Power Gating Control Block (PGCB)

Integration Guide

Revision 1.23

Hartej Singh and Yong J. Kim

1.0 Overview of PGCB

Acknowledgements

The Power Gate Control Block design is the result of the efforts of multiple people who spent a significant amount of time defining, reviewing and contributing to the design. Some of the key contributors are mentioned below.

Dhinesh Sasidaran – Dhinesh created the initial micro-architecture of the PGCB. He provided a significant part of the PGCB definition. The current implementation is based on the micro-architectural framework developed by him.

Asad Azam – Asad contributed to the initial definition of the PGCB by participating in brainstorming and review meetings for the PGCB. He provided valuable design feedback from the perspective of a Soft IP.

Bill Knolla – Bill was instrumental in ensuring that Chassis PG (power-gating) requirements were adequately captured as part of the PGCB definition. He also provided technical direction, design ideas, and valuable feedback on design proposals from the perspective of Chip uArchitect.

Jeff Wilcox, Mikal Hunsaker, Doug Moran – These architects have been closely involved in the review of the PGCB design. Jeff in particular has pioneered the idea of power-gating flows based on pre-commit, and the PGCB has been updated to use these flows as the design basis.

SPT State Retention TR Working Group – This WG led the technical readiness efforts for enabling use of state retention cells. This enabled faster definition and release of the PGCB.

Many Other SIP micro-architects – PGCB design has benefited greatly from the contribution of multiple other SIP team micro-architects who participated enthusiastically in multiple review sessions and took the time to perform detailed design evaluations, point out potential issues, and suggest resolutions for those.

Revision History

Version	Date	Author	Description
0.50	WW35.6	jwhavica	Initial Release • Split out PGCB related documentation from the State Retention HAS (authored by Dhinesh Sasidaran) to create PGCB Integration doc. • Based on the original definition of the PGCB and FSM designed by Dhinesh Sasidaran (refer section 3.0 of State_Retention_108900_HAS_rev0p7.doc - HAS related to the SPT Requirement: 108900) • Updated section on PGCB Scope, separated out PMC and IP interfaces to PGCB, added separate waveforms (in lieu of previous consolidated waveform). • Removed section on PGCB Phases (in keeping with focus on IP-centric behavior as documented in waveforms) • Added new interface signals and 3-signal handshakes for initiating power gating and for locking both ISMs.

			• Added new section on IP power gating requirements.
0.51	WW36.6	jwhavica	Minor Spec Tweaks: • Added Revision History • Added Abort Flow Waveform • Updated wording of stability requirement on config signals and ip_pgcb_pg_type to correct requirement.
0.52	WW40.1	jwhavica	Compliance to Chassis PG HAS 0.70 Final • IP-Accessible flow changes ○ ISM Unlock is now done after sleep deassertion ○ Added force_clks_on and all_pg_rst_up signals to avoid overloading clkgate_req on PG exit with state retention • IP-inaccessible flow changes ○ unique exit sequence with staggered ISM unlock handshakes ○ clkgate_req handshake on PG entry always happens ○ sleep will now assert as part of IP-Inaccessible flow to isolate ungated portion of SR cell from gated portion • Added Reset Values / Expected Reset Values of Interface Signals • Added Parameter to specify Powered On/Power Gated default state of PGCB • Relaxed requirement on when pg_type, sleep_en and cfg_t* signals can change • Interface Changes ○ Added: ▪ pgcb_ip_force_clks_on ▪ ip_pgcb_all_pg_rst_up ▪ cfg_tlatchen ▪ cfg_tsiderstup ○ Renamed ip_pgcb_sleep_ovr -> ip_pgcb_sleep_en and changed polarity to match Chassis PG HAS • Changed delays represented by cfg_t* signals • Added Abort Flow Diagrams • Added additional IP requirements/recommendations • Added Section regarding Latch Based Isolation
0.53	WW44.3	jwhavica	• Updated “Latch Based Isolation” section to reflect SPT isolation latch POR • Added two new configurable delay values (cfg_tsaveack and cfg_tlatchdis)

			Modified cfg_tclkgate to enforce delay from clkgate_ack_b assertion
0.53a	WW44.3	jwhavica	Updated the descriptions of cfg_t* inputs to make sure they match the latest implementation.
0.53p1	WW45.6	jwhavica	**Temporary** work-around for VCS modeling issue. • If sleep deasserts while reset is deasserted, VCS does not model the state-retention cell behavior properly and the cell will not be reset, until clocks start toggling. • Temporary workaround is to not assert SLEEP during IP-Inaccessible power-gating flows. This change will need to be reverted before tape-in as SR cells require sleep assertion if PGD power is removed for internal isolation from PGD. • Please note, the waveforms are not updated to reflect this change.
0.60	WW49.5	hsingh1	**Changes to sync up with Chassis PG flows (based on ECN)** • Several significant changes to most sections of this document (scope, interfaces, waveforms, IP power gating requirements, etc.) • PGCB SM still reflects the previous implementation, but this is to be updated in a subsequent release of this document.
0.61	WW51.2	hsingh1	**Changes to support interface requirements for reference design (based on clock domain controller – CDC)** • Support for forcing clocks ON during PG entry/exit flows is disabled by default – a new input is defined to support when this feature is required – refer to IP Power Gating Requirements section for more details (waveforms do not include this feature, but next incremental update will include those signals as part of the waveforms). • Integrated POK signal control function within the PGCB – which now provides an output signal “pgcb_pok” – that is intended to specify the POK signal value (both in PG state and unPG state, as well as during PG entry/exit flows). • Added the “pgcb_restore” signal – this asserts when the restore phase is needed (check signal description for more details). Also added “pgcb_pwrgate_active” signal (combinatorial output) that indicates that the PGCB is

			either processing the PG entry flow, or in stable PG state, or is executing the PG Exit flow. • Removed "*ism_lock_req_b" signals and also "*_clkgate_req_b" signals (those features are expected to be handled within the IP block – such as within the Clock Domain Controller [CDC] component of reference design). Removed all "*_nak" signals from IP-PGCB interaction signals. • Support for Warm Reset by making the "ip_pgcb_pg_type" signal to be 2 bits. Changed the expected encoding for this signal to match data payload of ForcePwrGatePOK message. • Added new count delays corresponding to additional steps of the waveforms.
0.7RC	WW02.4	hsingh1	**Following changes:** • Changes for "force_clks_on" support: ○ Added description for support for forcing clocks on during the PG Exit sequence for IP blocks that have synchronous resets and/or need clocks turned on for context propagation. ○ Added another strap/config input to support configuration of "force_clks_on" for synchronous resets separate from context propagation. ○ Updated waveforms to show behavior of pgcb_ip_force_clks_on (and *_ack) signal when relevant PGCB strap/config inputs are configured appropriately. • De-feature of "pgcb_force_prim_rst_b" and "ip_fuse_valid" signals ○ With the changes in Chassis Reset Arch HAS to require, in general, all IP blocks to send IP_Ready message to PMC if they need to fetch fuses/soft straps, these two signals are redundant

			○ With the requirement to use the "_force_rst_b" signals asynchronously, the sequencing of the separate "_force_*_rst_b" signals reduces to that equivalent to a single "_force_rst_b" signal. The signal "pgcb_force_rst_b" is the single signal that should be used for forced reset assertion and de-assertion. ○ Both these signals are still included in the interface, but will be removed in a future release. Refer to IO table for details on how these should be driven or used by IP blocks. • Other misc changes: ○ Added description for Warm reset support, and separate waveform for Warm reset entry/exit has been provided ○ Added placeholder inputs for extra timer values (intent is to keep PGCB interface same even if more timer values are needed in future revs) ○ Added description for pre-commit based power-gating flow (using CDC) and updated block diagram to include CDC within IP. ○ Edits to documentation/waveform based on review of v0.61and feedback from IP teams
0.71	WW6.4	Jwhavica	• Changing version number to 0.71 to indicate that this no longer a release candidate. ○ No documentation changes for PGCB ○ No Interface changes for PGCB nor CDC ○ Minor bug fix in PGCB relating to latching of pg_type ○ Some bug fixes in CDC (refer to CDC documentation)
0.8	WW10.5		• Version 0.8 release with following updates ○ [HSD: 1274074] Additional documentation on "Support for Turning on Clocks for PG Exit flow" feature for context propagation/synchronous reset logic. ▪ cfg_trsvd0 and cfg_trsvd1 are now used

			○ [HSD: 1274295] IP-Accessible exit now waits for all_pg_rst_up before deasserting force_clks_on ○ [HSD: 1274297] Removed force_prim_rst_b and ip_fuse_valid ○ Increase size of pgcb_visa from 16 to 24 bits ○ [HSD: 1274669] Documentation of changes to support DFx requirements
1.0	WW22.4		• Version 1.0 release with following updates: ○ [HSD: 1274704] - Removal of workaround to keep sleep control signal de-asserted in IP-Inacc state ○ [HSD 1276045] - Additional DFx overrides on pgcb outputs (pgcb_force_clks_on, pgcb_pwrgate_active, etc.) ○ [HSD 1276108] - Change ftap_tck port name to pgcb_tck – **interface change (to DFX input signal name)** ○ Added recommended values to timer inputs for the PGCB and additional implementation notes in section on “Timer and Configuration Signal Requirements” ○ Added section on “Waivers Around Usage of PGCB output signals” to indicate waivers that may be applicable to certain PGCB output signals. Also added note on PGCB Waiver list in Appendix. ○ Clarifications on the definition and requirements associated with “pgcb_pwrgate_active” signal. ○ Updated waveforms to remove IOSF-SB and IOSF-Pri buses, added waveforms for IP-Accessible PG with sleep_en == 0, and some other updates. ○ Updated requirements related to locking/unlocking of IOSF ISMs and for De-assertion of clkreqs for all clocks used only in the PGD.

			○ Updated recommendations related to PGCB clock. ○ Added design notes in section on Latch Based Isolation. ○ Added note on DFx considerations for TAP logic etc. in Appendix.
1.1	WW27.4	hsingh1	• Version 1.1 release with following minor updates: ○ Change to reset state for case of DEF_PWRON == '1' (Chassis ECN 1570775) – HSD: 1277156 PGCB/CDC need ability to default to Restore state . ○ Documentation updates (see change bars)
1.15	WW29.1	hsingh1	• Version 1.15 release – no changes to PGCB for this release.
1.20	WW24	Jwhavica	**RTL Changes:** [2281759] Requirement - Provide .sig files for VISA insertion • With this release of the PGCB blocks, we are providing .sig files to be used by the IP to insert VISA within the blocks using the VISA insertion tool. **The *_visa vector outputs are still available but it is recommended that IPs let them dangle and instead use the provided .sig files to enable easier silicon debug**. [2267263] Requirement - Updated to SIP CTECH methodology – requires integrating IPs to include global CTECH libraries in their ACE environment [2247555] Requirement – PGCB to support SCAN Dump • DFx Sequencer updated: ○ Decoupled pgcb_bypass and pgcb_ovr, so the Sequencer can be moved to a specific state before invoking overrides ○ Changed reset state of DFX Sequencer to be Powered On [2244571] [2280823] Requirement - Provided option for PGCB to force on clocks while power is ramping for contention clearing • Renamed *ip_pgcb_frc_clk_srst_en* to *ip_pgcb_frc_clk_srst_**cc**_en*, when set PGCB asserts *pgcb_ip_force_clks_on* and waits for *ip_pgcb_force_clks_on_ack* before requesting power-up from PMC, deassertion of force_clks_on remains the same. [2280825] Requirement – DFx Sequencer now deasserts sleep before deasserting reset to initialize SR cells [2244992] Requirement – New pgcb_sleep2 output added to provide means for multiple state-retention domains (XHCI request) • IP's not using pgcb_sleep2 can let it dangle [2267262] Enhancement – Replaced RTL muxes with mx22 muxes in DFx logic for better glitch prevention [2279699] Defect – Updated boundary_locked to assert when clocks are gated when ISM_AGT_IS_NS is set

			Doc/Assertion/Environment Changes: [2266897] Cleaned up assertions with potential Large Memory Footprints [2244151] Updated IP-Inaccessible waveforms to show pg_rdy_ack_b deasserting at the same time as pwrgate_active [2249492] Called out need to waive Caliber violations regarding logic on reset due to DFx and force_rst_b [2245081] Misc. doc updates [2267412] Moved Reset States Table for DEF_PWRON to IO Table section and clarified that values are only for DEF_PWRON==1 [2247058] Updated doc to show UNGATE_TIMER encodings [2392384] Replaced `ifdef ASSERT_ON with `ifndef SVA_OFF so assertions are enabled by default [2266993] Removed ASSERT_ON in HDL files [2246425] Updated CDC Waivers
1.21	WW50.1'2015	hsingh1, yjkim1	**Version 1.21 release** **RTL Changes:** • [1404446776] Defect – 1.16 Hot Fix for BXT-E0 Warm Reset Bug: Changes related to support for synchronous reset support for Warm Reset flow. ○ [HSD: 1203800828] (BXT-E0) Warm reset issue for PGCB IPs with sync reset using pgcb_force_clks_on signal. • Enhancement – PGCB CTECH map file update – Meta-Stability Required Parameter and define changes. • [1504055452] Enhancement – Rule 60702 which is about no ifdef allowed in ctech map **Doc/Assertion/Environment Changes** [1404351430] PGCB asserts enhancement for config register value change
1.22		yjkim1	**No Changes**
1.23	WW43.5'2016	yjkim1	**RTL Changes:** • TigerLake PCR: [HSD 1405184871] Add new SCAN DFT control signal to PGCB component **Doc Changes:** • [1404088966] DFX override diagram update.

Opens

Table 1: Opens list

#	Description	Status	Comments
1	DFx Support (non-functional and functional)	CLOSED	Rev0.8 release
2	Update waveforms for notes	TABLED	No request is outstanding for this change.
3	Waveforms for case when state retention is disabled (IP-Accessible flows)	CLOSED	Waveforms added.
4	Is ip_all_pg_rst_up relevant for Ip-Inacc PG exit flow?	CLOSED	No, this is not relevant – and waveforms are updated to indicate this.
5	May IP blocks deassert the POK signals in advance of the pgcb_pok signal de-assertion?	CLOSED	This is considered acceptable, since PMC waits for the last ip_pmc_req_b assertion before starting the timer for reset assertion on entry to Sx or host partition reset.
6	Pgcb_clk gating notes	TABLED	**To be published separately through HSD requirements db.**
7	Add table on recommended values for timer inputs	CLOSED	Section on Timer and Configuration Signal requirements is updated.
8	Need to get SD input on alternate clock usage for “Support for Turning on clocks during PG exit flow” feature	CLOSED	Refer “Support for Turning . . . . “ section for more information.

Introduction

The Power Gate Control Block (PGCB) is a sub-IP block intended to be instantiated within an IP to assist in Power Gating of the IP block in accordance with the Chassis Power Gating flows defined in the Chassis Power-gating (PG) HAS. The current version is designed for use by Soft IP (SIP) blocks. There is no intention to develop a common PGCB for the HIPs.

The PGCB is based on the Chassis defined power-gating flows. Please refer to the Chassis PG HAS for details on these flows.

The PGCB takes input from the IP and handles the power-gating related handshake with the PMC. It also sequences the internal resets, isolation controls and sleep signal (state retention control) to the power-gated domain of the IP block.

The PGCB has been designed to operate in conjunction with one or more instances of the Clock Domain Controller (CDC). The CDC component is packaged within the same folder as the PGCB in IRR. Refer to section on **"PG flow based on Independent Pre-commit"** for details on high level interaction between the CDC and the PGCB. For more details, refer to the CDC Integration Guide.

By default, the PGCB is expected to reset to a Power Gated state but the PGCB also supports the option of resetting to a Powered On state. Please refer to the section titled "Power Ungated Default State," for details.

The current version of this document includes the following:

- ⇨ the scope of the PGCB,
- ⇨ block diagram of the PGCB within a IP block,
- ⇨ flow diagram of the PG Entry and PG Exit flows with functional partitioning between PGCB and IP
- ⇨ the interfaces of the PGCB to the IP block and PMC,
- ⇨ requirements for the IP blocks to support the PGCB and PG flows, and
- ⇨ Wave-forms illustrating the sequencing of various signals between the PGCB, IP Block and PMC.

PGCB Scope

Included in scope

The following items are included within the functions performed by the PGCB.

1) **Sequencing for PG entry and Exit:**
    a. The PGCB's core function is to sequence the IP block's power gated domain into and out of power gating.
    b. This includes driving signals for isolation, forcing resets to assert in power-gated domain, controlling the sleep signal for state retention cells, etc.

2) **Support for Chassis PG flows:** The PGCB's intent is to support the Chassis specified Power-gating Flows for IP blocks
    a. PGCB interacts with PMC for signal handshake for actual PG entry (disabling of FET) and PG exit (re-enabling of FET).
    b. PGCB handles the restore request from PMC during the exit from Power-gated state and provides indication to the IP block about the restore support requirement (if and when applicable).

3) **Prim_pok/side_pok assertion/de-assertion control:** The PGCB provides an explicit signal to control the assertion/de-assertion of the prim_pok and side_pok signals (during various PG entry and exit flows) that are part of the IP block's interfaces to the IOSF primary and IOSF-SB fabrics.

4) **Support for Warm Reset flow:** The PGCB supports the Warm reset flow (indicated by the IP to the PGCB through the "ip_pgcb_pg_type" field) and ensures that the IP block PGD remains in a powered up state during this flow (even though the *_pok signals are de-asserted through PGCB control). For this flow, the PGCB only causes the "*_pok" signal de-assertion, but does not step through any other part of the actual PG sequence (such as isolation, force_rst_b, request to PMC to turn off FET block, etc.).

5) **Sync/Async nature of PGCB inputs/outputs:** The PGCB assumes that all its input signals that are driven by the IP block, are synchronous to the PGCB clock. This is assumed in order to avoid having to add redundant delay where synchronization is not needed.
    a. Outputs from PGCB <u>are</u> glitch free to the IP and the PMC (sent as flop outputs). There is one signal that is combinatorial – "pgcb_pwrgate_active".
    b. All inputs to PGCB from the IP are treated as synchronous to PGCB clock.
    c. Only PMC inputs to PGCB are treated as async and are synchronized to PGCB clock before being used.

Out of scope

The following items are NOT included within the PGCB, and hence are OUT of scope of the PGCB. Some of these items may be handled by the CDC component.

1) **Fabric Interface and Configuration storage:** The PGCB does not have either IOSF primary or IOSF sideband interfaces.
    a. PGCB does not comprehend state of fabric ISMs directly.
    b. PGCB does not support receiving any cycles from fabric for configuration. Therefore, power gating and state retention related configuration bits are physically outside the PGCB.

2) **Power-gate and Wake from PMC:** The PGCB does not directly/separately comprehend the PMC power-gate and wake indications (such as ForcePwrgatePOK message, pmc_ip_sw_pg_req_b, pmc_ip_wake, etc.). It is the IP block's responsibility to factor this indication into the consolidated power gating/wake request that it makes to the PGCB. The IP block is responsible for comprehending all the required conditions for power gating/ungating, including any SW triggered power gating, and all configuration signals that impact power-gating. In general, it is the IP block responsibility to decide if power-gating needs to happen, and to detect a wake event from Power-gated state.

3) **Clock control:** PGCB does not directly control any clocks, but expects the IP block to explicitly control the clocks such that all clocks in the PGD are gated during all parts of the PG entry and exit flows. For some IP blocks that need clocks to be turned on during the PG Exit flows for various reasons (such as synchronous resets, or context propagation), the PGCB can be configured to provide the control signals (as handshakes) at appropriate points. Refer to section on "**Support for Turning On Clocks for PG Exit flows**" and PG Exit flow waveforms for details.

4) **IOSF Interface ISM Lock/Unlock:** PGCB does not directly control the locking or unlocking of the ISMs for the IOSF interfaces of an IP block. It expects that the IP block has locked its ISMs before requesting the PGCB for power-gate entry. Also, when the PG exit flow is completed (or when the PGCB signals the beginning of the restore window as part of the PG exit flow), the IP block is expected to unlock the agent ISMs.

5) **Non-IOSF External interface (boundary) Lock/Unlock:** Similar to the IOSF interface ISM lock/unlock, the PGCB expects the IP block to handle the locking/unlocking of all Non-IOSF external interfaces. All non-IOSF external interface need to be locked down by the IP block before requesting the PGCB for power-gate entry. During PG exit, when the PGCB signals the completion of the flow (or completion of the restore window, if applicable), only then the non-IOSF external interfaces should be unlocked by the IP block.

6) **Reset sequencing for Fuse/soft strap pull:** Any reset sequencing that may be required by IP blocks to perform fuse/soft strap fetch during PG exit flow has to be handled outside of the PGCB. For example, IP blocks may require that side_rst_b de-assertion starts the fuse pull, and prim_rst_b de-asserts only after the fuse pull has completed.
    a. The PGCB requires that IP blocks do not perform any fuse pulls during exit from IP-Accessible PG state.
    b. PGCB assumes that it is SOC/PMC responsibility to sequence the external reset de-assertion into the IP appropriately for the IP-Inaccessible PG Exit flow.

PGCB and IP Block Diagram

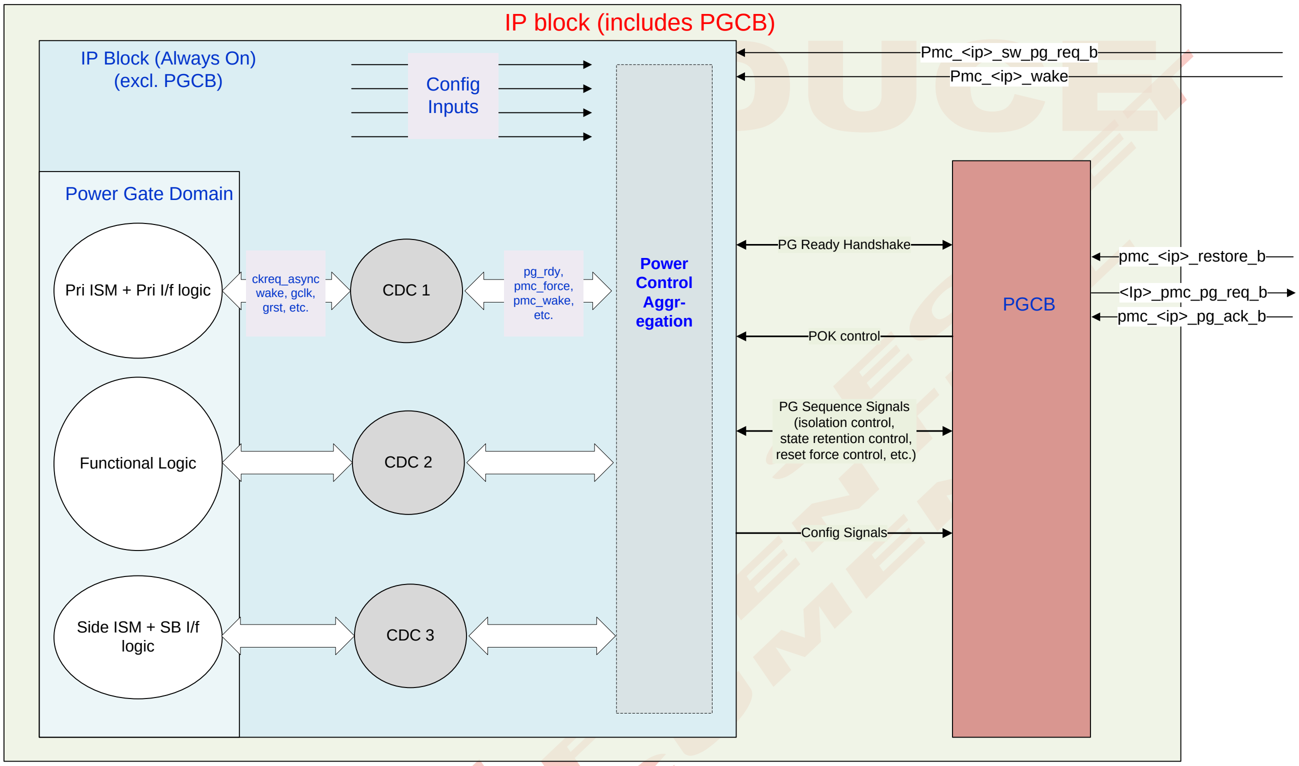


Figure 1: High level overview of PGCB in IP block

PG flow based on Independent Pre-commit

Power-gate Ready state

The idea of independent pre-commit based power-gating proposes that there are three high level states of any IP block's power-gated domain (PGD). When the PGD is fully functional, it is in the "Active" state. When the PGD is power-gated, it is in the "power-gated" state.

