

Name	Date modified	Type	Size
PLL_IP.vhd	29/12/2016 9:12	Text Document	16 KB
PLL3.v	26/10/2017 8:45	Text Document	17 KB
pwbtrn.vhd	20/11/2017 7:47	Text Document	3 KB
vpp_vddq_vh.vhd	20/11/2017 7:47	Text Document	3 KB
WD_VH.vhd	20/11/2017 7:47	Text Document	2 KB
vccio_en_vh.vhd	08/08/2019 14:33	Text Document	1 KB
voltage_monitor_new.vhd	02/09/2019 10:55	Text Document	7 KB
rsmrst_pwrgd.vh.vhd	10/11/2019 16:28	Text Document	3 KB
pch_pwrok.vh.vhd	18/11/2019 11:36	Text Document	3 KB
hda_strap.vh.vhd	15/06/2020 20:23	Text Document	3 KB
powered_vh.vhd	16/06/2020 10:44	Text Document	6 KB
dsw_pwrok.vh.vhd	11/06/2021 13:09	Text Document	2 KB
dsw_pwrok.vhd	11/06/2021 15:44	Text Document	3 KB
vccsa_vr_en_vh.vhd	15/06/2021 15:41	Text Document	1 KB

rsmrst_pwrgd_block

```

library ieee;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

-- RSMRSTn is an active-high power good signal for main S5 rails: +3V3A, +1V05A, +5VA.
-- RSMRSTn is 10 ms delayed (on rising edge only) RSMRSTn
-- There should be a 10msec delay between the PG of the rails to RSMRSTn assertion. NOW 50 msec.

entity rsmrst_pwrgd_block is
port (
    V33A_OK:      in std_logic; -- Open-drain, internal weak pull-up required
    V105A_OK:     in std_logic; -- Open-drain, internal weak pull-up required
    V5A_OK:       in std_logic; -- Open-drain, internal weak pull-up required
    tpm_gpio:     in std_logic; -- Provision
    SLP_SUSn:      in std_logic;
    clk_100k:      in std_logic; -- 100KHz clock, T = 10uSec
    RSMRSTn:      out std_logic; --with 10ms delay on rising edge
    rsmrst_pwrgd_out: out std_logic);--without delay
end rsmrst_pwrgd_block;

architecture rsmrst_arch of rsmrst_pwrgd_block is
type state_type is (pwrgd,no_pwrgd, delay);
attribute enum_encoding : string;
attribute enum_encoding of state_type : type is "01 00 10"; --<< no_pwrgd (00) is default after FPGA power-on
signal curr_state: state_type := no_pwrgd;
signal rsmrst_pwrgd: std_logic := '0';
signal count : unsigned(15 downto 0) := (others => '0');
begin

rsmrst_pwrgd_out <= rsmrst_pwrgd;
rsmrst_pwrgd <= '1' when (V33A_OK = '1') and (V5A_OK = '1') and (V105A_OK = '1') and (SLP_SUSn = '1') -- SLP_SUSn was added
else '0';

process (clk_100k)
begin
    if (clk_100k = '1') then
        case curr_state is

            when pwrgd =>
                if (rsmrst_pwrgd = '1') then
                    curr_state <= pwrgd;
                    RSMRSTn <= '1';
                else
                    curr_state <= no_pwrgd; -- Delay at RSMRST_PWROK transition from 0 to 1
                    RSMRSTn <= '0'; -- The RSMRSTn signal will not assert at pwrok glitches of less than 1T
                end if;

            when delay =>
                if (count = to_unsigned(10000,16)) then -- 10000 * 50uSec = 100 mSec (was 100msec at ATSKL)
                    curr_state <= pwrgd;
                    count <= (others => '0');
                else
                    count <= count + 1;
                    curr_state <= delay;
                end if;
                RSMRSTn <= '0';

            when no_pwrgd =>
                if (rsmrst_pwrgd = '1') then
                    curr_state <= delay;
                    count <= (others => '0');
                else
                    curr_state <= no_pwrgd;
                end if;
                RSMRSTn <= '0';

        end case;
    end if;
end process;
end rsmrst_arch;

