPLETIONS\_COUNTER parameters both to ‘1. This keeps the ism\_agent state machine from transitioning to the IDLE state while a NP request is outstanding.

Outgoing NP requests are satisfied by the integrator connecting the local IP’s NP queue status to the fpgpok\_holdoff input pin. The pin is active high meaning that if the NP queue has an empty status it should be inverted before connecting.

# SS impact of implementing this module in an IP

Many chassis IP’s did not implement the logic required to generate side\_pok. For wave1 and wave2, a module created by the RCF group was used as a helper block, connecting in-between the gpsb endpoint and the global router – the rcfwl\_cdc\_wrapper.

The generation of side\_pok is basically the same as implemented in this logic block, but the ClockDomainController inside the rcfwl\_cdc\_wrapper was designed to work with the PGCB, and have clock control.

To make this change, the SS should –

* Connect the clkreq/clkack from the endpoint to the clkreqaggr that the cdc\_wrapper connects to
* Connect the pok output to the PMA input pin that the cdc\_wrapper’s pok is connected to
* Connect the pok output as part of the standard iosf\_sb collage interface to the router
* Remove the cdc\_wrapper and all connections associated with it.

In chassis, if the iosf\_sb\_bridge were to implement this logic, these are the steps that would need to be taken, using scf\_mem as an example.

In tools/collage/common/subsystems/scf\_mem\_chassis/create.tcl:

Remove lappend ip\_instance\_list rcfwl\_cdc\_wrapper vcfcx12\_cdc\_wrapper

In tools/collage/common/subsystems/scf\_mem\_chassis/init.tcl:

Remove reference to rcfwl\_cdc\_wrapper

In tools/collage/common/subsystems/scf\_mem\_chassis/adhoc\_connection.txt:

1. Disconnect the clkreq/clkack from the cdc\_wrapper to the clkreqaggr and instead connect the clkreq/clkack from the endpoint directly to the clkreqaggr:

From:

C gpsb\_mem\_sbbridge{%hier\_str}/gbl\_agent\_side\_clkreq

vcfcx12\_cdc\_wrapper{%hier\_str}/gclock\_req\_async

C vcfcx12\_cdc\_wrapper{%hier\_str}/clkreq side\_vcfc\_vdom\_clkreqaggr{%hier\_str}/iclkreq[1]

C side\_vcfc\_vdom\_clkreqaggr{%hier\_str}/oclkack[1] vcfcx12\_cdc\_wrapper{%hier\_str}/clkack

C vcfcx12\_cdc\_wrapper{%hier\_str}/gclk\_async\_ack\_synced

gpsb\_mem\_sbbridge{%hier\_str}/gbl\_agent\_side\_clkack

To:

C gpsb\_mem\_sbbridge{%hier\_str}/gbl\_agent\_side\_clkreq

side\_vcfc\_vdom\_clkreqaggr{%hier\_str}/iclkreq[1]

C side\_vcfc\_vdom\_clkreqaggr{%hier\_str}/oclkack[1]

gpsb\_mem\_sbbridge{%hier\_str}/gbl\_agent\_side\_clkack

1. Remove all connections from the vcfcx12\_cdc\_wrapper to the sbbridge and vice versa
2. Remove the connections:

C vcfcx12\_cdc\_wrapper{%hier\_str}/ism\_lock\_b

gpsb\_mem\_sbbridge{%hier\_str}/gbl\_agent\_side\_ism\_lock\_b

gpsb\_mem\_sbbridge{%hier\_str}/gbl\_agent\_side\_ism\_lock\_b is unused, tie low or hi

1. Change side\_pok connect to the PMA:

From:

C vcfcx12\_cdc\_wrapper{%hier\_str}/pok mspmas0{%hier\_str}/ip\_pm\_pok[0]

To:

C gpsb\_mem\_sbbridge{%hier\_str}/gbl\_agent\_side\_pok mspmas0{%hier\_str}/ip\_pm\_pok[0]

1. Remove all other connections to the vcfcx12\_cdc\_wrapper
2. Add side\_pok to the collage standard connection

In tools/collage/common/subsystems/scf\_mem\_chassis/std\_connection.cfg

Remove connections to the sbbridge

In tools/collage/common/subsystems/scf\_mem\_chassis/

Remove tie-offs to the vcfcx12\_cdc\_wrapper

In tools/collage/common/subsystems/scf\_mem\_dft\_chassis/adhoc\_connection.txt

Remove clock connections to the vcfcx12\_cdc\_wrapper

In tools/collage/soc\_mini/integ\_specs/soc\_cdc\_wrapper\_adhoc\_connection.txt,

soc\_dfx\_std\_connection.cfg,

soc\_soc\_par.txt,

soc\_upf\_config.tcl:

Remove all connects to/from vcfcx12\_cdc\_wrapper\_scf\_mem\_c0\_r1

# User Registry

|  |  |
| --- | --- |
| Name | IP pok\_gen is used in |
| Michael Dejesus | All PM blocks |
| Smitha Bmv | IEH |