There is another intermediate state – "power-gate ready". This is applicable on a per clock domain basis. For a given clock domain, the "power-gate ready" state implies that the clock to the PGD is turned off (gated), ISM (if applicable) is locked and the boundary is locked, too.

When all clock domains in the PGD of the IP attain a "power-gate ready" state, only then the PGD may move to the power-gated state.

Flow sub-steps

1. The logic within the power-gated domain (PGD) of the SIP block is recommended to be sub-divided into smaller blocks on a per clock domain basis. Some examples of these blocks are the logic for IOSF-Primary interface (using the prim_clk), IOSF-SB endpoint logic (using the side_clk), etc.

2. Each of these sub-blocks independently decides whether it has met the requirements for power-gating or not (these would include some configuration settings, idle state, hysteresis timers, etc.).
3. When a given sub-block meets all requirements for power-gating, it proceeds to proactively place itself in a state appropriate for power-gating (i.e. pre-commits to power-gating) – for example by locking IOSF interface ISM, boundary interfaces, gating clock to the PGD, etc.
4. When all of the sub-blocks are in a state of readiness for powergating, the IP initiates the actual sequencing for the PG entry (through PGCB) which includes PG request to PMC (refer figure below).

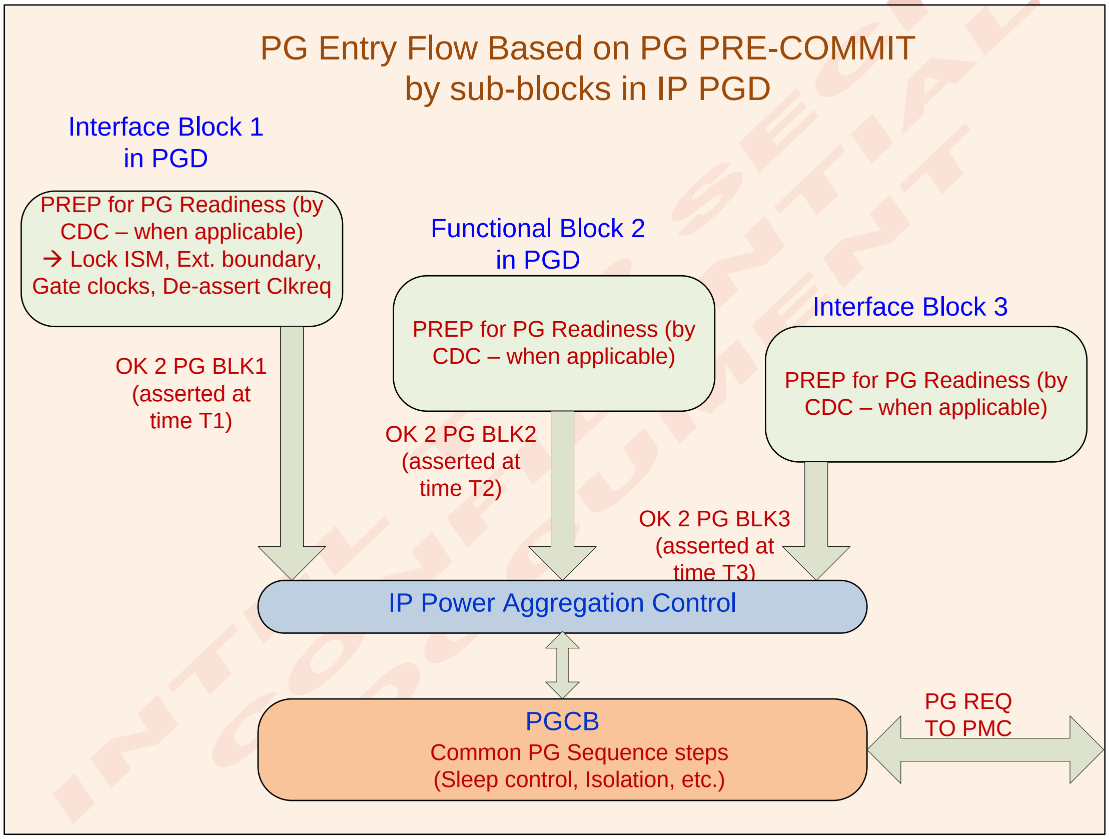

PG Entry Flow Based on PG PRE-COMMIT
by sub-blocks in IP PGD
Interface Block 1
in PGD
PREP for PG Readiness (by CDC – when applicable)
→ Lock ISM, Ext. boundary, Gate clocks, De-assert Clkreq
Functional Block 2
in PGD
PREP for PG Readiness (by CDC – when applicable)
Interface Block 3
PREP for PG Readiness (by CDC – when applicable)
OK 2 PG BLK1 (asserted at time T1)
OK 2 PG BLK2 (asserted at time T2)
OK 2 PG BLK3 (asserted at time T3)
IP Power Aggregation Control
PGCB
Common PG Sequence steps
(Sleep control, Isolation, etc.)
PG REQ TO PMC

Figure 2: High Level Overview of PG Entry Flow

5. When any of the sub-blocks detects a wake event, it requests back its clock. If the IP is either power-gated or in the middle of a power-gating sequence, the sub-block waits (till the PG exit is complete) and then proceeds to bring itself back to active state (refer figure below).
6. If the IP is neither power-gated nor in the middle of a power-gating sequence when a wake event is seen, the relevant sub-block/s proceed back to the active state (once the clock is available).

Note that the Clock Domain Controller (CDC) IP block supports the pre-commit based power-gating flow. Therefore, it is strongly recommended that SIP blocks use the CDC (one or more instances) in conjunction with the PGCB to implement support for power-gating.

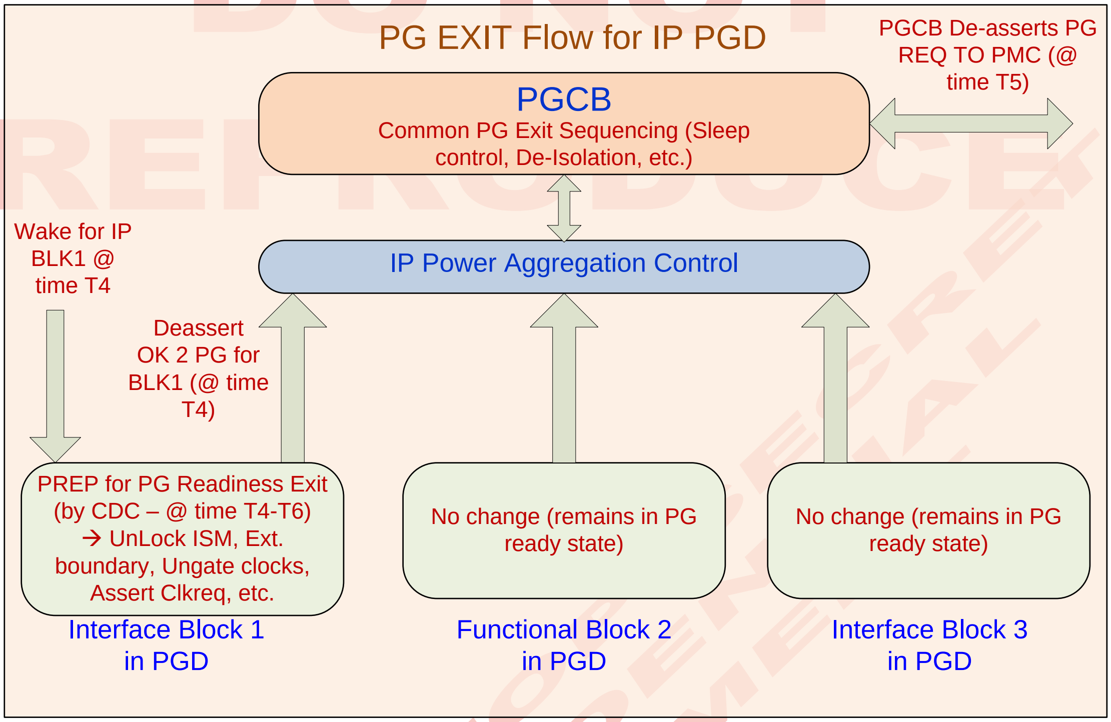

PG EXIT Flow for IP PGD
PGCB De-asserts PG REQ TO PMC (@ time T5)
PGCB
Common PG Exit Sequencing (Sleep control, De-Isolation, etc.)
IP Power Aggregation Control
Wake for IP BLK1 @ time T4
Deassert OK 2 PG for BLK1 (@ time T4)
PREP for PG Readiness Exit (by CDC – @ time T4-T6) → UnLock ISM, Ext. boundary, Ungate clocks, Assert Clkreq, etc.
No change (remains in PG ready state)
No change (remains in PG ready state)
Interface Block 1 in PGD
Functional Block 2 in PGD
Interface Block 3 in PGD

Figure 3: High Level Overview of PG Exit Flow

PGCB <-> PMC Interface

Interface Assumptions

- Outputs from the PGCB to the PMC will be glitch free.
- Inputs to the PGCB from the PMC are assumed to be **asynchronous and glitch free**. These are synchronized by the PGCB.

IO Table

The following signals are defined in the Chassis PG HAS. Note that signal names at the boundary of the PGCB have the term “pgcb” which needs to be replaced with the name of the IP block when mapping these signals at the IP top level. Please refer to the table below for details. For functional definition of these signals, the user is referred to the Chassis PG HAS.

The Reset Values and Expected Reset Values listed below pertain only to the case where the PGCB defaults to a Power Gated state. Please refer to the section titled “Power Ungated Default State,” for Reset Values and Expected Reset Values for a PGCB that is configured to default to a Powered on state.

Signal Name	Input/Output	Description
pgcb_pmc_pg_req_b	Output	Active Low, Power Gating request signal on behalf of the IP to the central power gating controller (in PMC). This signal is part of a two signal handshake involving: • **pgcb_pmc_pg_req_b** • **pmc_pgcb_pg_ack_b** Please refer to the Chassis PG HAS for the details of this handshake and the meaning of these signals. Note: The Chassis PG HAS refers to this signal as **ip_pmc_pg_req_b**. Note: “pgcb” in pgcb_pmc_pg_req_b should be replaced with the ip specific name when mapping out to top level SOC. For example: pgcb_pmc_pg_req_b -> <ipname>_pmc_pgd<domain#>_pg_req_b Reset Value: 1’b0
pmc_pgcb_pg_ack_b	Input	Active low, Power Gating acknowledge from PMC to the PGCB (on behalf of the IP). This signal is part of a two signal handshake involving: • **pgcb_pmc_pg_req_b** • **pmc_pgcb_pg_ack_b** Please refer to the Chassis PG HAS for the details of this handshake and the meaning of these signals.

		Note: The Chassis PG HAS refers to this signal as **pmc_ip_pg_ack_b**. Note: “pgcb” in pmc_pgcb_pg_ack_b should be replaced with the ip specific name when mapping out to top level SOC. For example: pmc_pgcb_pg_ack_b -> pmc_<ipname>_pgd<domain#>_pg_ack_b Expected Reset Value: 1’b0
pmc_pgcb_restore_b	Input	Active low signal from PMC to indicate that the IP must move through the RESTORE state on a PG exit. This signal is valid on the rising edge of pmc_pgcb_pg_ack_b (which indicates power has been returned to the IP). If pmc_pgcb_restore_b is low, it implies that PMC needs to restore some state information to the IP block before the IP block is allowed to open up all its interfaces. Please refer to the Chassis PG HAS for the details of this signal and the intent of the RESTORE window. Note: The Chassis PG HAS calls this signal **pmc_ip_restore_b**. Note: “pgcb” in pmc_pgcb_restore_b should be replaced with the ip specific name when mapping out to top level SOC. For example: pmc_pgcb_restore_b-> pmc_<ipname>_pgd<domain#>_restore_b Expected Reset Value: 1’b1

PGCB <-> IP Interface (Functional Signals)

Interface Assumptions

- Outputs from the PGCB to the IP are glitch free (all outputs are flopped except for one signal “pgcb_pwrgate_active”).
- Inputs to the PGCB from the IP are assumed **synchronous** to the PGCB’s clock. The IP is responsible for synchronizing all these inputs to the PGCB clock.

IO Tables

The Reset Values and Expected Reset Values listed below pertain only to the case where the PGCB defaults to a Power Gated state. Please refer to the section titled “Power Ungated Default State,” for Reset Values and Expected Reset Values for a PGCB that is configured to default to a Powered on state.

Functional Signals

Signal Name	Input/Output	Description
clk	Input	Clock used by the PGCB. The PGCB does not make any assumptions on the type of clock used (gateable vs non-gateable). The choice and hierarchy of clock is IP specific. The PGCB provides hints of when the pgcb clocks can be gated through assertion of **pgcb_idle**.
pgcb_rst_b	Input	Active low reset indication to the PGCB. This reset is expected to be asynchronously asserted to ‘0’ and synchronously deasserted to ‘1’ with respect to “clk” This reset must be exposed by the IP to the SOC as per Chassis Reset Arch HAS. Refer to that HAS for details on different options to drive this reset.
ip_pgcb_pg_type[1:0]	Input	Signal driven by the IP to indicate the type of Power Gating it is requesting (IP-Accessible, IP-Inaccessible or Warm Reset). ‘00’: IP-Accessible Power Gating ‘01’: Warm Reset ‘10’: Reserved ‘11’: IP-Inaccessible Power Gating Note: This signal is sampled by the PGCB when the “ip_pgcb_pg_rdy_req_b” is asserted AND both “pgcb_ip_pg_rdy_ack_b” and “pgcb_idle” are “1” (implying that the PGCB SM is in a state where it is ready to accept a request for PG entry). Therefore, if the IP is driving this signal and the “ip_pgcb_pg_rdy_req_b” signal in the pgcb_clk domain, it is required that the latest this

		signal can change to the desired value is the same clock as the assertion of the “ip_pgcb_pg_rdy_req_b” signal. For the case that the IP drives this signal and the ip_pgcb_pg_rdy_req_b signal in another clock domain, and immediately double-syncs it to the pgcb_clk just at the boundary of the PGCB, it is required that this signal be valid at least one source clock before the change in the source clock version of the ip_pgcb_pg_rdy_req_b signal. This timing guarantees that this signal crosses the double-sync clock crossing logic either earlier or at the same time as source clock version of the ip_pgcb_pg_rdy_req_b signal. Expected Reset Value: “11”
ip_pgcb_pg_rdy_req_b	Input	Active low signal from the IP to the PGCB indicating a request to power gate. This signal is only asserted by the IP when the IP specific and/or Chassis specific required conditions to power gate have been met. This signal is part of a two signal handshake involving: • **ip_pgcb_pg_rdy_req_b** • **pgcb_ip_pg_rdy_ack_b** NOTE: This signal is not required to follow the standard 4-phase handshake. The PGCB still functions properly if the “ip_pgcb_pg_rdy_req_b” deasserts after one (PGCB) clock or more of assertion without the “pgcb_ip_pg_rdy_ack_b” having yet asserted. In this case however, the PGCB completes the entire PG Entry flow and only then starts the PG Exit flow. **Important:** However, for the case of IP-Acc PG followed by IP-Inacc PG (or Warm Reset Entry), it **is *required*** for IP blocks to not assert this signal again to the PGCB (with valid type corresponding to IP-Inacc PG or Warm Reset) until at least one clock after the PGCB de-asserts the “pgcb_pwrgate_active” signal. IP blocks that use the CDC component need not do anything specific to meet this requirement since the CDC comprehends this requirement (please refer to CDC Integration doc). For IP blocks that do not use the CDC, it may be simpler to follow the standard 4-phase handshake between this signal and the “pgcb_ip_pg_rdy_ack_b” signal. However, the actual requirement is as stated above.

		Expected Reset Value: 1’b0
pgcb_ip_pg_rdy_ack_b	Output	Active low signal from PGCB to the IP indicating positive acknowledgment of power gating request. This signal is part of a two signal handshake involving: • **ip_pgcb_pg_rdy_req_b** • **pgcb_ip_pg_rdy_ack_b** Note: IP blocks should never gate/qualify any of the following signals with this signal: • pgcb_sleep • pgcb_isol_en_b • pgcb_force_rst_b The above requirement is related to the fact that the signals listed above are intended to be used through overrides in DFx modes (such as HVM) and no functional signal should interfere with the function of these signals. Reset Value: 1’b0
pgcb_pok	Output	Active high, level signal from the PGCB which should be used to drive the IP block’s side_pok and prim_pok signals to the fabric interfaces. The IP is required to synchronize this to the respective clock domain of the fabric/router before driving the appropriate *_pok signal. Reset Value: 1’b0
pgcb_restore	Output	Active high, level signal from the PGCB indicating to the IP that it is in the restore phase of the PG Exit flow. As per the Chassis PG spec, when this signal asserts the following must happen (so that PMC or some other platform agent may be able to fully restore the state of the IP block): • IOSF ISMs must be unlocked • All clocks are allowed to be ungated • All other boundaries of the IP block need to remain locked. The boundary lock is required to be kept asserted till either “pgcb_restore” de-asserts or

		"pgcb_ip_pgrdy_ack_b" de-asserts. Refer to the CDC Integration Guide for details of boundary lock function. Reset Value: 1'b0
pgcb_restore_force_reg_rw	Output	Active high, level signal from the PGCB indicating to the IP that all configuration registers need to be accessible with RW attributes over the primary and SB IOSF interfaces (so that PMC or some other platform agent may be able to fully restore any values to these registers). Refer to the section on IP Power Gating requirements for details on the functionality required for this signal. Reset Value: 1'b0
pgcb_isol_en_b	Output	Active low, level isolation enable signal from the PGCB to the isolation cells on the outputs of the PGD. This signal will be used as an asynchronous qualifier for all signals coming out of the PGD. This signal is relevant only for isolation of signals crossing from the PGD to the always-on domain. This signal is intended to be used for asynchronous isolation between the PGD logic and the ungated/AON logic. However, each IP block is required to provide any and all applicable waivers for static timing analysis related to this signal. Note that in some cases, the isolation signal assertion may cause an asynchronous transition on the isolation gate output. It is IP responsibility to make a judgment if such a scenario is possible, and the resolution thereof. The PGCB indicates that this may qualify for a waiver, but eventually, the IP blocks own responsibility for specifying all applicable waivers. Reset Value: 1'b0
pgcb_isol_latchen	Output	Active high, level signal used to provide latch isolation control if isolation latches are used. It is recommended that this signal be used for both Resettable and Non-Resettable Isolation Latches. Reset Value: 1'b0
pgcb_force_rst_b	Output	Active low, level signal that is intended to force the assertion of **all** the PGD's internal resets.

		It is recommended that IP blocks use this signal to gate their functional reset directly without synchronization – since the signal changes only when all the clocks in the PGD are gated. If an IP block chooses to synchronize this reset into its respective clock domain(s) prior to use (such that the de-assertion/rising edge is synchronous, while assertion edge (falling edge) is asynchronous), it needs to force clocks ON (during context propagation phase) to propagate the reset de-assertion synchronously. This is done by setting the input signal "ip_pgcb_frc_clk_cp_en" to 1. The IP is responsible for summing its existing "native reset" with this signal in order to factor in fundamental reset conditions. Reset Value: 1'b0 **IMPORTANT: T**his signal is intended to force ALL internal resets of the PGD. Any sequencing that is needed between side_rst_b and prim_rst_b domains (applicable for IP-Inaccessible exit flows) is provided by PMC based on IP_Ready message sent by the IP block to PMC.
ip_pgcb_all_pg_rst_up	Input	Active high, level signal that is used in the PG Exit flow to indicate to the PGCB that all sequential logic in the PGCB is out of reset. IP blocks that use the "pgcb_force_rst_b" signal in asynchronous manner (for both assertion and de-assertion of resets), this input must be tied high to 1'b1. Refer to Recommendations section "Reset Propagation Independent of Clocks" for details about using the "pgcb_force_rst_b" signal in asynchronous manner. This signal is required to be driven dynamically only for those IP blocks which connect the "pgcb_force_rst_b" to the input of reset synchronizers – and do not have a direct bypass path to assert/de-assert the reset (note that these IP blocks require clocks to run for the propagation of reset de-assertion). For these IP blocks, this signal must be asserted if and only if all resets with in the PGD are de-asserted, and this signal should remain de-asserted otherwise.

		Expected Reset Value: N/A (Don't Care)
pgcb_sleep	Output	Active High, level signal from the PGCB block to all State Retention flops/latches in the respective PGD. This signal asserts or deasserts as part of the power gating flows. Reset Value: 1'b1
pgcb_sleep2	Output	Active High, level signal from the PGCB block to State Retention flops/latches that should not be disabled when ip_pgcb_sleep_en=='0'. This signal always behaves the same as pgcb_sleep does when ip_pgcb_sleep_en=='1', regardless of the value of ip_pgcb_sleep_en. This signal can be used to create another State Retention "domain," such that SR cells controlled by pgcb_sleep can conditionally retain state (based on ip_pgcb_sleep_en), and SR cells controlled by pgcb_sleep2 would always be retained. IP's that have no need for this signal can let it dangle and only use pgcb_sleep. Reset Value: 1'b1
pgcb_idle	Output	Active high level signal from PGCB block to indicate that the PGCB is idle and that its clocks may be gated, until such time that another power-gate transition is required. Note: IP clock gating for pgcb_clk should follow any/all applicable Scan/DFx guidelines. Reset Value: 1'b0
pgcb_ip_force_clks_on	Output	Active high signal from PGCB to the IP indicating that IP block must turn on clocks in the PGD. This signal is applicable only for the PG Exit flow for those IP blocks that need to either propagate reset synchronously and/or need to propagate context within the PGD subsequent to reset de-assertion. This signal is part of a two signal handshake involving: • **pgcb_ip_force_clks_on** • **ip_pgcb_force_clks_on_ack** These two signals follow the standard 4-phase handshake protocol.

		By default, the PGCB does not assert this signal in any PG exit flow. Please refer to section "Support for Turning on Clocks during PG Exit Flows" for details of how the PGCB may be configured to assert this signal as part of the PG exit flow. Reset Value: 1'b0
ip_pgcb_force_clks_on_ack	Input	Active high signal from the IP to the PGCB indicating acknowledgment of the request to force turn-on of clocks. Assertion of this signal (in response to assertion of "pgcb_ip_force_clks_on") implies that PGD clocks have been turned ON and any time needed (if applicable) for reset assertion to propagate within the PGD has been met. De-assertion of this signal in response to de-assertion of "pgcb_ip_force_clks_on" signal implies that any time needed for context propagation (if applicable) within the IP block's PGD has been met and that clocks have now been fully turned OFF. This signal is part of a two signal handshake involving: • **pgcb_ip_force_clks_on** • **ip_pgcb_force_clks_on_ack** These two signals follow the standard 4-phase handshake protocol. Expected Reset Value: 1'b0
pgcb_pwrgate_active	output	Active high signal from the PGCB to the IP. When asserted, it indicates that either a PG Entry request has just been received (in same clock cycle), or the PGCB has put the PGD in power-gated state, or the PGCB is proceeding through either the PG Entry or the PG Exit flow. IMPORTANT: As long as this signal is asserted, none of the blocks within the PGD are allowed to be brought out of "power-gate ready" state ("PG Flow based on Independent Pre-commit" section defines the "power-gate ready" state). In other words, the clocks to the IP PGD have to remain gated, ISMs in PGD have to remain locked and boundaries of the IP remain locked while this signal is asserted.