```

pwrbtn_block

```

library ieee;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

-- The purpose of this block is to perform auto-on.
-- PWRBTN# can be preset before or after RSMRST. Here, RSMRST rising edge will trigger PWRBTN#.
-- Minimum period of PWRBTN# toggle is 16ms (de-bounce in PCH). Here, PWRBTN# is asserted for 30ms.

entity pwrbtn_block is
port (
    rsmrst_n:    in std_logic; -- RSMRST#, active low (connected on-board to DSW_PWROK)
    clk_100k:    in std_logic; -- 100KHz clock, T = 10uSec
    pwrbtn:      out std_logic);
end pwrbtn_block;

architecture pwrbtn_arch of pwrbtn_block is
type state_type is (asserted, not_asserted, idle);
attribute enum_encoding : string;
attribute enum_encoding of state_type : type is "01 00 10"; --<< not_asserted is default after FPGA power-on
signal curr_state: state_type := not_asserted;
signal trigger: std_logic := '0';
signal count : unsigned(15 downto 0) := (others => '0');
begin

process (clk_100k) -- 30 mSec delay process: asserted -> not_asserted
begin
    if (clk_100k = '1') then
        case curr_state is

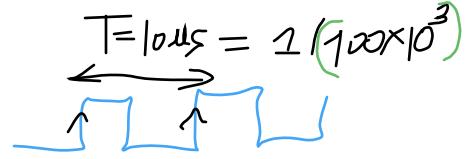
            when not_asserted => -- Before assertion
                if (count = to_unsigned(3000,16)) then -- 3000 * 10uSec = 30 mSec
                    curr_state <= asserted;
                    count <= (others => '0');
                else
                    if (rsmrst_n = '1') then
                        count <= count + 1;
                        curr_state <= not_asserted;
                    else
                        count <= (others => '0');
                        curr_state <= not_asserted;
                    end if;
                end if;
                pwrbtn <= '1';

            when asserted => -- during assertion
                if (count = to_unsigned(3000,16)) then -- 3000 * 10uSec = 30 mSec
                    curr_state <= idle;
                    count <= (others => '0');
                else
                    count <= count + 1;
                    curr_state <= asserted;
                end if;
                pwrbtn <= '0';

            when idle => -- After assertion
                if (rsmrst_n = '0') then
                    curr_state <= not_asserted; -- RSMRST# loss (relevant for major power failure while FPGA still has power)
                    pwrbtn <= '1';
                else
                    curr_state <= idle;
                    pwrbtn <= '1';
                end if;

        end case;
    end if;
end process;
end pwrbtn_arch;

```



only after finishing
dealing the if & else,
we reach here and
update pwrbtn




```

if (clk_100k = '1') then
    case curr_state is

when pwm_low_state => -- pwm is low
    if (count > to_unsigned(periodclocks,16)) then -- end of period, PWM goes high
        curr_state <= pwm_high_state;
        count <= (others => '0');
    else
        count <= count + 1;
        curr_state <= pwm_low_state;
        pwm_out <= '0';      -- Moved here
    end if;
    pwm_out <= '0';

when pwm_high_state => -- pwm is high
    if (count >= to_unsigned(nclocks,16)) then -- end of on period, PWM goes low
        curr_state <= pwm_low_state;
        count <= count + 1;
    else
        count <= count + 1;
        curr_state <= pwm_high_state;
        pwm_out <= '1';      -- Moved here
    end if;
    pwm_out <= '1';

when others =>
    curr_state <= pwm_low_state;

end case;
end if;
end process;
end powerled_arch;

```

```

hda_strap_block

library ieee;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

-- 

entity hda_strap_block is
port (
    pch_pwrok:      in std_logic; -- SLP_S3#
    GPIO_PCH:       in std_logic; -- Open-drain, internal weak pull-up required before reset, we pull GPIO_PCH=0
    clk_100k:        in std_logic; -- 100KHz clock, T = 10uSec
    HDA_SDO_FPGA:   out std_logic);
end hda_strap_block;

architecture hda_strap_block_arch of hda_strap_block is
type state_type is (start, startok, idle, b4reset, reset, afterreset);
attribute enum_encoding : string;
attribute enum_encoding of state_type : type is "000 001 010 011 100 101"; --<< State 'start' is default after FPGA power-on
signal curr_state: state_type := start;
signal count : unsigned(17 downto 0) := (others => '0');
begin

process (clk_100k) --
begin
    if (clk_100k = '1') then
        case curr_state is
            when start => -- After FPGA power on, waiting for PCH PWROK = 1
                if (pch_pwrok = '1') then
                    curr_state <= startok;
                else
                    curr_state <= start;
                end if;
                HDA_SDO_FPGA <= '0';
            when startok => -- After PCH PWROK = 1, waiting for 2 seconds
                if (count = to_unsigned(200000,18)) then -- 200000 * 10uSec = 2 Sec
                    curr_state <= idle;
                    count <= (others => '0');
                else
                    count <= count + 1;
                    curr_state <= startok;
                end if;
                HDA_SDO_FPGA <= '0';

            when idle => -- After PCH PWROK = 1 and 2 seconds, normal operation. Waiting for GPIO = 0
                if (GPIO_PCH = '0') then
                    curr_state <= b4reset;
                else
                    curr_state <= idle;
                end if;
                HDA_SDO_FPGA <= '0';

            when b4reset => -- After GPIO=0. Waiting for PCH PWROK = 0 (reset). HDA SDO goes to 1.
                if (pch_pwrok = '0') then
                    curr_state <= reset;
                else
                    curr_state <= b4reset;
                end if;
                HDA_SDO_FPGA <= '1';

            when reset => -- During cold reset (PCH_PWROK=0). Waiting for PCH_PWROK = 0 (reset)
                if (pch_pwrok = '1') then
                    curr_state <= afterreset;
                else
                    curr_state <= reset;
                end if;
                HDA_SDO_FPGA <= '1';

            when afterreset => -- After PCH_PWROK = 1, waiting for 2 seconds
                if (count = to_unsigned(200000,18)) then -- 200000 * 10uSec = 2 Sec
                    curr_state <= idle;
                    count <= (others => '0');
                else
                    count <= count + 1;
                    curr_state <= afterreset;
                end if;
                HDA_SDO_FPGA <= '1';

        end case;
    end if;
end process;

end hda_strap_block_arch;