		The are two exceptions to the above, which are listed below: – When the “pgcb_restore” signal is asserted (applicable only when PMC indicates that a restore phase is needed for the PG exit flow), the PGD is required to be brought out of “power-gate ready” state (even though the pgcb_pwrgate_active signal remains asserted through the restore phase). The “pgcb_pwrgate_active” signal de-asserts only at the end of the restore phase. Refer to section on “IP Requirements for Restore phase” for details. – The IP may need to force clocks during PG exit flow (refer to section on Support for Turning on Clocks for PG Exit flows). In this case, the clocks to the IP PGD are driven through an alternative available clock source (not the regular functional clocks) for some part of the PG exit flows. This happens even though the “pgcb_pwrgate_active” signal remains asserted during this time. Note: This output is combinatorial and is not guaranteed to be glitch free. It asserts as long as one of the following is true: – “ip_pgcb_pg_rdy_req_b” is asserted – “pgcb_idle” is de-asserted – “pgcb_ip_pg_rdy_ack_b” is asserted
pgcb_visa[23:0]	output	The latest recommendation is that the integrating IP have the VISA tool automatically insert VISA in the PGCB and that the pgcb_visa output be ignored. See the provided .sig files under $MODEL_ROOT/tools/visa. Vector containing internal state information for debug visibility. This output should be tied into the IP’s VISA muxes. It is recommended that the IP have at least a small VISA ULM in the Always-ON domain which the PGCB and CDC VISA signals should be connected to.

Configuration Signals

Signal Name	Input/Output	Description
ip_pgcb_sleep_en	Input	Active high indication to provide dynamic enable of **pgcb_sleep** assertion/deassertion to enable state-retention when required during the power gating flows. **This signal is only relevant when doing IP-Accessible power gating.** This signal should correspond with the SE (Sleep Enable) bit in the PCE (Power Control Enables) register defined in the Chassis PG HAS, unless the IP/SoC defines additional straps to override it. In all cases, the signal source driving this input is required to be configurable and should default to a value of '1'. '1': Sleep assertion is allowed during power gating entry flow '0': Sleep assertion is disabled during power gating entry flow Note: If this input is 0, the PGCB still asserts the "pgcb_sleep" (during the PG entry flow) before it asserts the "pgcb_pmc_pg_req_b" to PMC. This is done to provide the internal isolation required by Intel's state-retention cells. On the PG Exit flow, the PGCB de-asserts the "pgcb_sleep" before isolation is removed and resets are deasserted (to flush out any value that might be retained). When this input is '0', the "pgcb_sleep" signal sequencing for the IP-accessible flows is similar to the "pgcb_sleep" signal sequencing for IP-Inaccessible PG flows. Note: This signal is sampled (and latched in) by the PGCB when the "ip_pgcb_pg_rdy_req_b" is asserted AND both "pgcb_ip_pg_rdy_ack_b" and "pgcb_idle" are "1" (implying that the PGCB SM is in a state where it is ready to accept a request for PG entry). Therefore, if the IP is driving this signal and the "ip_pgcb_pg_rdy_req_b" signal in the pgcb_clk domain, it is required that the latest this signal can change to the desired value is the same clock as the assertion of the "ip_pgcb_pg_rdy_req_b" signal. This configuration signal is not relevant for IP-Inaccessible PG flows – therefore it is okay to latch this inside the PGCB. Also, this corresponds to a configuration bit defined in the Chassis PG HAS.

		Therefore, if this was not latched inside the PGCB, it would require IP blocks to provide isolation latch for this signal (assuming configuration bit is in the PGD of that IP) – which is extra overhead for an IP block that does not intend to use isolation latches otherwise. Based on the above, this signal is latched in the PGCB.
ip_pgcb_frc_clk_cp_en	Input	Active high indication to indicate to the PGCB that the “pgcb_ip_force_clks_on” signal needs to be asserted as part of the IP-Accessible PG Exit flow to meet the context propagation requirement for a specific IP block. Refer to PG sequence waveforms for the exact sequencing relationship. ‘1’: “pgcb_ip_force_clks_on” assertion is required during IP-Accessible PG Exit flow to meet the context propagation requirement ‘0’: “pgcb_ip_force_clks_on” is not required to be asserted during IP-Accessible PG Exit flow due to context propagation requirement. This signal is assumed to be Static (not latched inside the PGCB). Refer to section on **“Timer and Configuration Signal Requirements”** for details.
ip_pgcb_frc_clk_srst_cc_en	Input	Active high indication to indicate to the PGCB that the “pgcb_ip_force_clks_on” signal needs to be asserted as part of the PG Exit flows (both IP-Accessible and IP-Inaccessible) and exit portion of the Warm Reset Flow to cause any synchronously reset logic to get into reset state and/or to perform contention clearing while the domain is powering up. Refer to PG sequence waveforms for the exact sequencing relationship. If contention clearing is needed for a given PGD, when this signal is set, it will request the clocks to run before it requests power from PMC and will request the clocks to stop before it deasserts reset. ‘1’: “pgcb_ip_force_clks_on” assertion is required during PG Exit flow (both IP-Accessible and IP-Inaccessible) and Warm Reset Flow for the synchronous reset requirement ‘0’: “pgcb_ip_force_clks_on” is not required to be asserted for the synchronous reset requirement.

		This signal is assumed to be Static (not latched inside the PGCB). Refer to section on **"Timer and Configuration Signal Requirements"** for details.
cfg_tsleepact[1:0]	Input	The cfg_t* signals are used to specify the minimum number of delay clocks the PGCB should wait between various states. Please refer to the section on **"Timer and Configuration Signal Requirements"** for details regarding where these values are used. **Common for all cfg_t* inputs:** '00': 1 clock '01': 2 clocks '10': 8 clocks '11': 256 clocks For any of the cfg_t* inputs that an IP may feel are a don't-care and the IP does not feel they will be useful for post-silicon debug, the recommendation is to tie the input to 2'b00 (1 clock). Otherwise, the inputs may be tied to another relevant value or connected to a configuration register. All the timer values are assumed to be static by the PGCB, and are not latched inside the PGCB. Please refer section on **"Timer and Configuration Signal Requirements"** for details.
cfg_tisolate[1:0]	Input	Please see the description of cfg_tsleepact.
cfg_trstdown[1:0]	Input	Please see the description of cfg_tsleepact.
cfg_tdeisolate[1:0]	Input	Please see the description of cfg_tsleepact.
cfg_tlatchen[1:0]	Input	Please see the description of cfg_tsleepact.
cfg_taccrstup[1:0]	Input	Please see the description of cfg_tsleepact.
cfg_tinaccrstup[1:0]	Input	Please see the description of cfg_tsleepact.
cfg_tsleepinactiv[1:0]	Input	Please see the description of cfg_tsleepact.
cfg_tlatchdis[1:0]	Input	Please see the description of cfg_tsleepact.
cfg_tpokdown[1:0]	Input	Please see the description of cfg_tsleepact.
cfg_tpokup[1:0]	Input	Please see the description of cfg_tsleepact.
cfg_tclksonack_srst[1:0]	Input	Please see the description of cfg_tsleepact.
cfg_tclksoffack_srst[1:0]	Input	Please see the description of cfg_tsleepact.
cfg_tclksonack_cp[1:0]	Input	Please see the description of cfg_tsleepact.
cfg_trstup2frcclks[1:0]	Input	Please see the description of cfg_tsleepact.
cfg_trsvd0[1:0]	Input	Please see the description of cfg_tsleepact.
cfg_trsvd1[1:0]	Input	Please see the description of cfg_tsleepact.
cfg_trsvd2[1:0]	Input	Please see the description of cfg_tsleepact.
cfg_trsvd3[1:0]	Input	Please see the description of cfg_tsleepact.

cfg_trsvd4[1:0]	Input	Please see the description of cfg_tsleepact.

DFx Related Signals

Signal Name	**Input/Output**	**Description**
pgcb_tck	Input	FTAP/ clock used by the PGCB DFx Sequencer. This should be connected to a top-level port of the SIP with the same name.
fdfx_powergood_rst_b	Input	DFx powergood reset used by the PGCB DFx Sequencer. This should be connected to a top-level port of the SIP with the same name.
pmc_pgcb_fet_en_b	Input	The PGD's active-low PFET enable from the PMC. The PFET enable is routed through the PGCB to allow the DFx logic to be able to drive the final value to the PFET power-switch. This signal should be connected to the SIP's top level pfet_en_b signal.
pgcb_ip_fet_en_b	output	The PGD's final active-low PFET enable. This signal should be connected through UPF to the actual PFET power-switch. For cases where multiple PGCBs/Logical PGDs are merged into one physical PGD, only one of the fet_en_b outputs will be connected to the actual PFET control through UPF. In such a case, the IP team should sync up with the SoC Full-chip team and DFx team.
fdfx_pgcb_bypass	Input	PGCB DFx Bypass Enable. This should be connected to a top-level port of the SIP with the same name.
fdfx_pgcb_ovr	Input	PGCB DFx Sequencer Control. '0' – Power Up '1' – Power Down This should be connected to a top-level port of the SIP with the same name.
fscan_mode	input	DFX Scan Mode Enable. This should be connected to a top-level port of the SIP with the same name.
fscan_ret_ctrl	input	DFX Scan Control for pgcb_sleep. This should be connected to a top-level port of the SIP with the same name.

Timer and Configuration Signal Requirements

Any timer and/or other configuration signal that is used by the PGCB for the IP-Inaccessible PG Exit flow or for the IP-Inaccessible PG Entry flow needs to be either hard-wired or driven from the AON/Ungated domain. It is recommended that all timers used for the IP-Inaccessible PG Exit flow **be hard-wired to a safe value** or have a **safe default value** if driven from configuration registers in AON domain (since these are used in the first boot, with possibly no easy way to access them before these are used).

Note that there are several timer values that are common to both IP-Accessible and IP-Inaccessible flows. For these timer signals, the requirements (given above) for IP-Inaccessible flow are applicable. Under no condition should such timer signals be driven from the IP block's PGD.

The configuration signal, "ip_pgcb_frc_clk_srst_cc_en" is applicable for both IP-Inaccessible PG Exit flow and IP-Accessible PG Exit flow. Hence, this cannot be driven from a configuration register in the PGD of the IP. One option is to drive this through a strap or hard-wire this to a specific value. If configurability is really needed, the IP block should drive this signal from a register in the AON/ungated domain.

All others may be driven from configuration registers within the PGD, but these need to be passed through isolation latches in that case. Some or all of these signals may also be hard-wired.

The PGCB implements a shared counter that is loaded with different times based on the cfg_t* inputs. The following table describes the PGCB FSM states where the various cfg_t* inputs are used.

For recommended values of the timers, these are the following considerations:

1) For values that relate to analog behavior (such as isolation activation or removal, sleep signal assertion or de-assertion), 2 clocks or larger values should be used.

2) Two sets of values are provided – the first related to IP blocks that use a slow clock as the pgcb_clk (e.g. 12 MHz). The second set of values (in parentheses) is applicable for IP blocks that use a faster clock as pgcb_clk (and the IP block potentially uses other slower clocks as part of the PGD)

3) For IP blocks that support IP-Accessible PG in Sx/CMOFF state (before BIOS has an opportunity to program the values), the default values for IP-Accessible timers should also be chosen to be "safe" (conservative).

4) The given values are recommendations – not requirements. IP blocks are encouraged to review the design/analyze the analog considerations and other risk factors in selecting the final values.

#	Timer	Conditions	From State	To State	Recommended default values (values in parentheses indicate values to be used if pgcb_clk is a fast clock such as 120 MHz or higher)
1	cfg_tsleepinactiv	IP-Accessible Exit w/ State Retention	ACCSRETLOW	ISOLLATCHEN	2 clocks (8 clocks)
		IP-Accessible Exit w/o State Retention	INACCSRETLOW	INTISOLDIS / CLKSON_SRST	

		IP-Inaccessible Exit	INACCSRETLOW	INTISOLDIS / CLKSON_SRST	
2	**cfg_tdeisolate**	IP-Accessible Exit	INTISOLDIS	ACCRSTINACTIV	2 clocks (8 clocks)
		IP-Inaccessible Exit	INTISOLDIS	POKHIGH1	
3	**cfg_tpokup**	IP-Inaccessible Exit	POKHIGH1	INACCRSTINACTIV	2 clock (8 clock)
4	**cfg_tinaccrstup**	IP-Inaccessible Exit	INACCRSTINACTIV	RESTORE	1 clock (2 clocks)
5	**cfg_taccrstup**	IP-Accessible Exit w/ frc_clk_cp_en==0	ACCRSTINACTIV	ACCSRETLOW	2 clocks (8 clocks)
		IP-Accessible Exit w/ frc_clk_cp_en==1	CLKSOFFACK_CP	ACCSRETLOW	
6	**cfg_tlatchen**	IP-Accessible Exit w/ State Retention or ISOLLATCH_NOSR_EN==1	ISOLLATCHEN	RESTORE	2 clocks (8 clocks)
7	**cfg_tpokdown**	IP-Inaccessible Entry / Warm Reset	POKLOW	ISOLLATCHDIS / WARMRST/ WRSTCLKSON	1 clock (2 clocks)
8	**cfg_tlatchdis**	IP-Accessible Entry w/ State Retention or ISOLLATCH_NOSR_EN==1	ISOLLATCHDIS	ACCSRETHIGH	2 clocks (8 clocks)
9	**cfg_tsleepact**	IP-Accessible Entry w/ State Retention	ACCSRETHIGH	INTISOLEN	1 clock (2 clocks)
		IP-Accessible Entry w/o State Retention	INACCSRETHIGH	PWRDNREQ	
		IP-Inaccessible Entry	INACCSRETHIGH	PWRDNREQ	
10	**cfg_tisolate**	All Entry	INTISOLEN	RSTACT	2 clocks (8 clocks)
11	**cfg_trstdown**	All Entry	RSTACT	INACCSRETHIGH	1clock (2 clocks)
12	**cfg_tclksonack_srst**	All Exit w/ frc_clk_srst_cc_en==1	CLKSONACK_SRST	CLKSOFF_SRST	IP-specific (if applicable to IP)
13	**cfg_tclksoffack_srst**	All Exit w/ frc_clk_srst_cc_en==1	CLKSOFFACK_SRST	INTISOLDIS	IP-specific (if applicable to IP)
14	**cfg_tclksonack_cp**	IP-Accessible Exit w/ frc_clk_cp_en==1	CLKSONACK_CP	CLKSOFF_CP	IP-specific (if applicable to IP)
15	**cfg_trstup2frcclks**	IP-Accessible Exit w/ frc_clk_cp_en==1	RSTINACTIV	CLKSON_CP	IP-specific (if applicable to IP)
16	**cfg_trsvd0**	Warm Reset w/ frc_clk_srst_cc_en==1	WRSTCLKSONACK	WRSTCLKSOFF	This is the minimum delay between these states, it is also dependent on ip_pgcb_pg_rdy_req_b deassertion. Value is IP-specific (if applicable).
17	**cfg_trsvd1**	Warm Reset w/ frc_clk_srst_cc_en==1	WRSTCLKSOFFACK	PWRSTBLE	IP-specific (if applicable to IP)
-	**cfg_trsvd2**	N/A	-	-	Reserved for future use
-	**cfg_trsvd3**	N/A	-	-	Reserved for future use
-	**cfg_trsvd4**	N/A	-	-	Reserved for future use

Parameters

Parameter Name	Default Value	Description
DEF_PWRON	0	‘0’ – PGCB will default to an IP-Inaccessible power-gated state. ‘1’ - PGCB will default to a powered on state. Please see the section titled “Power Ungated Default State” for details
ISOLLATCH_NOSR_EN	0	‘0’ – pgcb_ip_isol_latchen will only deassert when state retention is enabled and as part of the IP-Accessible flows, preventing isolation latches from closing if state-retention is disabled (ip_pgcb_sleep_en=0). ‘1’ – pgcb_ip_isol_latchen will deassert/assert at the normal points in the IP-Accessible flows even if state retention is disabled (ip_pgcb_sleep_en=0).
USE_DFX_SEQ	1	‘1’ - The DFx sequencer is used to override the defined outputs from the PGCB. ‘0’ – The DFx sequencer is not used, instead, the outputs of the functional PGCB are latched and held constant based on fdfx_pgcb_bypass. The following outputs will be affected by this feature: • pgcb_isol_en_b • pgcb_isol_latchen • pgcb_force_rst_b • pgcb_sleep • pgcb_ip_fet_en_b As the value is SoC specific, it is recommended that this parameter be brought up to the top level of the SIP for the SoC to be able to define.
UNGATE_TIMER	2’b01	Defines the number of ftap_tck clocks that the DFx override sequencer waits between turning on the FET and deasserting isol_en_b/force_rst_b. Encoding: 1’b00 – 8 clocks 1’b01 – 16 clocks 1’b10 – 32 clocks 1’b11 – 256 clocks As the value is SoC specific, it is recommended that this parameter be brought up to the top level of the SIP for the SoC to be able to define.
DELAY_ASSN_RST	0	If set, the deassertion of the reset used by the assertions within the PGCB will be delayed by 1 clock to ensure they

		see a clock edge with the reset asserted. This is relevant for IPs that use a ungated clock to synchronize pgcb_rst_b while the clock is gated to the PGCB. This parameter does not affect any functional behavior of the PGCB.

Macros (`defines)

PGCB_FORCERST_DIS	If defined, will prevent the pgcb_force_*_rst_b outputs from asserting. This is to enable partial State-Retention emulation in FPGAs. This macro should NOT be defined for actual silicon.
SVA_OFF	If defined, disables the embedded assertions within the PGCB.
DC	If defined, disables the embedded assertions within the PGCB. Intended to be set during synthesis
SVA_FORMAL	If defined, enables certain assertions/assumptions that are required for Formal Property Validation (FPV). Should not be set during simulation.

Reset States (DEF_PWRON==1)

The reset values of the interface signals when DEF_PWRON==0 is shown in the tables above. If the IP defaults to powered-on (DEF_PWRON==1) the reset values are as shown below.

Signal Name	Input/Output	Reset Value / Expected Reset Value
pgcb_pmc_pg_req_b	Output	1'b1
pmc_pgcb_pg_ack_b	Input	1'b1
pmc_pgcb_restore_b	Input	Could be either 1'b1 or 1'b0
ip_pgcb_pg_type	Input	N/A (Don't Care)
ip_pgcb_pg_rdy_req_b	Input	1'b1
pgcb_ip_pg_rdy_ack_b	Output	1'b0
pgcb_pok	Output	1'b1
pgcb_restore	Output	1'b1
pgcb_restore_force_reg_rw	Output	1'b1
pgcb_isol_en_b	Output	1'b0
pgcb_isol_latchen	Output	1'b1
pgcb_force_rst_b	Output	1'b1
ip_pgcb_all_pg_rst_up	Input	N/A (Don't Care)
pgcb_sleep	Output	1'b0
pgcb_idle	Output	1'b0
pgcb_ip_force_clks_on	Output	1'b0
ip_pgcb_force_clks_on_ack	Input	N/A (Don't care)
pgcb_pwrgate_active	Output	1'b1

Waveforms

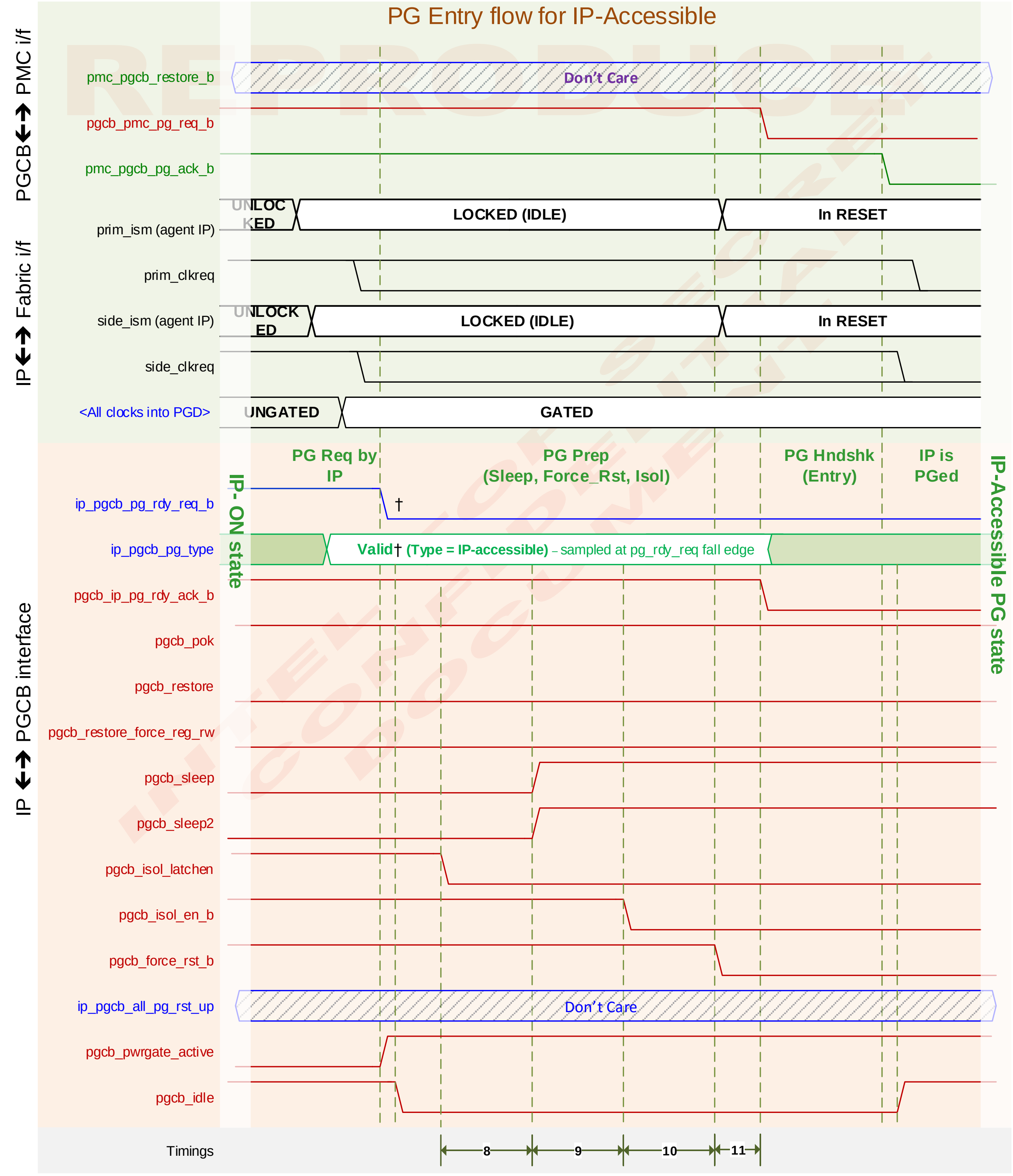
Waveform Notes

For the sequencing of some of the signals at the top level of the IP block, such as side_pok, prim_pok, side_rst_b, prim_rst_b, clkreq, clkack, etc., the reader is referred to the Chassis Reset Architecture HAS. The PGCB integration doc includes these signal in the waveforms with an intent to describe the complete flow, however, the Reset Architecture HAS is the original source for the definition of the sequencing of the above top-level signals at the interface of the IP and SOC. In case of any conflict between the sequencing shown here versus the sequencing depicted in the Chassis Reset Arch. HAS, the Reset Architecture HAS definition prevails.

IP-Accessible Exit (both with and w/out restore):

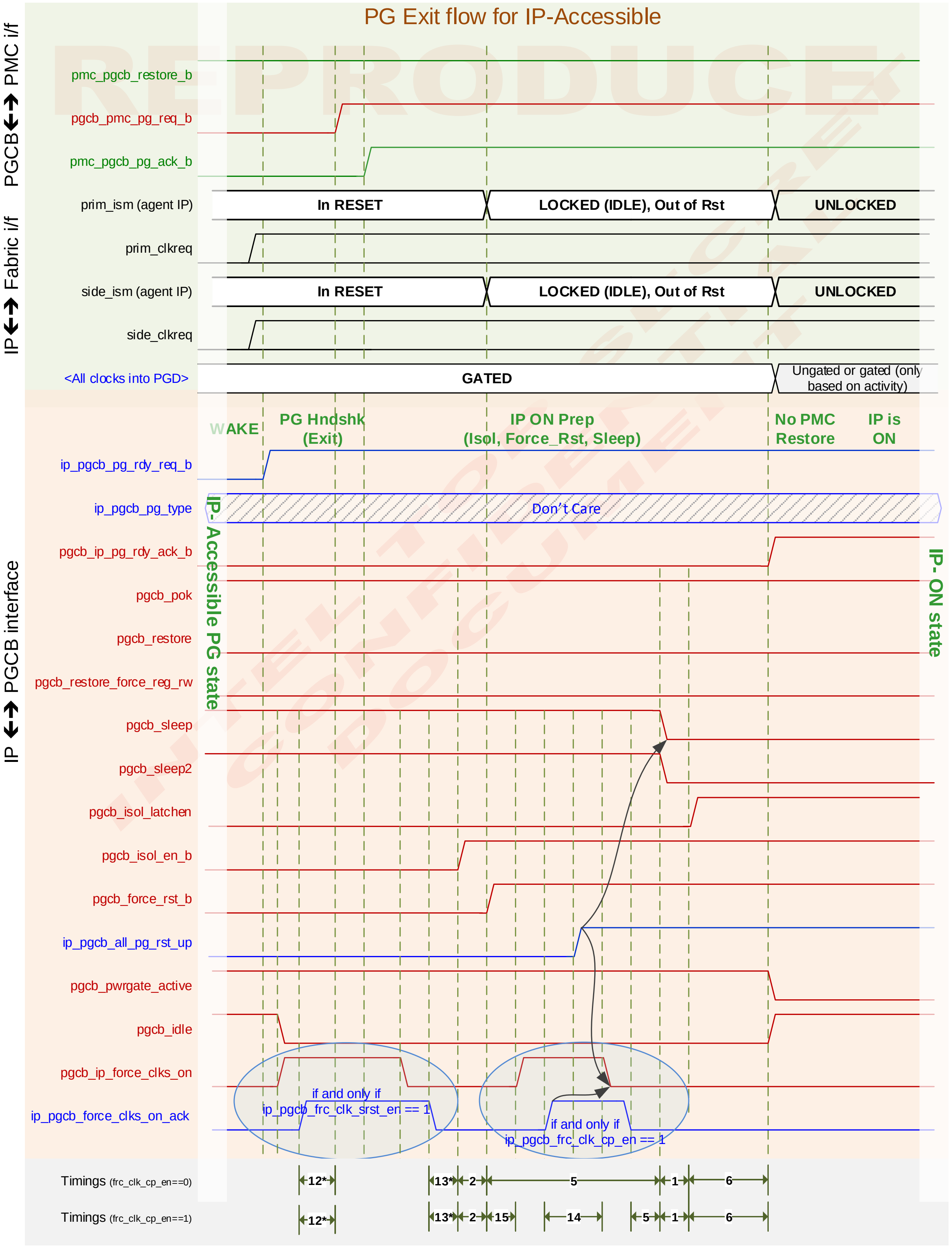
The waveforms indicate that both IOSF-SB and IOSF-Prim interfaces are coming up (clkreq for both side_clk and prim_clk is being asserted) – however, in the general case, it is intended that only those interfaces that really have a wake input need to be waking up and requesting clocks. Other interfaces/clock domains (per CDC) should remain in a “power-gate ready” state (with clkreq de-asserted) until such time they have a specific wake.

IP-Accessible PG Entry



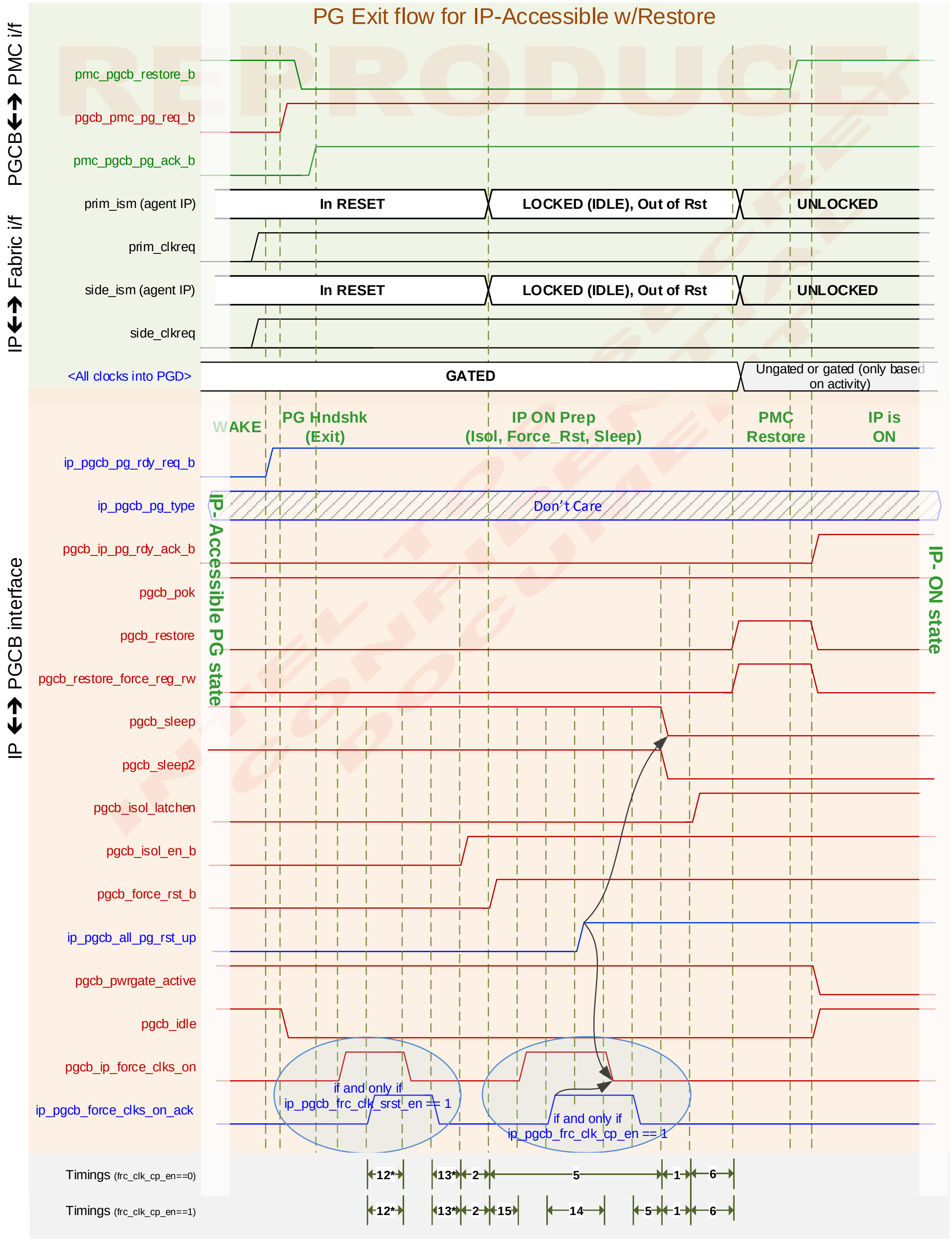
† pg_rdy_req_b and pg_type are allowed to change after they are sampled, although they are shown to be stable here. Please refer to the signal description tables for details on when they are sampled.

IP-Accessible PG Exit



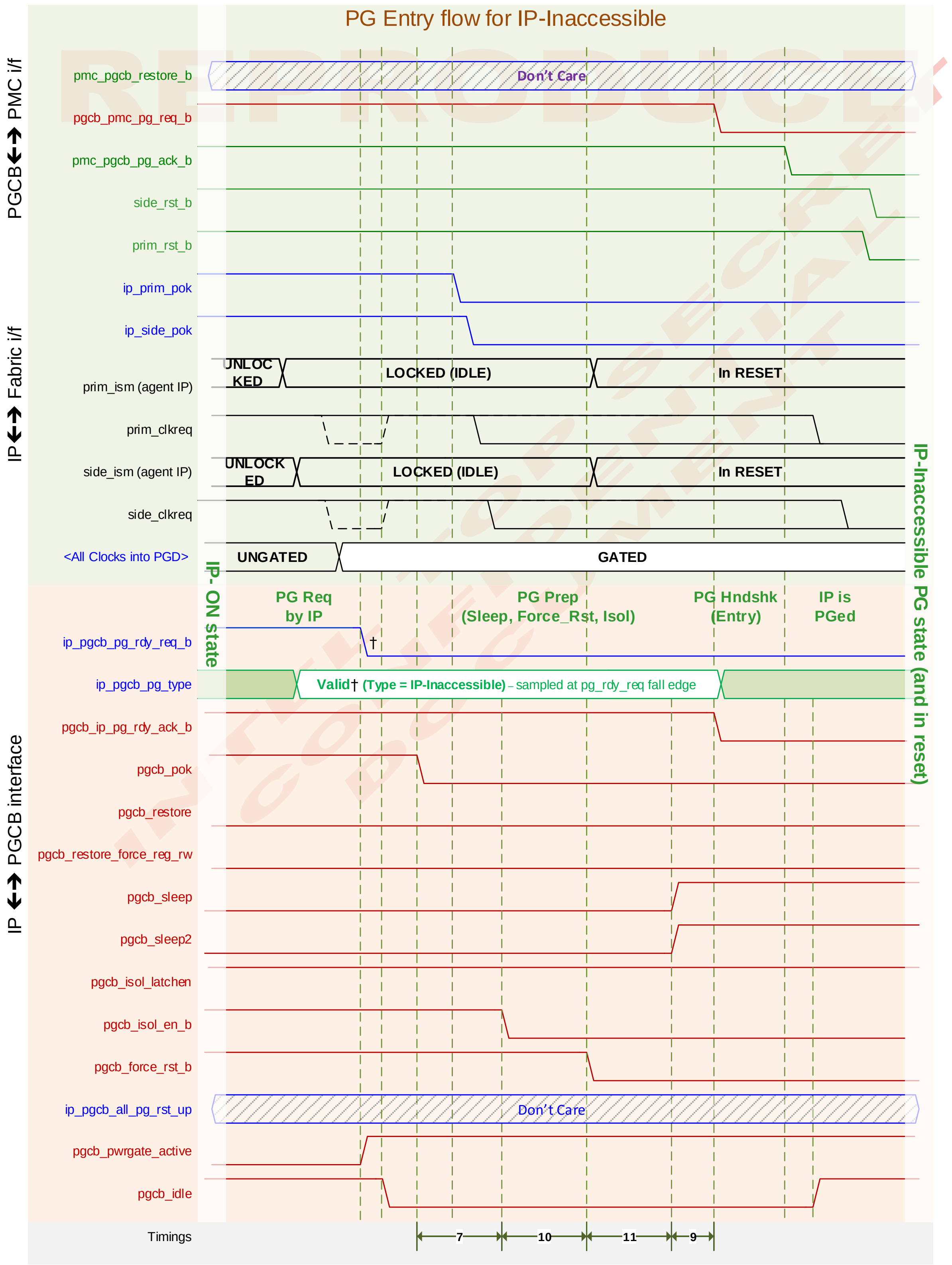
*** Note**: Timings marked with (*****) are only applicable if frc_clk_srst_en==1

IP-Accessible PG Exit with Restore



* **Note**: Timings marked with (*) are only applicable if frc_clk_srst_en==1

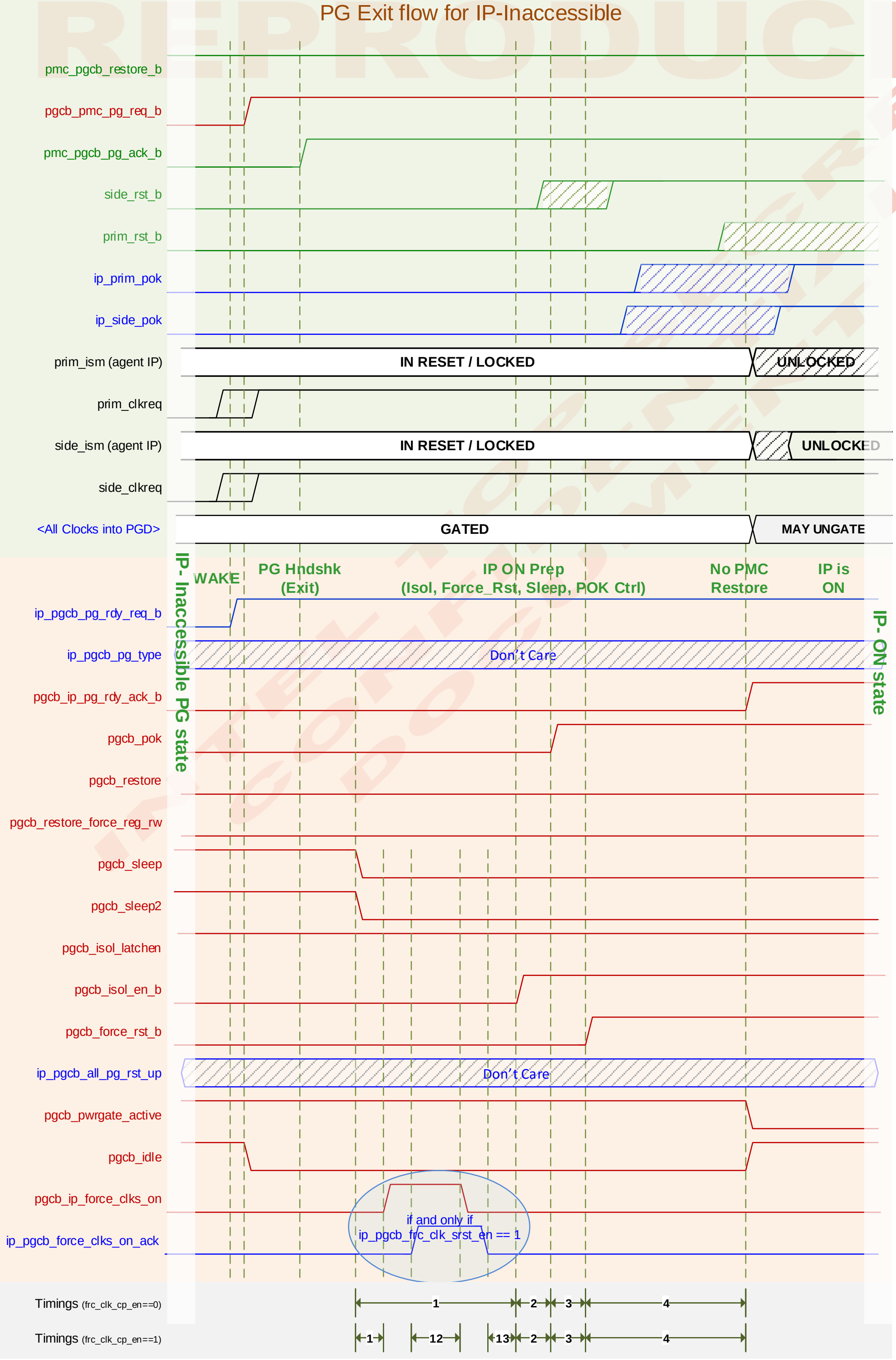
IP-Inaccessible PG Entry



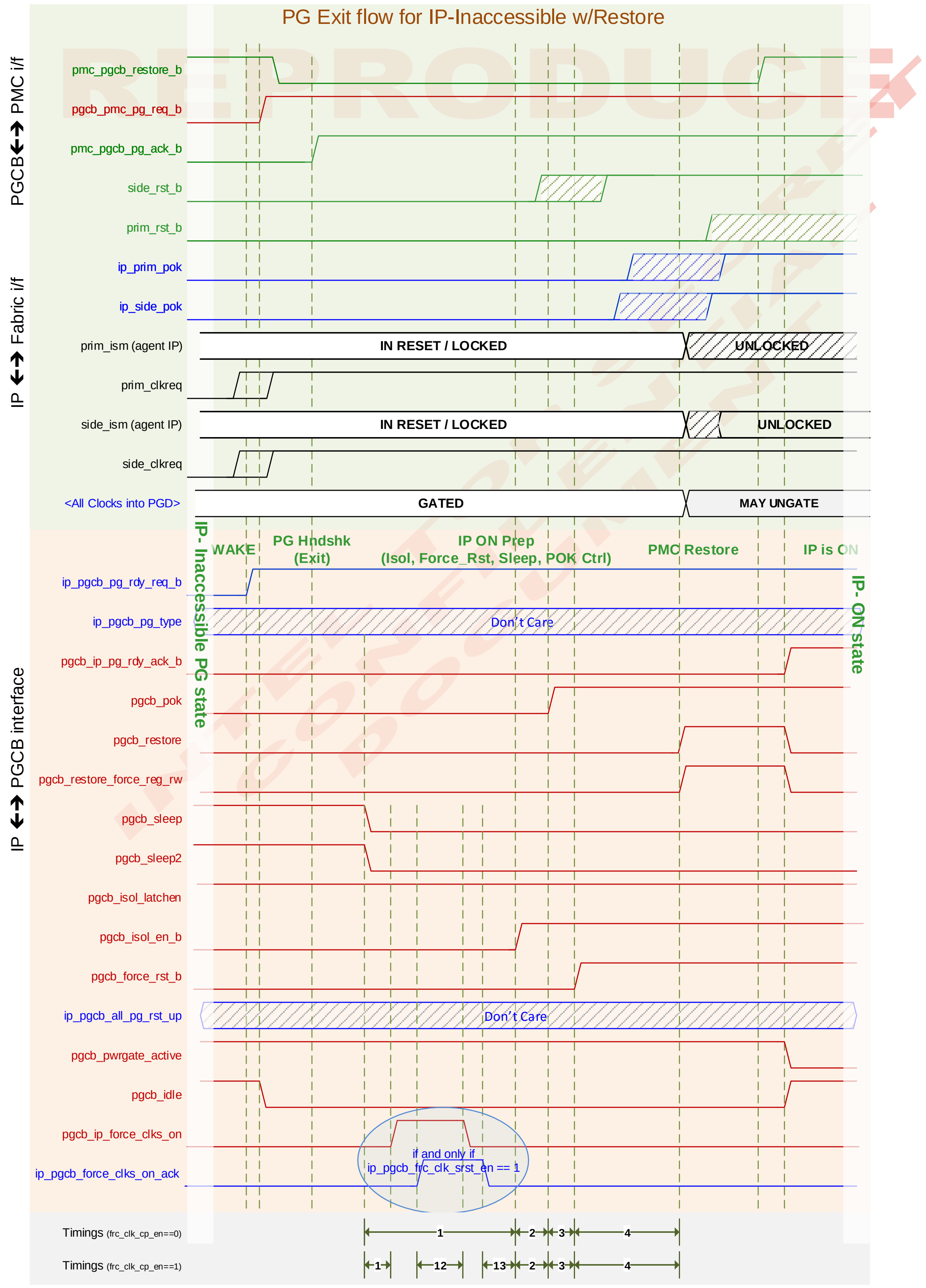
† pg_rdy_req_b and pg_type are allowed to change after they are sampled, although they are shown to be stable here. Please refer to the signal description tables for details on when they are sampled.

IP-Inaccessible PG Exit

**Note: This waveform also represents PG Exit flow from cold boot, and PG Exit flow after PGCB is reset cycled (if IP is configured to be power-gated by default), in addition to any other PG exit from IP-Inaccessible state.*

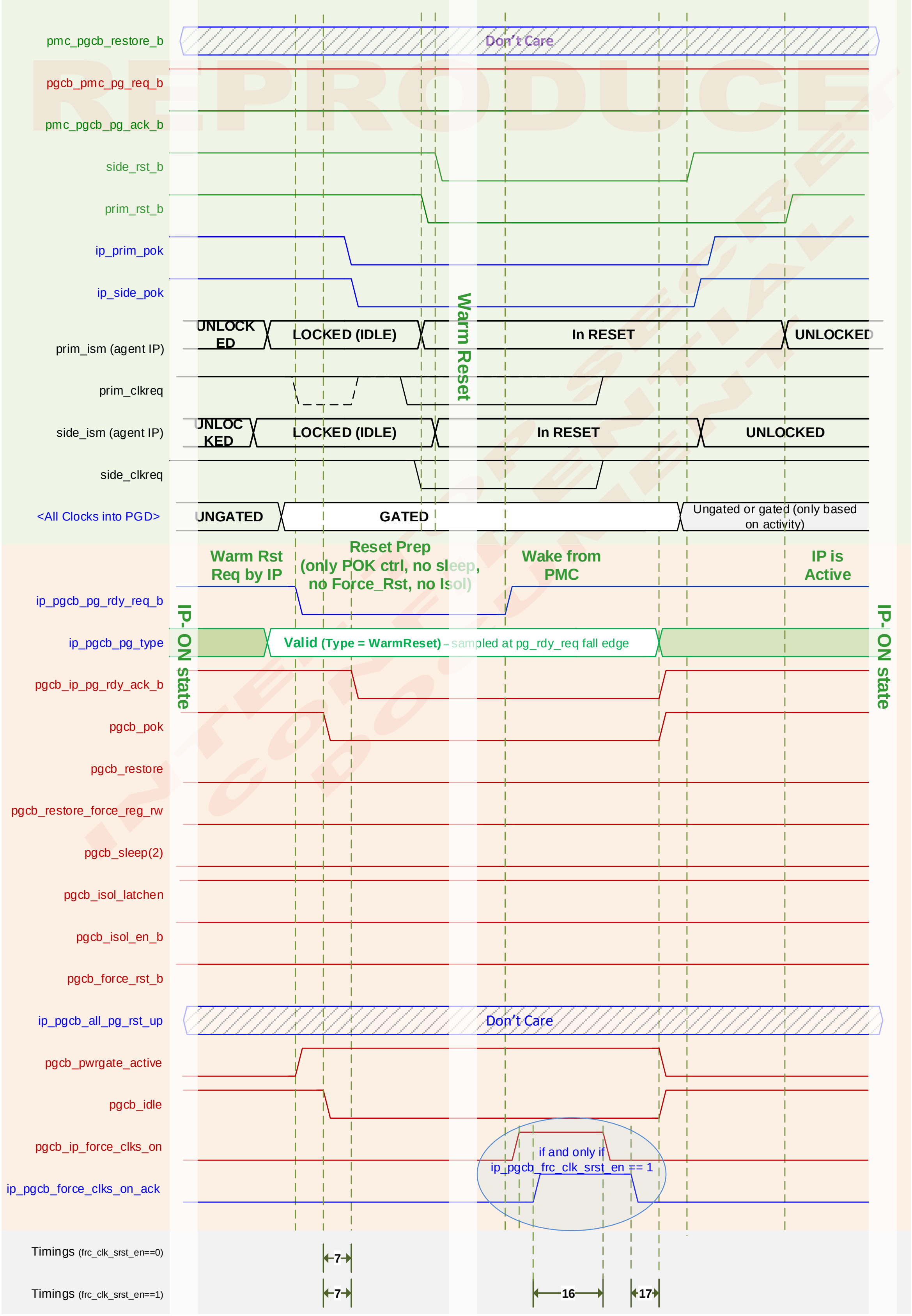


IP-Inaccessible PG Exit with Restore

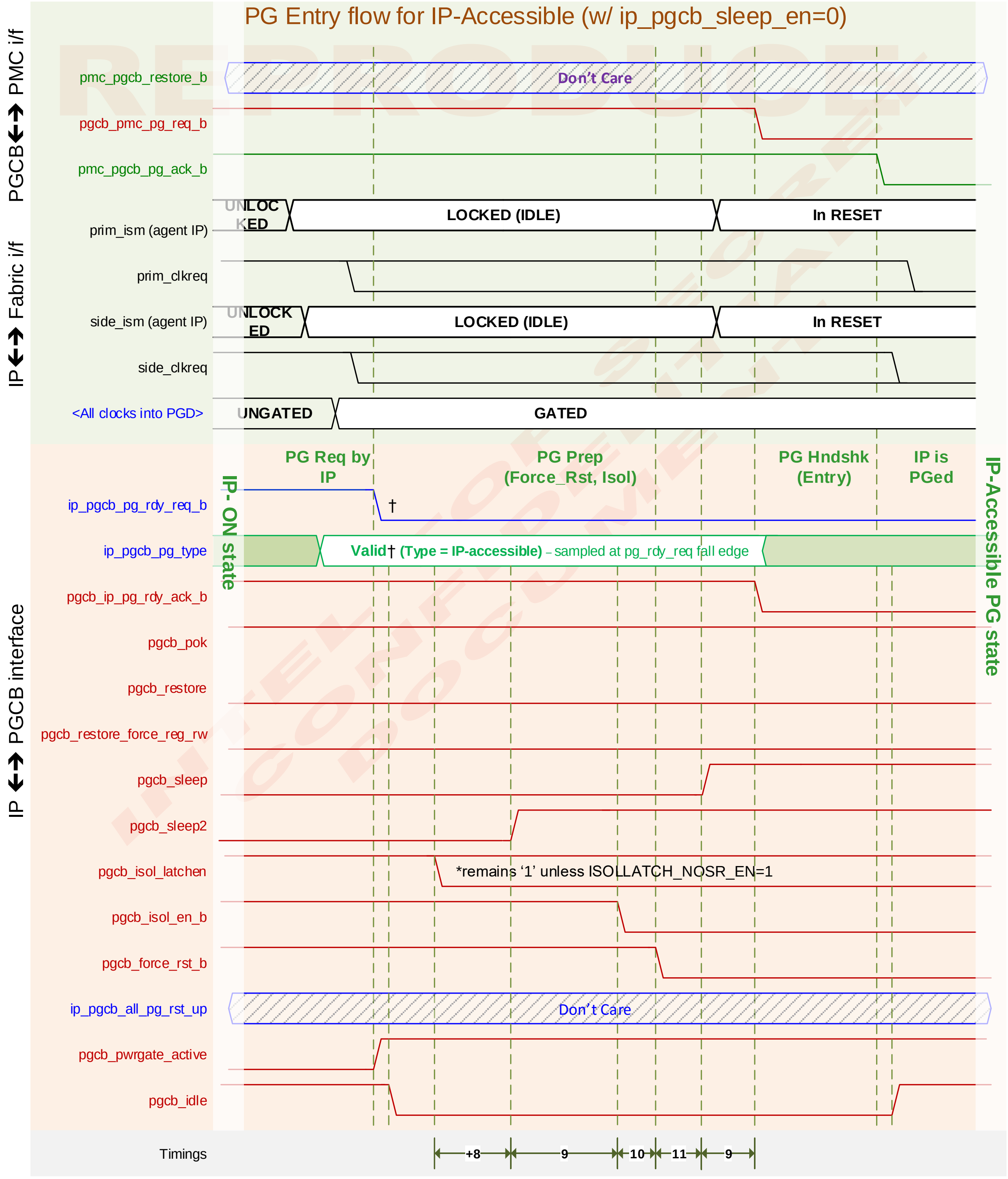


Warm Reset Flow

Entry/exit flow for Warm Reset



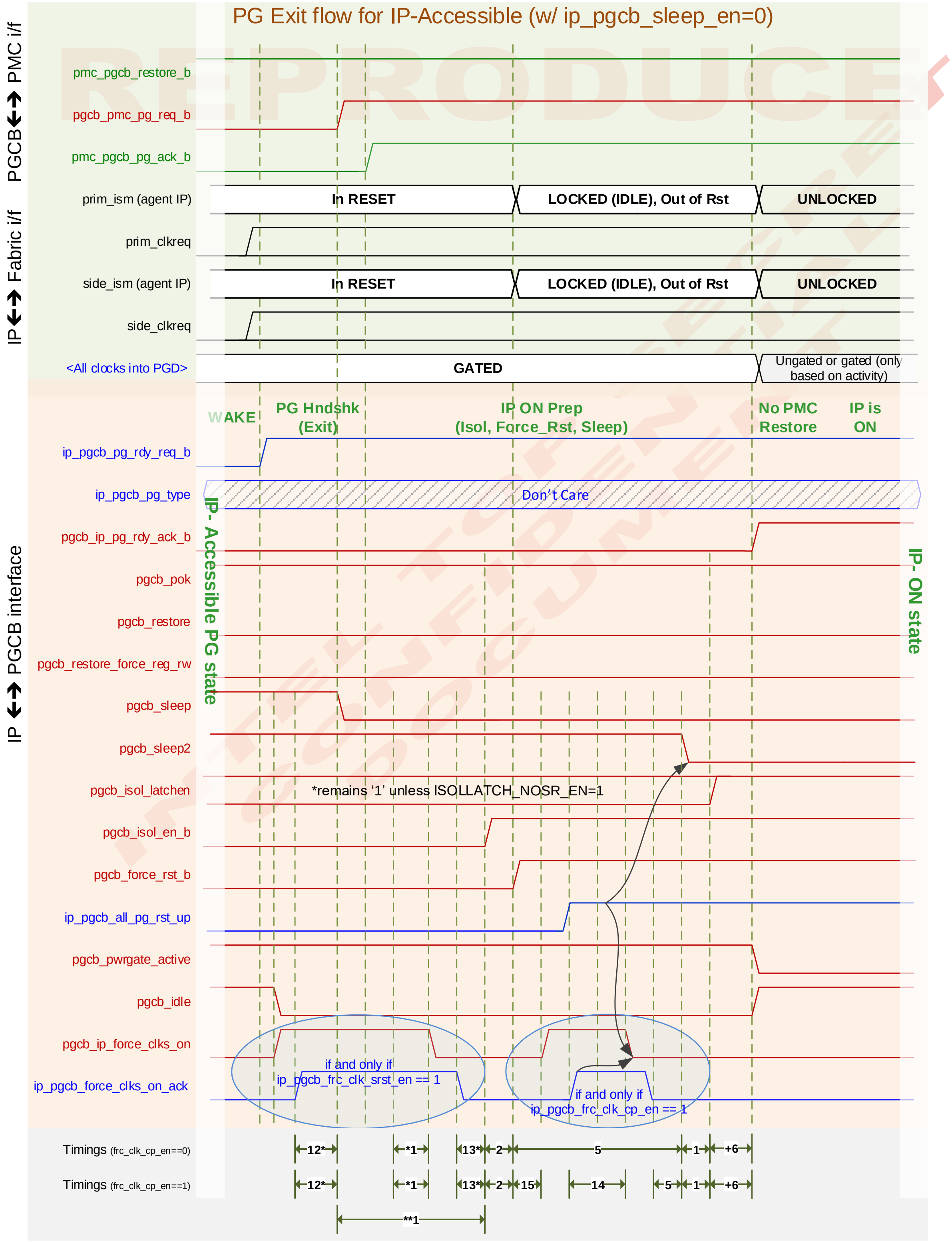
IP-Accessible PG Entry (w/ ip_pgcb_sleep_en==0)



† pg_rdy_req_b and pg_type are allowed to change after they are sampled, although they are shown to be stable here. Please refer to the signal description tables for details on when they are sampled.

+ This timer (8) is only relevent if ISOLLATCH_NOSR_EN==1

IP-Accessible PG Exit (w/ ip_pgcb_sleep_en==0)



*** Note:** Timings marked with (*****) are only applicable if frc_clk_srst_en==1
**** Note:** If frc_clk_srst_en==0, timer 1 will be from sleep (fall) to isol_en_b (rise)

+ This timer (8) is only relevent if ISOLLATCH_NOSR_EN==1,
if ISOLLATCH_NOSR_EN==0, timer 5 will be until pg_rdy_ack_b (rise)

IP Power Gating Requirements

High Level Requirements

The high level functions expected from IP blocks with respect to the PG flows are listed below. Some of these functions may be handled by the Clock Domain Controller (CDC) component (refer to the CDC Integration Guide for more information).

1. **PG trigger:** The power-gating process/flow is initiated by the IP block through a request to the PGCB. It is the responsibility of the IP block to be aware of (and respond appropriately to) various reasons to trigger the PG entry flow – such as based on the pmc_<ip>_sw_pg_req_b signal, or the IP block receiving ForcePwrGatePOK message from PMC, or D0i3 power-gating, or simply HW-autonomous power-gating.

2. **IP prep for PG entry and Exit:** The IP block is required to gate all clocks going to the PGD, lock the agent ISMs for the primary/SB fabrics, and lock external interfaces, before making a request for power-gating entry to the PGCB for a given PGD. This is also referred as getting the PGD (in parts or together as one entity) into a “power-gate ready” state. On exit from power-gating, the IP block is required to put the PGD in a fully “Active” state by ungating clocks to the PGD, unlocking ISMs, and unlocking external boundaries (based on the “pgcb_restore” output from PGCB, if applicable).

    a. **Support for PGD clock gating:** The IP block is required to perform the actual clock gating for all clocks going to the PGD before it requests power-gating entry for that PGD. Upon exit from power-gating, the PGCB provides an indication of when the clocks may be ungated, but it is required for the IP block to handle the ungating of the clocks to the PGD.

    b. **Support for Agent ISM lock/unlock and external interface lock/unlock:** The IP block is required to perform the actual lock/unlock of the agent ISMs for the fabric interfaces within the IP block. Similarly, it is the IP block that performs the locking/unlocking of all external interfaces relevant to the PGD. How the IP block achieves these functions is beyond the scope of this document – assumption is that the IP block comprehends any relevant IOSF spec requirements.

    c. **Support for driving prim_pok/side_pok signals in fabric clock domain:** The IP block is expected to synchronize the “pgcb_pok” signal into fabric clock domains in order to drive the side_pok and prim_pok signals for the IOSF-Sb and IOSF primary interfaces, respectively. Before the IP block may request the PGCB for a PG entry, the IP block is required to comprehend any requirements relevant to these signals from IOSF spec.

3. **Configuration related to PG**: The IP block is responsible for physical storage of configuration information related to PG, as well as for ensuring access to this configuration for firmware/software. For example, some of the timer configuration values may need to be driven from AON/Ungated domain (if these are programmable), while some others may need to be protected with isolation

latches before connecting them to the PGCB (assuming these originate in the PGD). Refer to section "Timer and Configuration Signal Requirements" for details on specific requirements related to the timer values (and some other controls) relevant to the PGCB sequencing for PG flows.

4. **Wake function:** The IP block receives and consolidates various wake indications such as pmc_<ip>_wake signal, wakes from fabric ISMs transitioning out of idle, etc.

5. **Restore phase related requirements:** There are at least two specific IP requirements pertaining to support of restore phase.

    a. **Support for Restore phase read/writes to all config registers (independent of attributes):** Some IP blocks support Hw-set/SW-clear (or vice versa) and other register attribute types (different from regular RW attribute) for configuration registers within the IP block. During the restore phase, PMC may need to read/write to some of these registers in order to place them in the appropriate state. Therefore, the IP block needs to have these registers behave the same as regular RW registers during the restore phase. The PGCB provides a separate signal to indicate when this function is desired. NOTE: At this time, it is expected that **only some** IP blocks are impacted by this restore phase requirement. IP teams should contact their architects to comprehend if this requirement is relevant to them.

    b. **Support for Boundary Lock during Restore phase of PG exit flow:** During the restore phase, the IP block is required to ungate the clocks in the PGD, unlock all IOSF interface agent ISMs, but keep all other (external) boundaries locked. Refer to CDC Integration Guide for more details on boundary lock function.

NOTE: For comprehending if an IP block needs to send an IP_Ready message to PMC within a given SOC or not, the IP block teams need to contact the relevant SOC or otherwise refer to Chassis Reset Arch HAS. The PGCB does not require or mandate any specific behavior from the IP block in this regard.

Details of IP Block Requirements:

IP must handle locking/unlocking IOSF ISMs

The current versions of the ISM shared components (IOSF-SB base endpoint, COMLIB IOSF primary ISM module, IOSF IU primary interface component) provide support for locking/unlocking through specific signals. IP blocks need to be aware of these features and use them to achieve ISM locking/unlocking. The CDC integration guide has further details on the method for ISM locking.

Respond Promptly to an IP-Inaccessible Request

The IP is required to respond as quickly as possible to the ForcePwrGatePOK message requesting an IP-Inaccessible PG entry and should not wait for hysteresis timers that may typically be required to expire before IP block initiates entry into a PG state. Refer to the Chassis PG HAS for details. In most cases, if there were any existing transactions that were important to complete, the IP would have already had an opportunity to complete them based on a preceding handshake with the PMC. Refer to the Chassis Reset Arch HAS for details.

Decode and Track ForcePwrGatePOK Message

The IP needs to decode and keep track of the ForcePwrGatePOK message (both versions of the message with PG+POK, as well asPOK only) until it has entered the IP-Inaccessible state (ForcePwrGatePOK with PG+POK) or the Warm Reset state (based on receiving ForcePwrGatePOK with POK only) as applicable. It is possible that the IP receives the ForcePwrGatePOK message while it is initiating an IP-Accessible PG entry. If this occurs, then the IP needs to wake up from the IP-Accessible state and issue a new PG request with the appropriate PG type encoding to the PGCB.

Be aware of IP-Accessible flow, IP-Inaccessible flow and Warm Reset flow

The IP must be aware of what type of power gating it is requesting (IP-Accessible, IP-Inaccessible, Warm Reset, etc.) and indicate this to the PGCB through the “ip_pgcb_pg_type[1:0]” signal.

Perform Fuse Pull (if/when needed)

When coming out of IP-Inaccessible PG state, the IP block should initiate a fuse pull (if it needs any fuses or soft straps) after either “pgcb_restore” is asserted or “pgcb_ip_pgrdy_ack_b” is de-asserted. The PGCB does not support any reset sequencing for fuse/soft strap pulls.

Fuse/Strap Retention

For an IP block that supports state retention (ip_pgcb_sleep_en==1), it is required that all fuses/soft straps needed by that IP block are either stored in retention cells or within the AON/Ungated domain of the SIP. Therefore the IP block must not perform a fuse/soft strap pull when exiting an IP-Accessible PG state. All IP blocks are allowed to perform fuse/soft strap pulls (if fuses/soft straps are required) only on exit from IP-Inaccessible PG state.

- If the IP block were to try to pull fuses/soft straps when coming out of power gating with state retention, pgcb_sleep would still be asserted when the fuse pull is happening and this may cause undesirable behavior.

Monitor Gate/Wake Conditions

The logic that makes up the ip_pgcb_pg_rdy_req_b indication should include the following:

- Configuration bits as defined in the Chassis PG HAS
- IOSF Fabrics ISM state
- Any other relevant idle/wake terms from the IP/PMC

Please refer to the Chassis PG HAS for details on the priority and validity of these signals.

PGCB vs IP signals from Chassis PG HAS

These are the signals from the Chassis PG HAS that the PGCB will comprehend directly:

- pgcb_pmc_pg_req_b
- pmc_pgcb_pg_ack_b
- pmc_pgcb_restore_b
- sleep_en (ip_pgcb_sleep_en may be connected directly to the sleep_en register defined in the Chassis PG HAS if the IP does not have any IP-specific reason to override it)

All other signals defined in the Chassis PG HAS must be comprehended by the IP.

Reset Propagation Indication to the PGCB

State retention cells have a requirement with respect to sequencing of the reset and sleep control during the PG Exit flow. It is required that on an IP-Accessible PG exit, reset to the state retention cells be de-asserted before the sleep control signal is de-asserted. The retained value of the SR cell may be lost otherwise. The PGCB defines an input signal “all_pg_rst_up” which is required to be asserted by the IP block (after the PGCB un-forces the resets to the PGD) to indicate that reset de-assertion has completely propagated inside the IP block PGD logic. This signal must be asserted only after the resets have propagated to the reset pin of the state retention cells.

Clock Gating Requirements for State Retention

Power Gating flows comprehend the usage of State Retention cells, Reset Synchronizers, etc. Because of these, there is a specific requirement for all clocks in to the PGD to be gated during the PG flow/state (refer to signal description for the PGCB output “pwrgate_active”).

This PGCB is designed to operate with State Retention Cells which require that the sleep control assert/deassert only when the clock to the SR cell is gated low.
Note: An IP block cannot rely on de-assertion of the clkack (signal that is part of the clkreq/clkack protocol) to imply that the clock is not toggling. It is required to have an internal logical clock gate for each clock consumed by the IP in the PGD.

Note that during IP-Inaccessible PG entry, it is not necessarily required that the clock be gated to the state-retention cell when sleep is asserting as any state that might be retained will be cleared out on the exit from PG. This scenario is possible if a CDC is using a really slow clock compared to the PGCB clock, and it is acceptable.

Deassertion of clkreqs for all clocks used only in the PGD

From a power savings perspective, every IP block is required to de-assert the clkreq signals corresponding to clocks used only in the power-gated domain, when the PGD is power-gated. An IP block may choose to add some hysteresis delay before the clkreq signal is de-asserted (after the PGD has been power-gated). However, in normal functional mode, it is not allowed for an IP block to keep clkreq signal asserted indefinitely for a clock used only in PGD when that PGD is power-gated. The CDC component (provided along with the PGCB release) supports de-assertion of clkreqs for clocks used only in the PGD.

Note: The IOSF spec comprehends that clkreqs can be deasserted due to a power gated state even while in reset. (refer sections titled “Clock and Reset Rules” within the IOSF spec v1.1). Therefore, it is required (from power saving perspective) that even primary and sideband clkreqs should be de-asserted when power gated, if the corresponding clocks are only used in the PGD.

For IP blocks that do not use a CDC for clock/clkreq control, please refer to Recommendations Section for guidelines to keep clkreqs de-asserted during a power-gated state.

Isolation Latch Requirements

Refer to the Requirements heading under the Latch Based Isolation section.

Placement of IOSF POK, ISM Lock, External Interface/s Lock Control Logic

Logic driving the IOSF POKs, controlling the locking of the IOSF ISMs (although not the ISMs themselves), and controlling the locking of external interfaces, should be in the ungated domain of the IP (not the PGD). This is required so that the lock control signal/s and POKs can be released even if the PGD is still in reset.

Placement of PG Wake and Gate Logic

Logic responsible for driving the ip_pgcb_pg_rdy_req_b signal should be in the ungated domain of the IP block to be able to respond appropriately to wake and gate conditions.

At least some of this logic must be in the pgcb_rst_b domain (logic for some IP-specific wakes/gate conditions may be okay to reside in shallower reset domain).

Handle Any IP-Specific Sequencing Requirements for Reset Assertion during PG flow

The PGCB provides only a single signal for forcing reset assertion/de-assertion, "pgcb_force_rst_b". If the IP block has any sequencing requirement/s for resets during IP-Accessible PG entry/exit flows and IP-Inaccessible PG Entry flow, the IP is expected to handle this requirement/s itself. Fuse pull related reset sequencing requirements for IP-Inaccessible PG Exit flows are handled by PMC (refer to Chassis Reset Arch HAS for details).

Gating/Ungating the PGCB Clock

The IP is responsible for ensuring the PGCB clock is ungated whenever it changes the state of the "pg_rdy_req_b" signal and the PGCB clock should remain ungated until the next assertion of "pgcb_idle".

If the IP desires to gate the PGCB clock, it may do so only when the "pgcb_idle" signal is asserted and there is no request to change the PG state of the IP.

IP Requirements for Restore Phase

The PGCB provides a dedicated signal "pgcb_restore" to indicate to the IP block when the restore phase needs to be initiated or ended. During the restore phase, the PMC (or some other platform agent) intends to write to the IP configuration registers to restore the IP state to what it was before power-gated state was entered (refer to Chassis PG HAS for details on restore flow).

The various PG exit flow waveforms provide detailed sequence of signals applicable to the restore phase. The restore phase is required to be started when PGCB asserts the "pgcb_restore" signal. At this point, the IP needs to bring the PGD out of "power-gate ready" state back to active state by ungating clocks to the PGD, and unlocking ISMs (if applicable). However, the IP block is required to keep the boundary locked for other interfaces (refer to CDC Integration Guide for details on boundary lock function). Note that in the normal PG exit flow, the IP block is required keeps its clocks gated until the de-assertion of "pgcb_pwrgate_active". However, when the "pgcb_restore" signal is asserted, the IP block is required to honor that signal at a higher priority than "pgcb_pwrgate_active" and ungate its clocks and unlock its ISMs (but keep the boundary locked for other interfaces).

During the restore phase, the PGCB asserts another signal "pgcb_restore_force_req_rw". This signal is intended to cause all configuration registers to be RW for PMC accesses. Once the PMC completes restore phase accesses, it de-asserts the "pmc_ip_restore_b" signal – on which indication, the PGCB de-asserts the “pgcb_restore” signal to the IP block. The PGCB next de-asserts the “pgcb_pwrgate_active”, this is when the IP must unlock its boundaries.

PMC Requirement for Restore Phase

It is to be noted that as per current Chassis PG HAS definition of the restore flow, there is some delay between the time that the PMC de-asserts the “pmc_pgcb_restore_b” signal to the IP, and the time that the IP block detects that the PGCB has de-asserted the “pgcb_restore” signal. This delay includes the time for the PGCB to synchronize the rising edge of the PMC signal, and then respond to this change (typically 3 cycles of the PGCB clock). Therefore it is required that PMC wait for some time before open up the path for external agents (including CPU) to access the IP blocks after the PMC restore signal has been de-asserted (where the timer value is SOC-specific).

In case PMC opens up the path for other agents in the platform to access the IP blocks immediately after de-assertion of the PMC restore signal (“pmc_pgcb_restore_b”), it is possible that the IP block’s registers may still be in the forced RW mode – and this may present a security issue. Hence the recommendation for at least having a timer-based delay to allow all IP blocks to revert to normal register attributes before PMC opens up the path for external accesses in the platform.

Recommendations

PGCB Clock

The PGCB supports clock gating on the pgcb_clk by providing an idle indication through the “pgcb_idle” signal. PGCB design assumes that the IP logic that is requesting the PGCB to start a PG exit/entry transition is responsible for ungating and/or waking up the pgcb_clk (if it is not running). When the PGCB is done with its processing, it indicates its readiness for clock to be gated/shut off by asserting the idle signal. Please refer to waveforms for various PG entry/exit flows to comprehend the behavior of “pgcb_idle” signal.

Note: Though this is not precluded, there is no common recommended method for async assertion of clkreq based on a wake event in order to request the clock for the PGCB (assuming the PGCB is not using an always running clock).

Note: the PGCB design assumes that the Warm Reset event is short in duration, and therefore, there is no meaningful power impact if the “pgcb_idle” signal is not asserted during this event (pgcb_idle not being asserted prevents the pgcb_clk from being gated during this event). Please refer to Chassis Reset Arch HAS for definition of the Warm Reset event.

Isolating clkreqs

Caution: using isolation to drive clkreqs is not recommended as the behavior of clkreq will be different when running non-UPF simulations as isolation would not be emulated and the clkreq may not deassert. The information below is only being kept for reference and should not be used to implement clkreqs.

~~For IP blocks that use the CDC component, the clkreqs are driven from the AON/Ungated domain. If an IP block is not using the CDC component, there are some considerations related to driving the clkreq from the~~

~~PGD of the IP block. The IP block is allowed to internally drive a clkreq high when in reset, however from an SOC perspective, the clkreq for a clock used exclusively in the PGD should be seen deasserted while the PGD is power gated. In addition, for a PG exit there is need for a mechanism to allow the clkreq to assert before the internal reset is deasserted.~~

~~One possible approach is to use isolation to keep clkreqs (for clocks that are used exclusively in the PGD) de-asserted while power gated.~~

~~As an example:~~
~~For an IP block where prim_clk and side_clk are used exclusively within the PGD, the prim_clkreq and side_clkreq signals from the functional logic in the PGD, could be de-asserted as part of the IP preparation for PG entry (prior to the IP block making a request to the PGCB for PG Entry).~~

~~It is expected that at the boundary of the IP block, the Chassis clkreq/clkack protocol is followed while doing the clkreq deassertion. This ensures that the clkreqs deassert cleanly instead of potentially violating the clkreq protocol if the isolation control forced them low without any previous handshake.~~

~~In the PG entry sequence, the isolation control is asserted before the internal prim/side resets are forced to be asserted, therefore, the externally visible prim/side clkreqs remain low even though the internal clkreq signals may change to an asserted state when internal resets are forced to be asserted.~~

~~In the PG exit sequence, when the PGD has been powered up, the isolation control de-asserts before the resets are de-asserted. The isolation de-assertion may be used to allow the propagation of the internally asserted clkreq signals (because the logic is still in reset) to the externally visible clkreq signals.~~

Reset Propagation Independent of Clocks

It is recommended that IP blocks design the usage of the “force_rst_b” signal from PGCB to propagate the reset de-assertion without any dependence on clocks. This signal is intended to be asserted and de-asserted when all clocks are gated to the PGD, so that should handle concerns related to asynchronous reset changes.

One way to implement such a design would be to insert an AND gate (ctech or similar specially marked cell, which is not touched by synthesis or physical optimizations) subsequent to every reset synchronizer for a given clock domain in the PGD of the IP block.

The default behavior of the PGCB is to assume that the IP block supports the recommended mechanism of clock-independent propagation of reset de-assertion (when the PGCB un-forces the reset signals).

Some IP blocks may require that clocks be turned ON for reset propagation – the PGCB also supports this requirement if the “ip_pgcb_frc_clk_cp_en” input is asserted (please refer to the waveform for details of the sequencing).

Placement of Agent ISMs

The power gating flows do not require the agent ISMs to be part of AON or ungated domain. It is recommended that the IP include the agent ISMs as part of the power gated domain.

Reset domain of PGCB

Refer to the Chassis Reset Arch HAS for guidelines and recommendations.

Latch Based Isolation

NOTE: For SPT/WLV/BXT, it has been decided that isolation latches have to be hand-instantiated/coded instead of being inferred through UPF. Based on that, the rules and recommendations related to how IP blocks should create the isolation latch functionality are described below.

Key features of hand-coded isolation latch scheme:

1. Isolation latches are intended to be used only for IP-Accessible PG and remain transparent in all other cases, including cold boot
2. IP blocks need to instantiate non-resettable latches (for capturing the isolation value) manually within the RTL
3. Isolation gates are inserted through UPF to precede the latches (for capturing isolation value) – polarity of the isolation gate has to be determined based on the individual requirements/reset behavior on each pin.
4. When the isolation latch enable control (driven from PGCB, in pgcb clock domain) is changed, the IP block is already required to have the clocks be gated to all logic in the PGD (as per earlier requirements in this document)
5. Control signal for the isolation gates inserted through UPF is “pgcb_isol_en_b” and is different from the isolation latch enable control signal, “pgcb_isol_latchen”.

Please refer to the section on waveforms for exact timing of these two signals versus other signals.

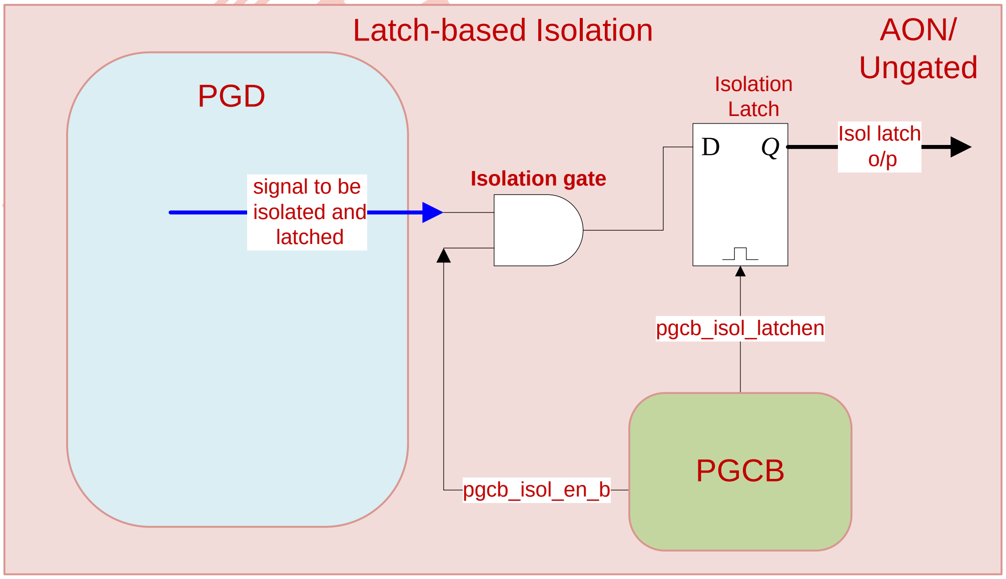


Figure 4: Illustration of Isolation Latch usage

Rules for/around control of isolation latches:

1. Default value of latch enable control is to have the latch open (transparent) – which implies that for cold boot, or if PGCB is reset, the latch simply relays the value held by the isolation gate preceding the latch (and there is no non-determinism for the value on account of the non-resettable latches).

2. Stable D input of isolation latch must be guaranteed by IP block around the times when latch enable control transitions.

    a. This is required to prevent any asynchronous transitions at the destination logic

       IP block requirement: The IP block is **required** to gate the clocks to the logic driving the Isolation latch prior to requesting PG entry request to the PGCB, and keep these clocks gated until the PGCB either de-asserts the "pwrgate_active" signal or it asserts the "pgcb_restore" signal.

       Also refer to section on "Support for Turning on Clocks for PG Exit flows".

    b. A recommendation related to preventing asynchronous transitions when the latch is closed/opened, is for the IP block to ensure that signals having isolation latches are driven directly by SR cells in the PGD (no combinatorial logic or non-SR sequential logic – other than buffers – between the SR cell and isolation gate).

        i. If an IP block cannot meet the above recommendation for some reason, alternate mitigation needs to be in place (such as reviewing the interface to ensure that receiving logic is able to tolerate asynchronous transitions during the relevant time, etc.)

3. Latch enable control transitions (closes/opens) only for IP-Accessible PG (remains open through IP-Inaccessible PG entry/exit, and at all other times, including reset)

    a. Implies that for IP-Inaccessible PG, the IP block must guarantee that enabling/disabling of isolation gates does not cause asynchronous transition at destination/receiving logic during PG exit/entry

    b. In prep for IP-Inaccessible PG, it is recommended that IP blocks synchronously bring all signals going through isolation latches to the reset values (which should be the isolation values as well) prior to making power-gating request to PGCB

        i. NOTE: IP blocks that do not intend to be put into IP-Inaccessible PG state, do not need to support this above recommendation. If an IP block cannot meet the above recommendation for some reason, alternate mitigation needs to be worked out (such as reviewing to ensure that receiving logic can tolerate the asynchronous transitions during the relevant time, etc.)

4. The pgcb_isol_latchen signal should drive the enable pin of the latch directly. It is required that the latch stay open while isol_latchen is asserted so that the data can flow through freely, for this reason clock gating components such as the ctech_gc_latchen should not be used.

Design Note:

If an IP block needs to have the isolation latches remain closed (holding the state of the output) for some time around an IP-Inaccessible entry (so as to prevent the consumers of the output of the isolation latch from being impacted), the IP block could choose to set a flag that keeps the isolation latches closed during IP-Inacc Entry until such time that the external reset to the IP block asserts.

Validation Note:

Isolation Latches will never be PV'ed as the control signal is asynchronous. It is recommended that there be checks in place during simulation to ensure that the input to the isolation gate before the isolation latches never toggles while clocks are gated in the PGD.

Structural Design Note:

Isolation latches show up as unclocked registers, and need to be waived by IP blocks that are using these isolation latches.

Support for Turning on Clocks for PG Exit flows

In general, using the pre-commit based power gating flow (with a combination of CDCs and PGCB), all clocks to PGD logic are fully gated during the PG Entry and PG Exit flows (for IP-Inaccessible as well as IP-Accessible).

However, there are certain design structures that need the clocks to be turned on during specific part/s of the PG exit flow. There are three known types of these design structures:

1. Sequential logic using synchronous resets (for example, flops that are reset using the D input rather than the async reset input)

2. Sequential logic between two or more state machines (where the state machines are using state-retention cells).

3. Contetion clearing on cold boot

For case (1) above, it is required to ensure that the sequential logic using synchronous resets is in the reset state prior to isolation removal and reset de-assertion during the PG exit flow. In order to achieve this, at least one clock edge is required to be seen by this sequential logic. This requirement applies to both IP-Inaccessible and IP-Accessible PG exit flows.

For case (2) above, it is required that sequential logic between any two state machines (where the state machines use state retention cells) be given some clocks after reset de-assertion (but before de-assertion of the state retention control – sleep signal) in order to let the context propagate from output of one state machine to the input of the other state machine. Allowing the context to propagate enables all the sequential logic in the PGD to be in the same state that it was in, prior to the PG entry. It is possible to avoid this requirement by having the intermediate sequential logic also use state retention cells, but that comes with consideration of some additional gate count (state retention cells use more gates, in general). This requirement is applicable only to the IP-Accessible flow (since state retention is relevant only for this flow).

For case (3) above, it is required that the clocks be running in the power-gated domain while the power is ramping. This is to prevent a large current draw due to contention from uninitialized logic leading to competing drivers in the domain. This does not need to be the actual functional clock that is typically used in the domain, instead, an always running clock can be muxed in. The control to run clocks while power is ramping is the same control used for sync reset propagation ("ip_pgcb_frc_clk_srst_cc_en").

The PGCB provides separate inputs (“ip_pgcb_frc_clk_srst_cc_en” and “ip_pgcb_frc_clk_cp_en”) that modify the PGCB sequencing behavior to support these requirements during the PG exit flows. When any of these inputs are set, the PGCB asserts the “pgcb_ip_force_clks_on” signal at relevant times of the PG exit flow and waits for assertion of the corresponding “ip_pgcb_force_clks_on_ack” signal from the IP block. After a configurable timer delay, the PGCB de-asserts the “pgcb_ip_force_clks_on” signal and waits for the de-assertion of the “ack” signal from the IP before proceeding with the remaining sequence for the PG exit flow. Refer to PG exit flow waveforms for sequencing of the “pgcb_ip_force_clks_on” signal.

Default PGCB behavior:

Note that the design structures listed above are not expected to be commonly used, hence the PGCB default behavior is to never assert “pgcb_ip_force_clks_on” signal.

Potential Design Hazard:

For the case that an IP block sets the context propagation related control “ip_pgcb_frc_clk_cp_en” to 1, there are potential design hazards, as described below.

During the PG Exit flow (say, IP-Accessible PG Exit), if clocks are running in an IP block’s PGD while reset is de-asserted, and sleep is asserted, that part of the design which uses state retention cells is frozen inactive (since the sleep control is still asserted). However, other parts of the design are out of reset and have clocks running, so that they are responding to events and other inputs. This behavior may end up moving the PGD logic to a bad and/or unrecoverable state.

Therefore, any IP team choosing to set the control “ip_pgcb_frc_clk_cp_en” to “1”, must thoroughly analyze, review, and validate the design to ensure that no such design hazards impacts the PDG logic. One method that may help ensure this is to keep all state machine sequential logic on state retention cells.

Design recommendation:

For IP blocks that need to turn on this feature, there are some other associated changes that need to be done in the IP logic in order to completely meet the requirement of turning on the clock. These include using the “pgcb_ip_force_clks_on” signal to enable clocks to toggle within the PGD of the IP block.

It is recommended that a separate clock, typically the pgcb_clk, be used when the clocks need to be turned on within the PG exit sequence. This clock needs to be routed on to the functional clock tree going into the PGD. Normally, this clock path is gated. Only when the PGCB asserts the “pgcb_ip_force_clks_on” signal (at which time it is guaranteed that the functional clock is gated), this clock is ungated and allowed to propagate on to the functional clock tree. Since the PGCB clock is available and toggling during the PG exit sequence, this is strongly recommended to be used as the alternate clock for routing on to the clock tree. See figure below for the general idea of this approach. Note, that in normal functional usage, when the IP is not power-gated, the alternate clock branch being used to route to the functional clock tree, remains gated.

There are both PV (SD related) and Scan issues to consider for this proposal of routing an alternate clock on to the functional path. First the PV concerns:

1. The alternate clock needs to be of equal or lower frequency than the functional clock.

2. The SD team needs to be informed about the alternate clock to be used, so that in case the uncertainty of the alternate clock is greater than that of the functional clock either of the following two options may be used:- run the PV STA with the larger uncertainty value OR check that the clock period of the alternate clock (reduced by twice its uncertainty) is still larger than the clock period of the functional clock.

3. In case the alternate clock is of higher frequency than the functional clock, it is required that the IP block add some logic to throttle down the clock (for example by masking off some edges of the clock to reduce its effective frequency) such that the alternate clock frequency meets the requirement in previous item.

4. Only if the above requirements are met, then this alternate clock path should be declared as a false path..

5. O therwise, the IP block should ensure that the design is PV'ed for STA considering both clocks (functional clock as well as alternate clock).

6. Note: Even if an IP block meets the above requirements, it is still required for the IP block team to obtain final approval from the relevant experts of the scan/DFx/backend/structural design from the SOC where the IP block is to be used.

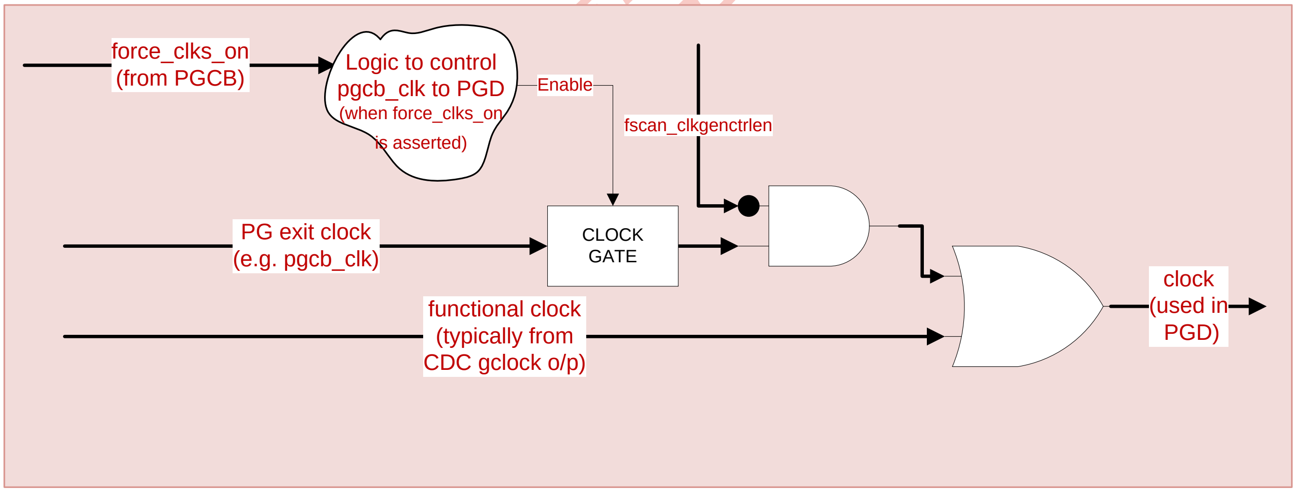


Figure 5: Recommended option for using alternate clock path to run clocks during PG exit flows

To address Scan related concerns, an additional AND gate is required to be inserted at the output of the clock gating cell for the alternate clock. The other input of the AND gate is driven by a SCAN control signal – which when asserted, disables the alternate path – ensuring only the functional clock is guaranteed to use clock tree into the PGD. Note that this logic uses the post-SCC (post-Scan Clock Controller) version of the clocks. The output of the OR gate is also directly used in the PGD logic (i.e. it is a post-SCC clock). No additional SCC components are required with this approach.

Additionally, please note that special “Ctech”-like components must be used for the alternate clock path (including AND and OR gate in above figure), and SD team may have specific naming requirements for these components. IP teams need to consult relevant SD/Scan teams for the SOC for this information.

Note that if instead of following the above approach of routing another available clock (typically, pgcb_clk), the original clock is required to be turned on, there may be power-unfriendly behavior exhibited by the IP through the clkreq. For example, consider the following case: an IP block’s IOSF-SB interface may need to process a wake, so the IP block’s PGD exits PG state. However, as part of the exit flow, clock needs to be forced ON for the IOSF primary logic of that IP block. If the original clock is used for the “force clocks” function, this implies that the clkreq be asserted for the IOSF primary clock in this case. This may require the PLL driving that clock to wake up – even though this IOSF primary logic has no transaction to process otherwise. Bringing up the PLL in this case (when it is not required for any functional reason) is considered power-unfriendly behavior. This is the motivation for the recommendation of using an alternate clock (pgcb_clk) for this feature.

Warm Reset Support

The PGCB supports the Warm reset flow – where the key requirements are that the "*_pok" signals at the IOSF interfaces of the IP block need to be de-asserted prior to the Warm reset assertion, but the IP block PGD must not enter a PG state (refer to Chassis Reset Arch HAS for more details).

The PGCB input "ip_pgcb_pgtype" has an encoding corresponding to Warm Reset– when the IP block asserts the "ip_pgcb_pgrdy_req" signal while the "*_pgtype" signal has the Warm Reset encoding, the PGCB follows the sequence as illustrated in the "Warm Reset flow" waveform. Note that the "*_pgreq_b" signal to PMC is never asserted as part of this flow – and remains de-asserted for the entire flow.

The PGCB supports turning on the clocks for synchronous reset requirements during the Warm Reset Flow. Refer to the "Warm Reset Flow" waveform for the timing.

Reset Isolation Design Hazard:

For an IP block that is subject to a Warm Reset flow and has logic in a reset domain that is not put into reset during this flow, there are potential issues around reset isolation of the deeper reset domain. Please refer to Chassis Reset Architecture Specification for more details of the Warm Reset flow.

The following cases are worth mentioning:

a. If this logic (on the reset domain that remains out of reset during Warm Reset) is all within the power-gated domain (PGD) of the IP block -> therefore, because all functional clocks to the PGD are gated during this flow, there may be reduced exposure to reset isolation violations.

b. However certain other cases, such as when a protocol handshake happens across reset domains still need to be considered (an inactive clock does not solve that problem).

c. Further, if the IP block uses the "Force Clocks ON" feature of the PGCB to propagate synchronous resets within the PGD, then it must be noted that because resets propagate when the clock is active, there may be increased exposure to reset isolation violations.

d. For the case where the logic (on a deeper reset domain) is outside the PGD of the IP block – since isolation is not triggered by the PGCB for the Warm Reset flow, therefore the IP block has to analyze and solve any related reset isolation concerns.

In general, the PGCB does not attempt or claim to prevent or solve any reset isolation violations within the reset domains of the IP block. It is the responsibility of the IP block to analyze all such cases and provide solutions for those.

Power Ungated Default State

Overview

In general, IPs should default to a Power Gated state out of reset. If an IP needs to be Powered On out of reset, the IP owners must get buy-in from PMC and the FC integration team, as they all must be in sync on the default state of the IP. The default assumption is that all IPs are Power Gated out of reset.

If an IP is required to be Powered On out of reset, the PGCB provides a compile time parameter (DEF_PWRON) to change the default state. If this parameter is to be exposed at the top level, the IP must ensure that any supporting logic also comprehends it and can support either option.

The following sections cover additional IP requirements and the Reset States of the PGCB interface when defaulting to Powered On.

If the DEF_PWRON parameter is set, the PGCB's reset state is the Restore state (refer to PGCB IP-InAccessible PG Exit flow waveforms and PGCB FSM state diagram in Appendix).

The reset states of the interface signals when DEF_PWRON==1 are shown in the section titled "Reset States (DEF_PWRON==1)"

IP Requirements

Ensure other IP logic also supports defaulting to Powered On

The interface to the PGCB must match the expected Reset State listed in the following table. Any other supporting logic (such as logic driving the IOSF POKs) must also comprehend that the IP is powered-on by default.

IP may only request Power Gating once it is fully out of reset

The pg_rdy_req_b signal should only assert once the IP's resets have all deasserted. There should be a mechanism in the IP block to ensure that on powering up, the IP block waits to be fully out of reset before it can make the first request for power gating.

PGCB Reset

If the IP is defaulting to Powered On, pgcb_rst_b must be deeper than powergood reset such that the IP is not powered up on a partition reset. Refer to Chassis Reset Arch for further guidelines.

Waivers around usage of PGCB output signals

Some of the PGCB output signals may need to be associated with waivers applicable for timing analysis, clock-crossing checks, etc. It is the complete responsibility of the IP block to review its usage of the PGCB output signals and provide any applicable waivers.

Most of the PGCB signals operate on IP logic in PGD (or ungated domain) in an asynchronous manner. It is to be noted that all clocks to the PGD are required to be gated before the PGCB is requested to start a PG entry flow, and clocks remain gated until the PG exit flow is complete. Therefore, in many cases, using these signals in an asynchronous manner should not cause a functional issue. But the IP has the responsibility to carefully review the impact of this behavior and certify that no functional issues exist. The IP block would need to then add waivers (if applicable) corresponding to the relevant PGCB output signals. All the PGCB signals listed below change only when the signal “pgcb_pwrgate_active” is asserted – which in turn implies that clocks to the IP PGD are gated when these signals change. This is a necessary (though not sufficient) reason for providing waivers for the following signals.

Isolation control

The “pgcb_isol_en_b” signal and “pgcb_isol_latchen” signals may need some waivers for static timing analysis and/or clock-crossing checks.

An IP block needs to satisfy the requirements outlined in the section “Latch Based Isolation” in order to use the “pgcb_isol_latchen” signal.

Force_Reset control

The “pgcb_force_rst_b” signal may need some waiver for static timing analysis and/or clock-crossing checks.

As this signal will be combined with the functional reset, “Logic on Reset” violations may be flagged by the Caliber tool, which can be waived.

Sleep control

The “pgcb_sleep” signal may need waivers for static timing analysis and/or clock-crossing checks.

DFX requirements related to Power-gating

Motivation

- For non-functional test modes such as Scan and BIST testing, DFx needs control over the structurally-relevant portions of the power gating control logic. The objective in these test modes is to either force the power gated domain to be on or off. Since the PGCB (Power Gating Control Block) within each Soft IP that performs power gating directly controls power-gated state of the power-gated domain, it is necessary to define a method for DFx to take control of those signals.

Note: the DFx implementation in the PGCB is not suitable for most functional debug (other than SCAN DUMP) as forcing the domain on or off using the sequencer does not put the IP in a clean and functional on or off state. Rather it only forces the structural control signals such as the PFET and resets.

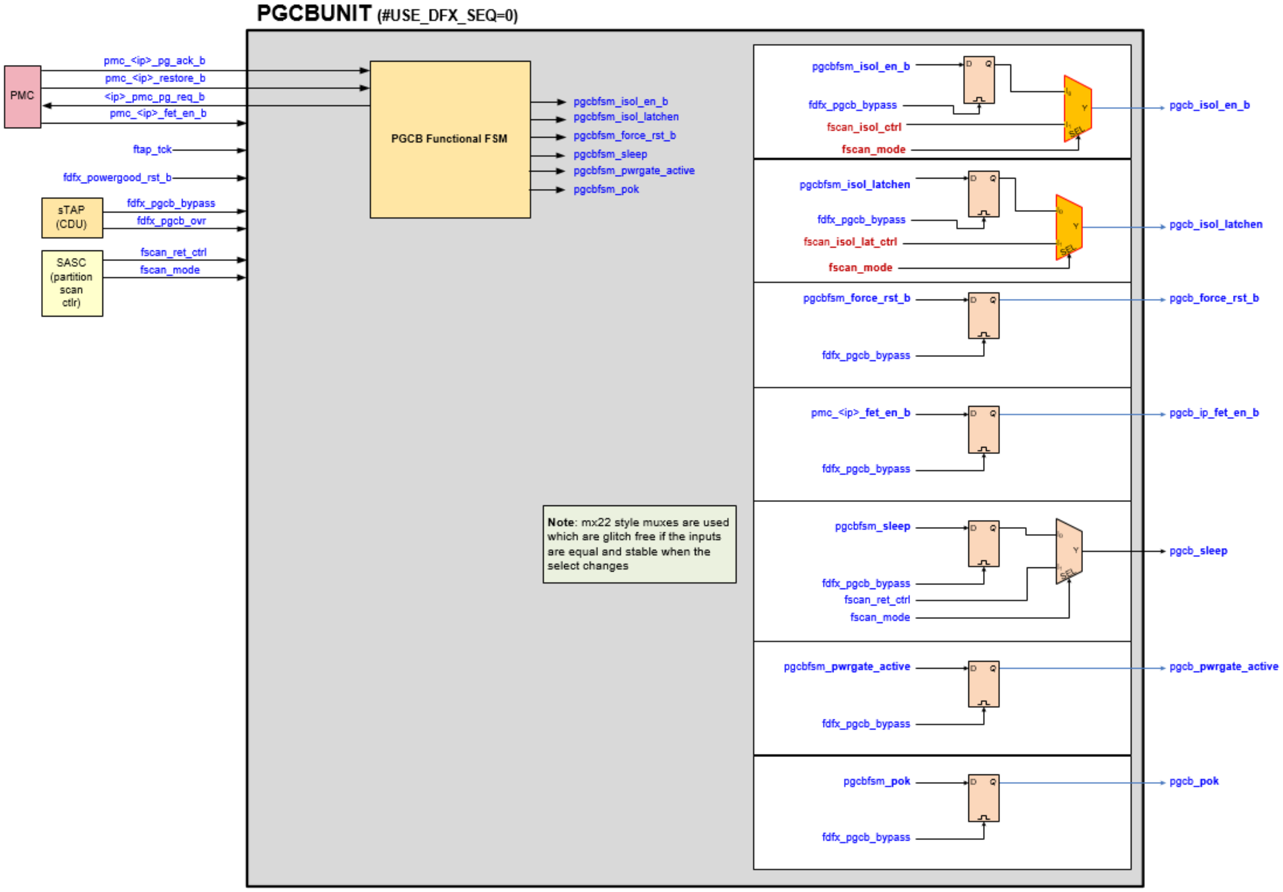
DFx requirements for Non-functional test modes

Summary of PGCB and power-gating related signals that DFx needs to control in a non-functional test mode:

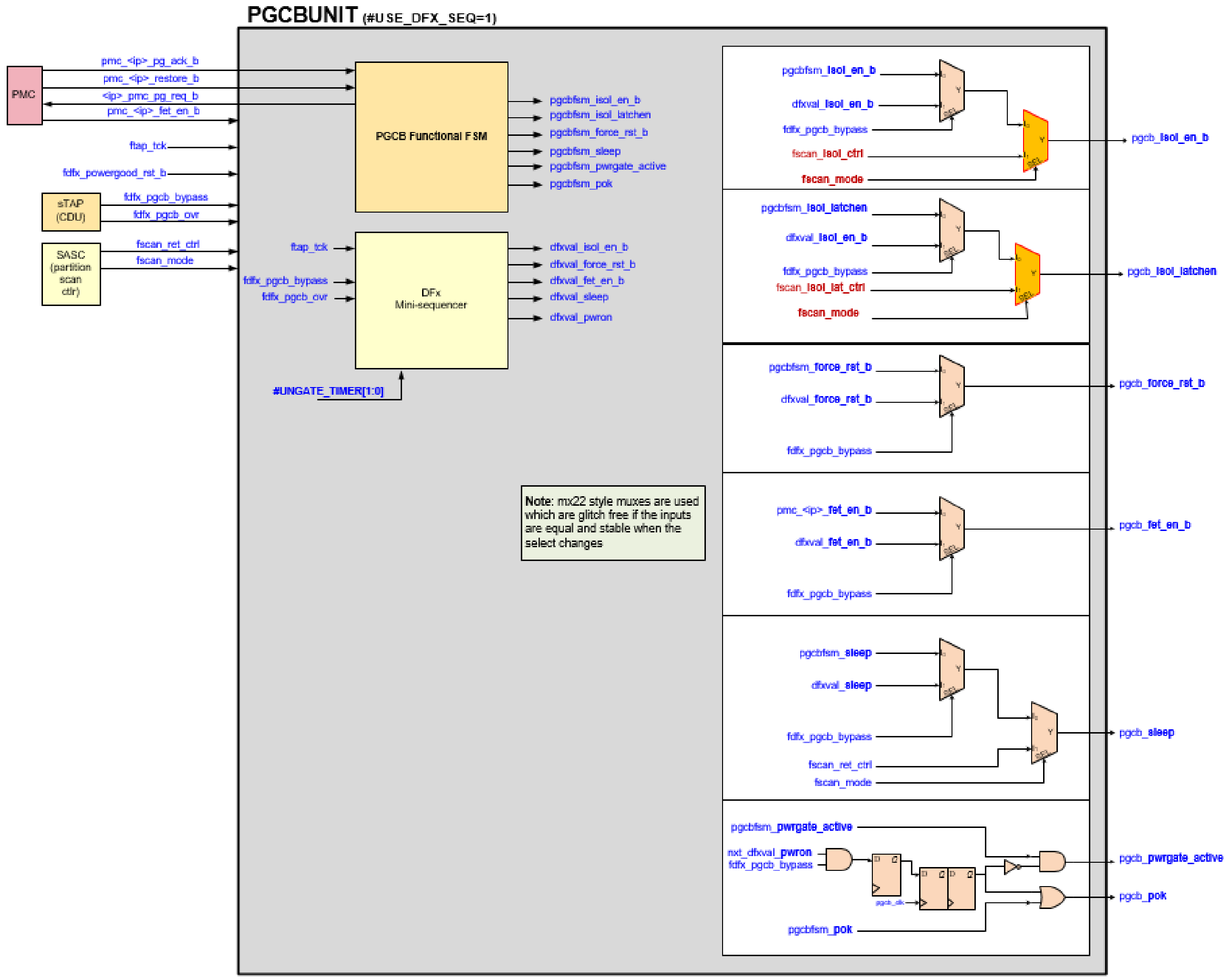
- pgcb_ip_fet_en_b: PFET enable (to control the actual power)
- pgcb_isol_en_b: isolation control signal (to control whether signals flow naturally from the PGD or are isolated)
- pgcb_force_rst_b: reset forcing control signal (to control whether the resets going into the PGD are being forced asserted for power gating or de-asserted for power ungating)
- pgcb_sleep: state retention control signal (SLEEP)
    - In most of the test modes, DFx needs to ensure that the flops are in normal functional mode rather than retention mode
    - In some test modes (i.e. SCAN retention testing), DFx needs to toggle SLEEP to test the flop in both modes

Summary of the approach:

- PGCB will implement two methods for controlling the signals listed above:
    - Method #1 – mini sequencer for DFx
        - When triggered by an incoming DFx signal, a short sequence is invoked that toggles the signals listed above in the correct order to power up or power down the PGD
        - Note that this is **not** the full sequence which is executed by the functional logic in the PGCB, but rather a subset of the steps that targets only the relevant signals for DFx
    - Method #2 – latch current state and switch from functional to DFx mode
        - SoC starts in normal functional mode, perhaps with the PMC telling the PGCB to power up or down
        - When triggered by an incoming DFx signal, all of the signals listed above are effectively "latched" (i.e. held at their current state) and are disconnected from any further changes driven by the functional logic in the PGCB
- A compile-time parameter (#USE_DFX_SEQ) on the PGCB will select between Methods 1 and 2
    - Note: Sunrisepoint must set USE_DFX_SEQ to '1'.



Note: mx22 style muxes are used which are glitch free if the inputs are equal and stable when the select changes



Note: mx22 style muxes are used which are glitch free if the inputs are equal and stable when the select changes

Assumptions

1. Initial state of SoC before any of the power gating override flows described below are invoked:
    a. all external power rails and their associated power good signals are high
2. Usage model for HVM would only override the power gated domains to be powered up
3. However, the HW defined by this ECN does not assume that, and other usage models such as Power Validation may override the domains to be either up or down
4. ftap_tck is continuously running throughout the power gating override flows described below. As PGCB does not come with its' own slave TAP, the ftap_tck would be tapped from the regional ftap_tck, which would house the required DFx control.
5. As the mini-DFx sequencer needs to be in a predetermined default state (as it has no knowledge on the IP's current power state), the sequencer's output is designated to output values that are associated with a powered-on state for all control signals. This is true for both cases whether the IP is in a powered off state or powered on state at the time when the override (fdfx_pgcb_bypass) is applied.
    a. If a given DFx scenario requires the power-up sequencing of reset, isolation, etc (rather than overriding all outputs at the same time), it must move the DFx sequencer to a powered off state by driving fdfx_pgcb_ovr to '1', wait at least 3 clocks, drive fdfx_pgcb_bypass to '1' to

take control of the outputs, and then drive fdfx_pgcb_ovr to '0' to sequence the IP to an "ON" state.

6. Assume that even if the control signals glitch when the DFx overrides are applied, it is acceptable for DFx usage given that adequate settle time will be provided before the actual test starts. Also note that the logic does not remain functional in these test modes, so glitches are not a concern from a functionality standpoint.

<u>Method #1 Definition</u>

<u>i) Flow – DFx Force Power On, IP Defaults to Powered On:</u>

1. In this flow, since the IP defaults to powered on the relevant signals out of the PGCB are in the states associated with being powered up (i.e. not power gated)
2. When fdfx_pgcb_bypass asserts, all signals immediately transition to the DFx version, which defaults to the powered on state for all control signals.
3. At this point, the PGD is fully powered up and ready to be used by DFx

<u>ii) Flow – DFx Force Power On, IP Defaults to Powered Off:</u>

1. In this flow, since the IP defaults to powered off the relevant signals out of the PGCB are in the states associated with being powered off (i.e. power gated)
2. When fdfx_pgcb_bypass asserts, all signals immediately transition to the DFx version, which defaults to the powered on state for all control signals.
    a. If sequencing of isolation, resets and sleep is required, the FSM should be moved to the OFF state by driving fdfx_pgcb_ovr to '1', waiting at least 3 clocks for the FSM to transition, then drive fdfx_pgcb_bypass to '1' (to take control of the outputs) and subsequently drive fdfx_pgcb_ovr to '0' to walk the FSM back to ON.
        i. The DFx sequencer will first turn on the PFET and then delays the deassertion of the state retention control signal ("sleep") by a sufficient amount of time to allow the PGD to power up.
            1. The value of the delay timer is configured by the SoC via a compile-time parameter (#UNGATE_TIMER)
        ii. One ftap_tck clock later the DFx sequencer then deasserts the isolation control signals and the force reset signals

<u>iii) Flow – DFx Force Power Off, IP Defaults to Powered On</u>

1. In this flow, the IP starts in the powered on state and the DFx desire is to force power off. Therefore, all signals out of the PGCB are in the states associated with being powered up (i.e. not power gated)
2. When fdfx_pgcb_bypass asserts, all signals immediately transition to the DFx version, which defaults to the powered ON state for all control signals.
3. If fdfx_pgcb_ovr is subsequently (or in parallel with step 2) driven to '1', the DFx sequencer asserts isolation and resets, asserts sleep 1 clock later, and on the next clock turns off the PFET.

<u>iv) Flow – DFx Force Power Off, IP Defaults to Powered Off</u>

1. In this flow, the IP starts in the powered off state and the DFx desire is to force power off. Therefore, all signals out of the PGCB are in the states associated with being powered down (i.e. power gated)
2. fdfx_pgcb_ovr should be driven to '1' first to bring the DFx sequencer to an OFF state (as it defaults to ON). After 3 clocks the DFx outputs will be in an OFF state.
3. When fdfx_pgcb_bypass asserts, all signals immediately transition to the DFx version, which are now drivn to a powered off state.

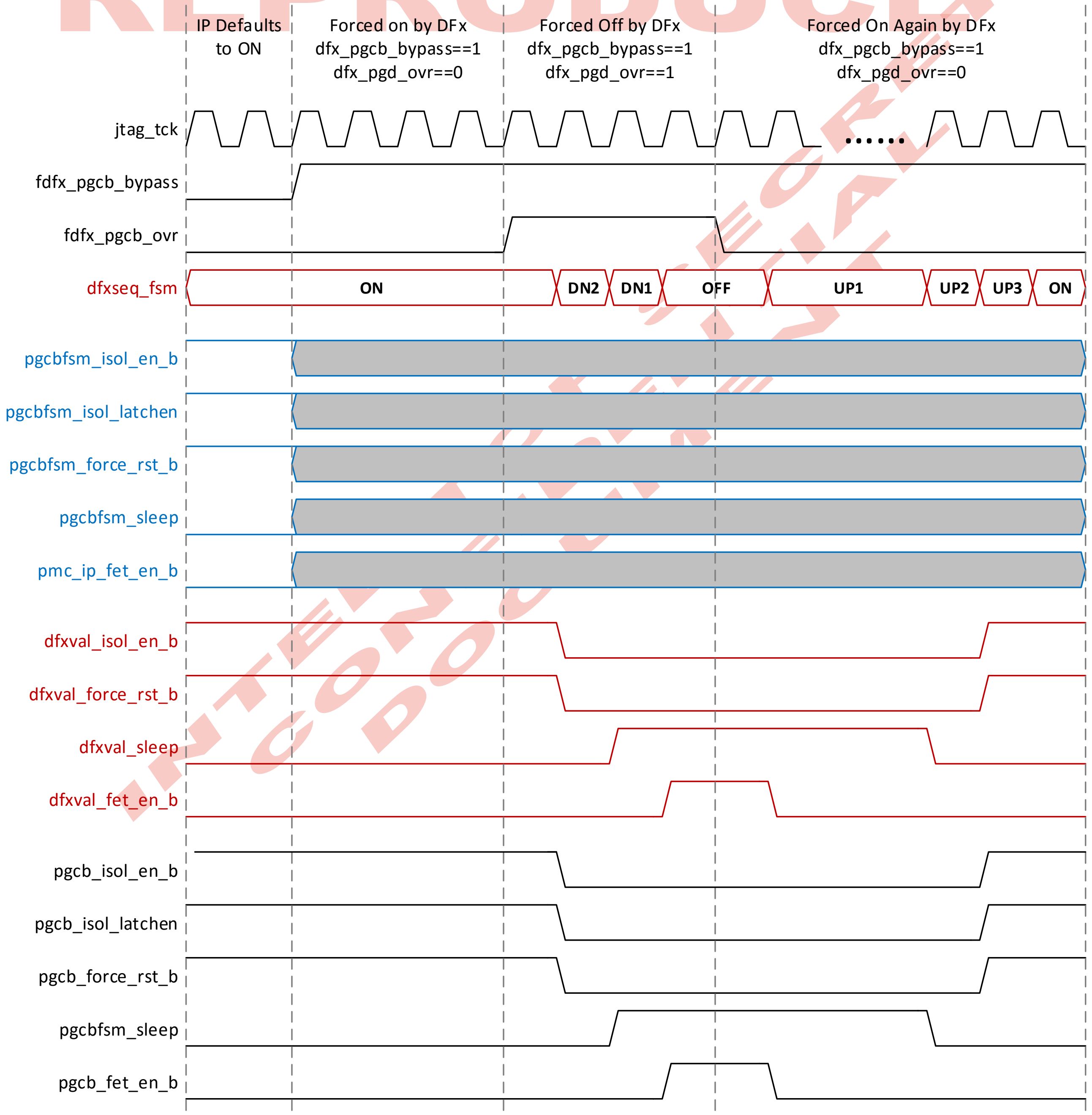
The flows described above are summarized in the table below:

fdfx_pgcb_bypass	fdfx_pgcb_ovr	Result
0	0	PGD state is determined by PGCB
0	1	PGD state is determined by PGCB
1	0	PGD is forced on (powered up)
1	1	PGD is forced off (powered down)

Waveforms

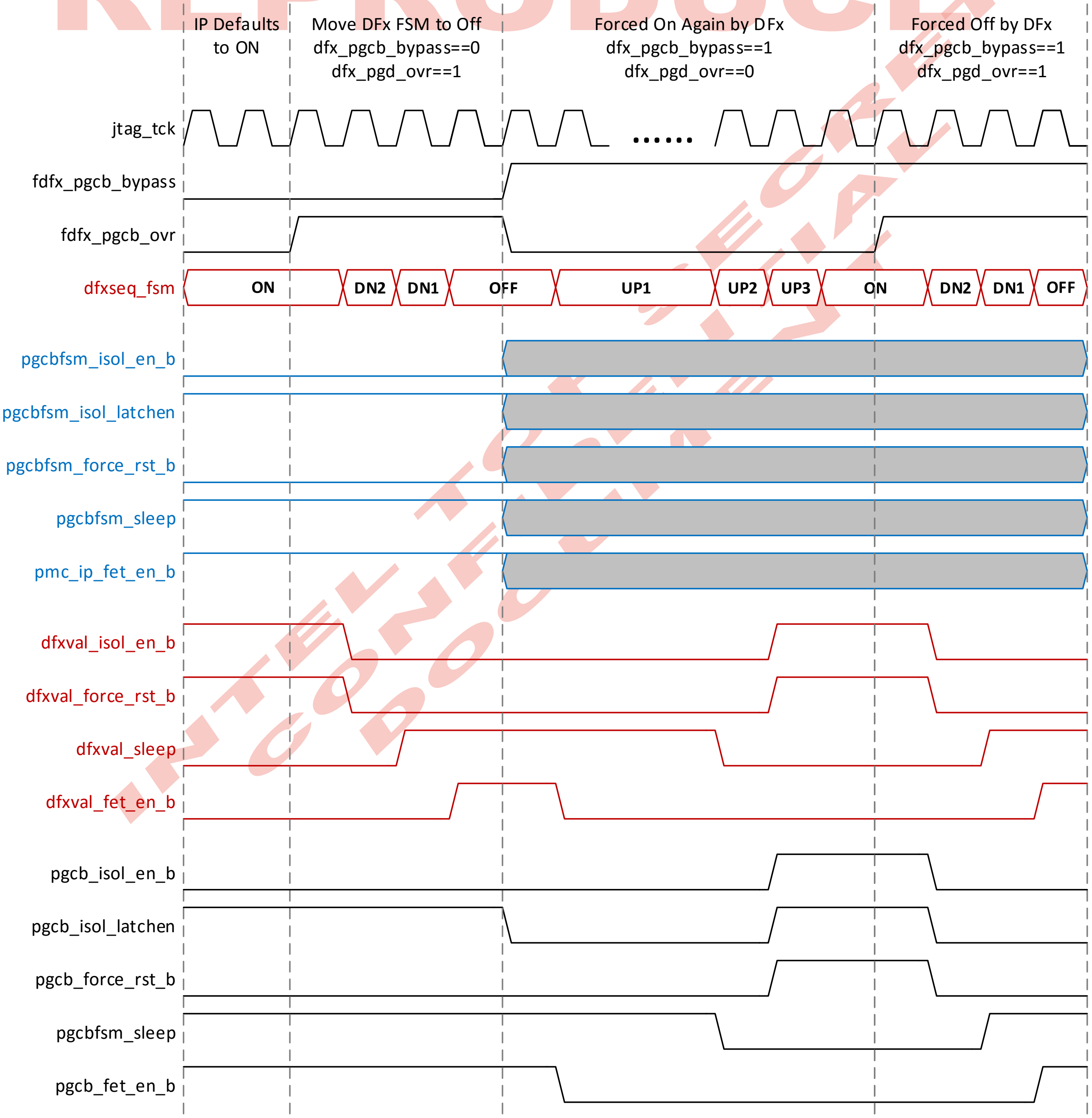
i) Flow – DFx Force Power On, IP Defaults to Powered On:

Flow – DFx Force Power On, IP Defaults to Powered On



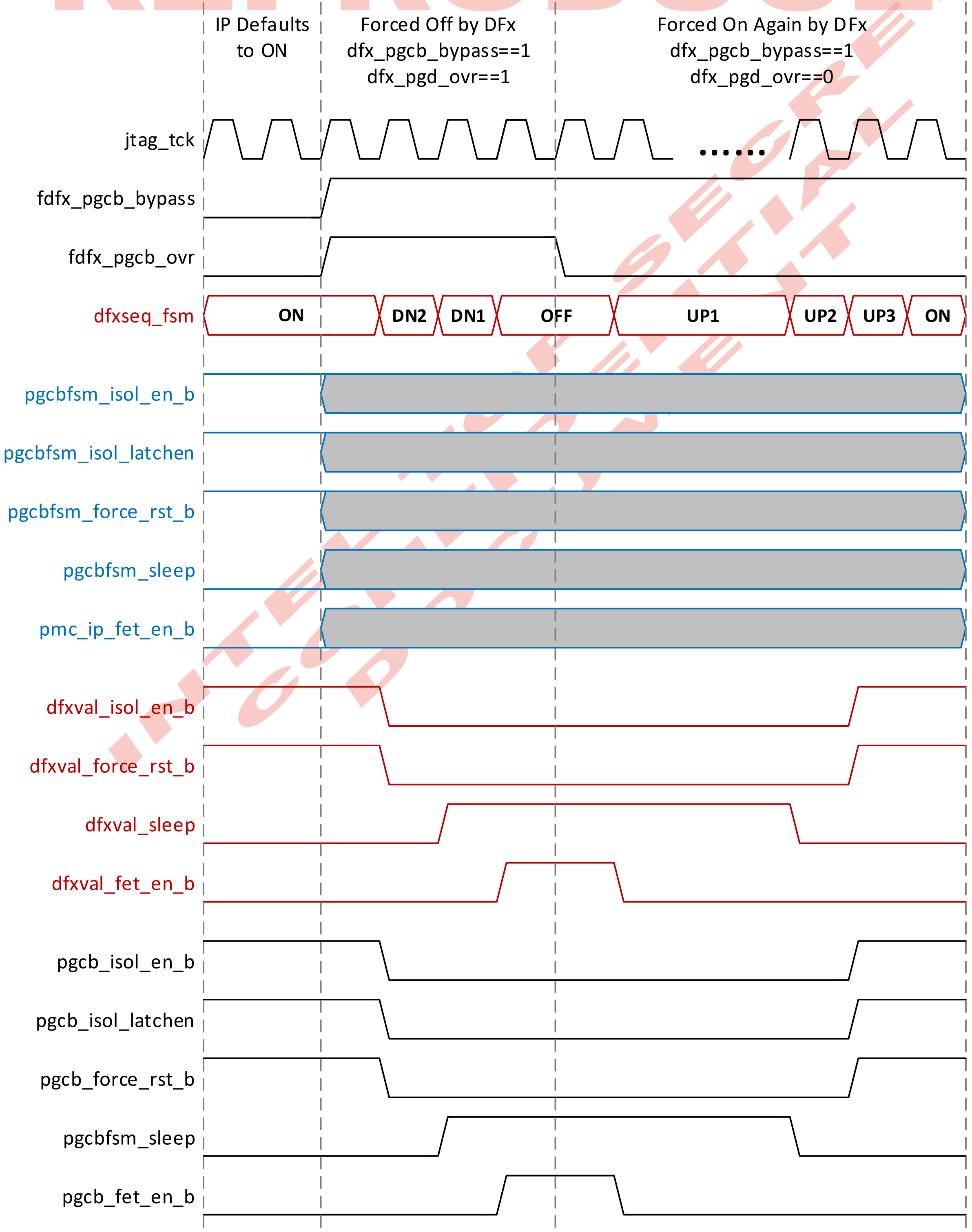
ii) Flow – DFx Force Power On, IP Defaults to Powered Off:

Flow – DFx Force Power On, IP Defaults to Powered Off



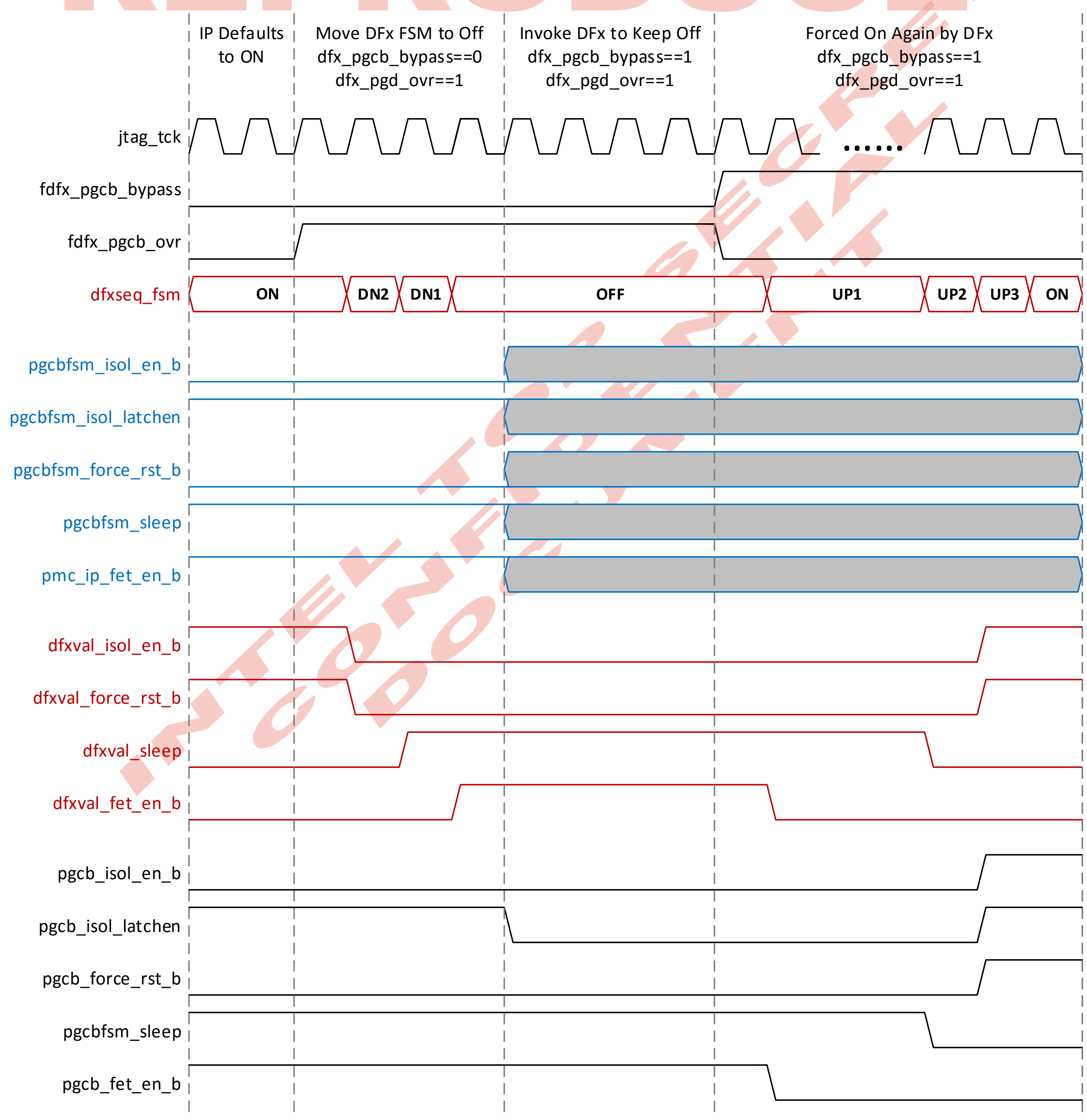
iii) Flow – DFx Force Power Off, IP Defaults to Powered On

Flow – DFx Force Power Off, IP Defaults to Powered On



iv) Flow – DFx Force Power Off, IP Defaults to Powered Off

Flow – DFx Force Power Off, IP Defaults to Powered Off



Method #2 Definition

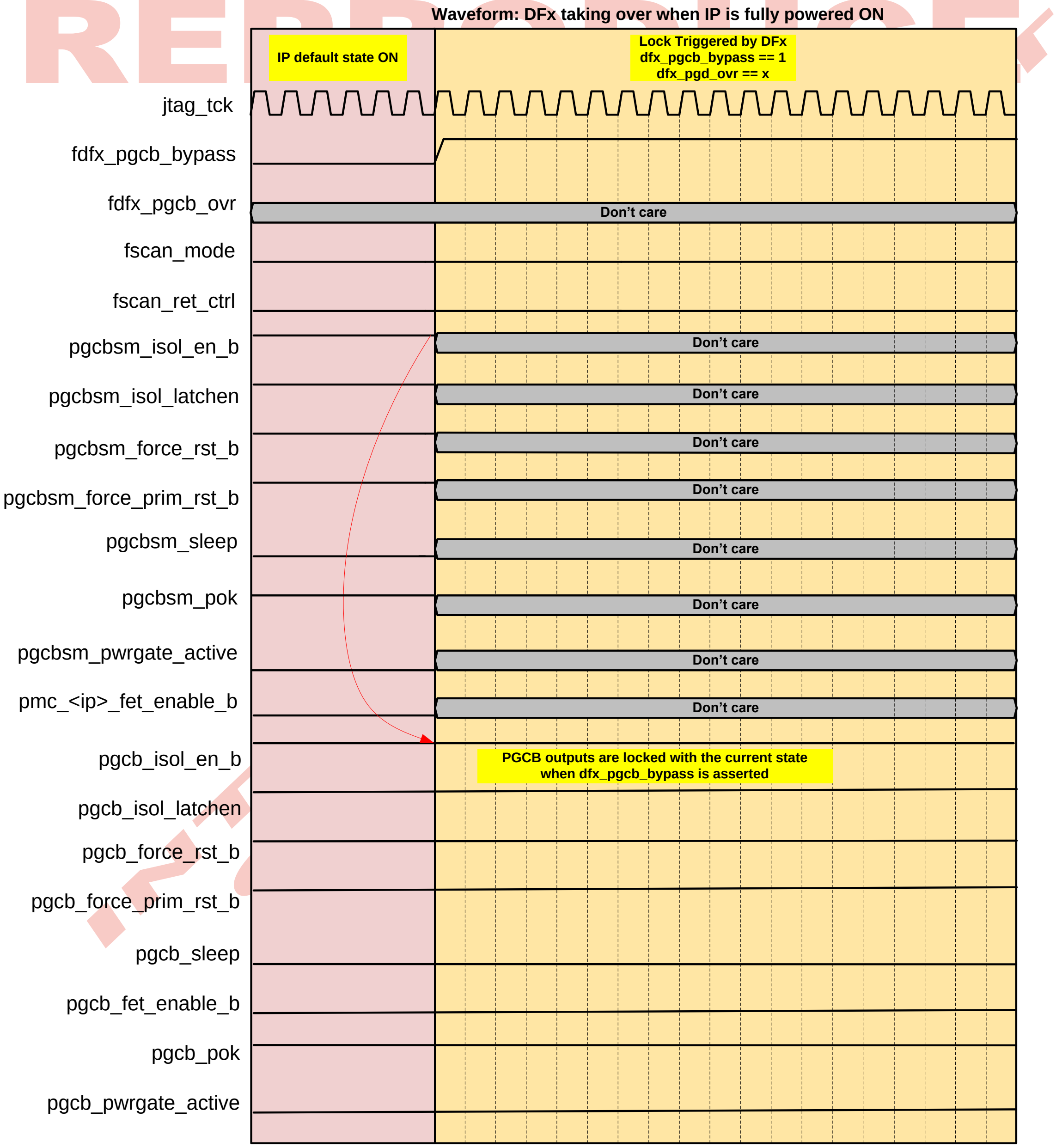
This method does not use the DFx sequencer module. When the #USE_DFX_SEQ compile time parameter is set to 0, the PGCB will be using a locking-mechanism in which it captures the last state of PGCB output and outputs that captured state when DFx mode is enabled. This mode assumes that SoC will have a mean to completely power up the IP's power island using the actual PMC FW image during HVM, before capturing and locking the PGCB signals which are associated with a powered-up state. This method does nothing other than continuously sampling the actual key outputs from the PGCB state machine, and when the fdfx_pgcb_bypass is set to '1', outputs of the PGCB will be transitioned to the last values captured by latches. If the PMC FW fails to completely bring up the IP's power island, this method will not be able to force any of those signals to toggle, as it's more of a capture and output methodology. Method #2 does not use the fdfx_pgcb_ovr control signal input, hence SoC that uses this method need not create a register to control this signal.

Flow – DFx Freeze PGD Controls after PMC Power-Up Sequence:

1. In this flow, the PMC FW of the SoC will bring the IP's power island to be out of reset and fully powered up (i.e. not power gated).
2. When IP is on a powered up state and all PGCB outputs are associated with a powered on state, PGCB is switched to DFx mode by setting the fdfx_pgcb_bypass to '1'.
3. When the control signal is set, the latches that continuously sample the current state of the PGCB outputs will close and last value sampled will be retained.
4. The PGCB outputs immediately transition to the values that are retained by these latches.

Waveform

Flow – DFx Freeze PGD Controls after PMC Power-Up Sequence:



DFx requirements for Functional test modes

The only known functional test mode that the PGCB needs to support is SCANDUMP. This is supported as of PGCB 1.16 which decouples the fdfx_pgcb_ovr and fdfx_pgcb_bypass allowing the FSM to be moved to a required state and only then invoking the overrides. This allows the PGCB's critical outputs to be frozen to "ON" values.

There are no other functional test modes known that the PGCB needs to support.

Special Considerations

Merged PGDs

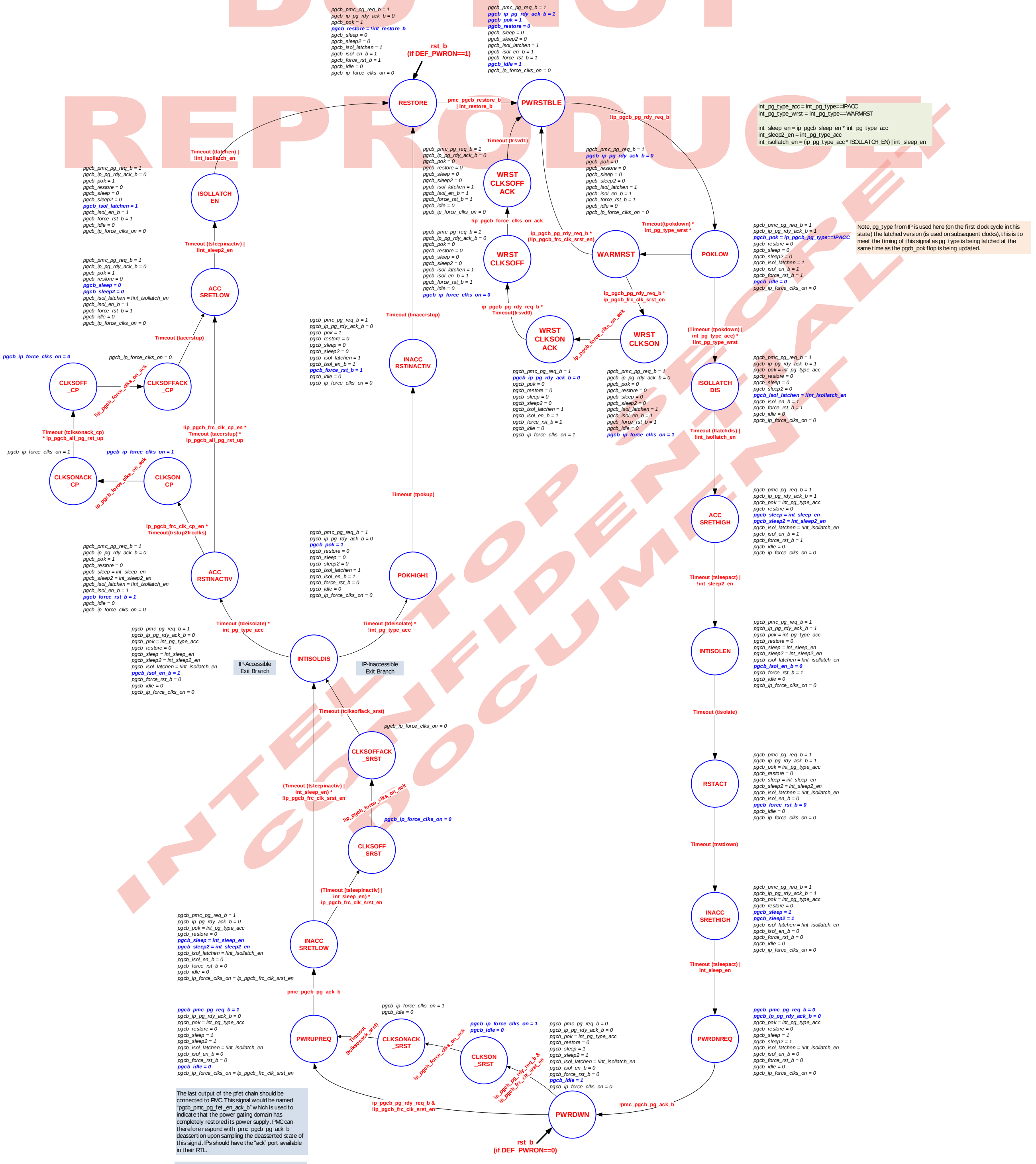
If multiple IPs are merged into one physical power-domain (ie only one PFET control from PMC). Forcing on a PGCB that does not actually control the PFET could potentially cause issues if the PGCB that does control the PFET is still off. It is recommended that the PGCB controlling the PFET be forced on first and forced off last. Such sequencing must be taken care of at a higher level by the test itself.

DFx Sequncer Ungate Timer

The time the DFx sequencer waits for the domain to power up is limited to the # of tclks set by the UNGATE_TIMER parameter (max 256) at compile time and cannot be changed in silicon. If more time is needed to cleanly power up a domain, the only option would be to slow down the tclk to increase the delay.

Appendix

PGCB State Machine



int_pg_type_acc = int_pg_type==IPACC
int_pg_type_wrst = int_pg_type==WARMRST

int_sleep_en = ip_pgcb_sleep_en * int_pg_type_acc
int_sleep2_en = int_pg_type_acc
int_isollatch_en = (ip_pg_type_acc * ISOLLATCH_EN) | int_sleep_en

Note, pg_type from IP is used here (on the first clock cycle in this state) the latched version (is used on subsequent clocks), this is to meet the timing of this signal as pg_type is being latched at the same time as the pgcb_pok flop is being updated.

The last output of the pfet chain should be connected to PMC. This signal would be named "pgcb_pmc_pg_fet_en_ack_b" which is used to indicate that the power gating domain has completely restored its power supply. PMC can therefore respond with pmc_pgcb_pg_ack_b deassertion upon sampling the deasserted state of this signal. IPs should have the "ack" port available in their RTL.

It is expected that PMC will turn on the pfet some time after the request to power ungate has been received from the IP.

PGD starts out power gated from boot and will only be woken up by PMC via assertion of pmc_ip_wake which will trigger the IP to deassert ip_pgcb_pg_req_b to power ungate. (if DEF_PWRON parameter is 0)

PMC will turn off pfet some time after request to power gate from IP has been presented and the corresponding acknowledgement has been triggered.

PGCB VISA Signals

The latest recommendation is that the integrating IP have the VISA tool automatically insert VISA in the PGCB and that the pgcb_visa output be ignored. See the provided .sig files under $MODEL_ROOT/tools/visa.

The following table describes the implementation of the 24 bit output vector ‘pgcb_visa’ (however it is not recommended that this signal be used):

Bits	Signal	Clock domain	Description
23:15	Reserved	n-a	Reserved
14:12	i_pgcbdfxovr1.dfxseq_ps	pgcb_tck	DFx Sequencer Present State
11	i_pgcbfsm1.int_isollatch_en	pgcb_clk	Internal indication of whether isolation latches should ever be closed during flows
10	i_pgcbfsm1.int_sleep_en	pgcb_clk	Internal indication of whether state retention should ever be enabled during flows
9:8	i_pgcbfsm1.int_pg_type	pgcb_clk	Internal latched indication of ip_pgcb_pg_type
7	i_pgcbfsm1.sync_pmc_pgcb_restore_b	pgcb_clk	pmc_pgcb_restore_b synced to pgcb’s clock
6	i_pgcbfsm1.sync_pmc_pgcb_pg_ack_b	pgcb_clk	pmc_pgcb_pg_ack_b synced to pgcb’s clock
5	i_pgcbfsm1.ip_pgcb_pg_rdy_req_b	pgcb_clk	Request from IP to power gate
4:0	i_pgcbfsm1.pgcb_ps	pgcb_clk	PGCB Functional FSM Present State

PGCB Waiver list

Please refer to waiver files for 0-in (clock-crossing waivers) and Spyglass in the documentation folder for the PGCB/CDC release package.

The tools folder in the release package includes the Lintra waivers.

DFx Considerations for TAP logic, etc.

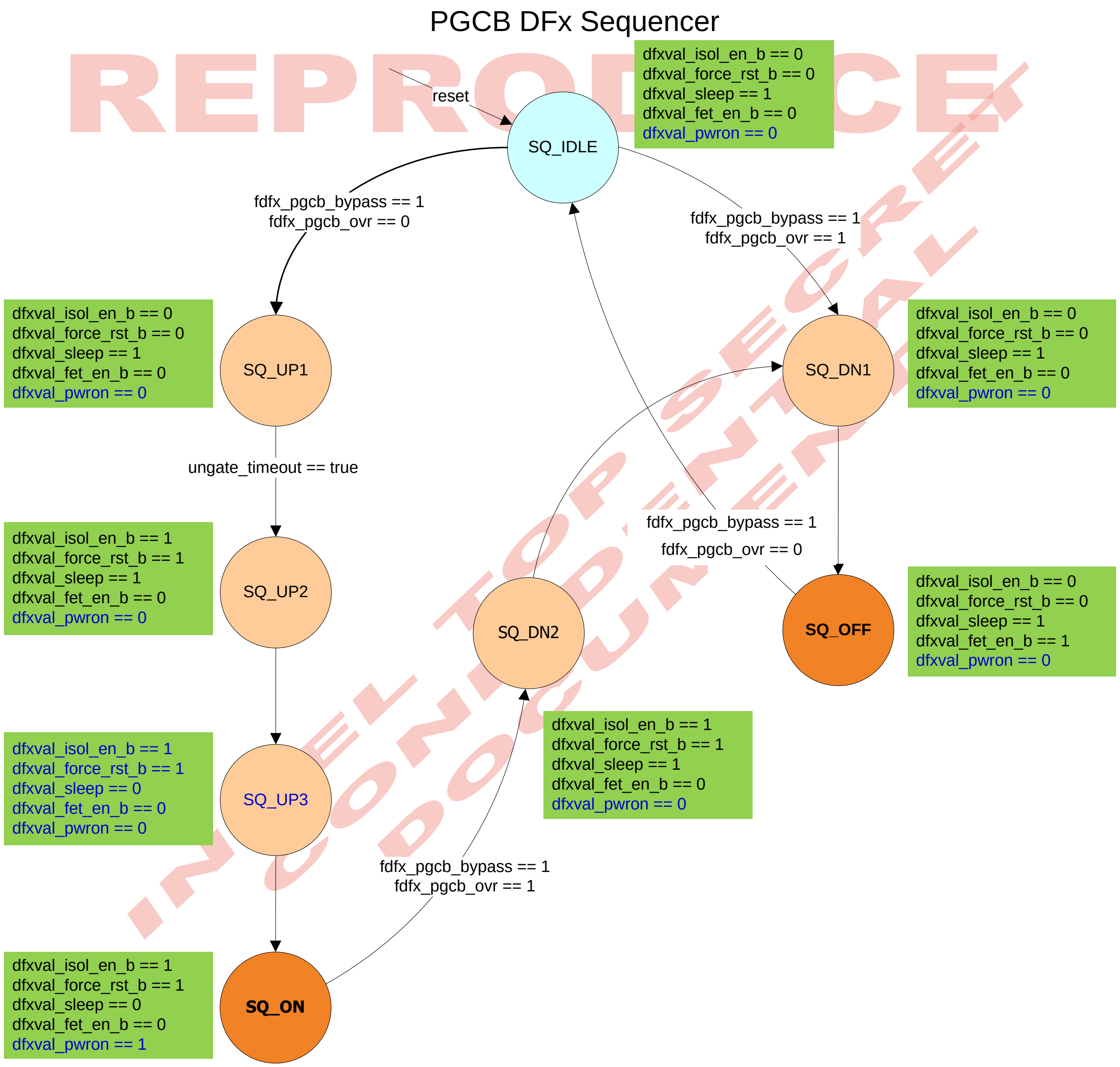
The following considerations apply to the DFX related logic within the PGD of an IP:

1. IP blocks should not gate TCK at all
2. It is okay for an IP to try to powergate when TCK is active – because this actually means that the flow is not working as intended – the intended flow would be to disable PG before activating TCK to an IP block
3. Also, no state retention is implemented for any logic on TCK, so do not see any functional impact at this time
4. "fdfx_powergood_rst_b" should not be synchronized to TCK at the IP top level
5. "pgcb_force_rst_b" needs to be applied to all DFx reset signals (fdfx_rst_b, fdfx_powergood_rst_b, trst_b, etc.) that is consumed by IP logic in power-gated domain. CTECH AND gate should be used to apply the pgcb_force_rst_b overrides on DFx reset signals.
6. Currently, there's no solution identified for TAP that has synchronous reset mechanism using the TMS assertion for 5 TCKs only. SIP that uses sTAP IP provided in IRR should not have this issue as the IP comes with a trst port, which can be constrained with the pgcb_force_rst_b, however if there are SIP using unique TAP without fdfx_powergood and ftap_trst_b and relying only on synchronous reset mechanism, this could still be a problem.

For the case that an IP has clock muxes applied for test/debug purposes (such as HVM tester clock), the following considerations apply:

1. If the IP has an AON domain, it is recommended that the clock mux reside in the AON, prior to being intercepted by a CDC with its associated clock gate (muxing is managed the same way as the SCC clock outputs).
2. If the IP places the mux in the PGD, then the onus is on the test clock provider/source to gate this clock until the time the PGD is up. In other words, this creates a platform level requirement to ensure that the IP is not power-gated when the muxed clock (HVM tester clock for example) is being used.

DFx Sequencer State Machine


PGCB DFx Sequencer
reset
SQ_IDLE
dfxval_isol_en_b == 0
dfxval_force_rst_b == 0
dfxval_sleep == 1
dfxval_fet_en_b == 0
dfxval_pwron == 0
fdfx_pgcb_bypass == 1
fdfx_pgcb_ovr == 0
fdfx_pgcb_bypass == 1
fdfx_pgcb_ovr == 1
SQ_UP1
dfxval_isol_en_b == 0
dfxval_force_rst_b == 0
dfxval_sleep == 1
dfxval_fet_en_b == 0
dfxval_pwron == 0
SQ_DN1
dfxval_isol_en_b == 0
dfxval_force_rst_b == 0
dfxval_sleep == 1
dfxval_fet_en_b == 0
dfxval_pwron == 0
ungate_timeout == true
SQ_UP2
dfxval_isol_en_b == 1
dfxval_force_rst_b == 1
dfxval_sleep == 1
dfxval_fet_en_b == 0
dfxval_pwron == 0
fdfx_pgcb_bypass == 1
fdfx_pgcb_ovr == 0
SQ_DN2
SQ_OFF
dfxval_isol_en_b == 0
dfxval_force_rst_b == 0
dfxval_sleep == 1
dfxval_fet_en_b == 1
dfxval_pwron == 0
dfxval_isol_en_b == 1
dfxval_force_rst_b == 1
dfxval_sleep == 1
dfxval_fet_en_b == 0
dfxval_pwron == 0
SQ_UP3
dfxval_isol_en_b == 1
dfxval_force_rst_b == 1
dfxval_sleep == 0
dfxval_fet_en_b == 0
dfxval_pwron == 0
fdfx_pgcb_bypass == 1
fdfx_pgcb_ovr == 1
SQ_ON
dfxval_isol_en_b == 1
dfxval_force_rst_b == 1
dfxval_sleep == 0
dfxval_fet_en_b == 0
dfxval_pwron == 1