```

dsw_pwrok_block

```
library ieee;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

-- DSW PWROK is an active-high power good signal for main S5 rails: +3V3DSW.
-- There should be a 10msec delay between the V33DSW OK assertion to DSW PWROK assertion. (tPCH02)
-- At power loss, DSW PWROK must go low as fast as possible, for RTC isolation.

entity dsw_pwrok_block is
port (
    V33DSW_OK: in std_logic;
    vdc_ok:     in std_logic; -- From power monitor
    clk_100k:   in std_logic; -- 100KHz clock, T = 10uSec
    DSW_PWROK: out std_logic);
end dsw_pwrok_block;

architecture dsw_pwrok_arch of dsw_pwrok_block is
type state_type is (pwrgd,no_pwrgd, delay);
attribute enum_encoding : string;
attribute enum_encoding of state_type : type is "01 00 10"; --<< no_pwrgd is default after FPGA power-on
signal curr_state: state_type := no_pwrgd;
signal all_pwrok: boolean;
signal V33DSW_OK_delayed: std_logic;
signal count : unsigned(15 downto 0) := (others => '0');
begin

    DSW_PWROK <= V33DSW_OK_delayed and vdc_ok ;
    all_pwrok <= (V33DSW_OK = '1');

process (clk_100k)
begin
    if (clk_100k = '1') then
        case curr_state is
            when pwrgd =>
                if (all_pwrok) then
                    curr_state <= pwrgd;
                    V33DSW_OK_delayed <= '1';
                else
                    curr_state <= no_pwrgd; -- Delay at RSMRST_PWROK transition from 0 to 1
                    V33DSW_OK_delayed <= '0'; -- The V33DSW_OK_delayed signal will not assert at pwrok
glitches of less than 1T
                    end if;
            when delay =>
                if (count = to_unsigned(1000,16)) then -- 1000 * 10uSec = 10 mSec (was to 100usec at ATSKL)
                    curr_state <= pwrgd;
                    count <= (others => '0');
                else
                    count <= count + 1;
                    curr_state <= delay;
                end if;
                V33DSW_OK_delayed <= '0';

            when no_pwrgd =>
                if (all_pwrok) then
                    curr_state <= delay;
                    count <= (others => '0');
                else
                    curr_state <= no_pwrgd;
                end if;
                V33DSW_OK_delayed <= '0';
        end case;
    end if;
end process;
end dsw_pwrok_arch;
```

we reach pwrgd
state only after the
delay is over

WD

```
library ieee;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity WD is
generic (timeout_sec:integer; -- Time to reset parameter [sec]
          reset_msec:integer); -- Reset duration [msec]
port (
      wdto_reset_n: out std_logic;
      wd_rst:   in std_logic;
      clk_100k: in std_logic); -- 100KHz
end WD;

architecture WD_arch of WD is
type state_type is (reset, waiting, timeout);
attribute enum_encoding : string;
attribute enum_encoding of state_type : type is "01 00 10"; --<< waiting is default after FPGA power-on
signal curr_state: state_type := timeout;
signal next_state: state_type := timeout;
signal milisec : unsigned(23 downto 0);
signal milisec_max : unsigned(23 downto 0);
signal sec : unsigned(7 downto 0);
signal sec_max : unsigned(7 downto 0);
begin
begin

milisec_max <= to_unsigned(100000,24); --periods of 0.01 msec in one second

process (next_state, wd_rst)
begin
  if (wd_rst='0') then
    curr_state <= reset;
  else
    curr_state <= next_state;
  end if;
end process;

process (clk_100k)
begin
--  if (wd_rst='0') then
--    curr_state <= reset;
--  elsif (rising_edge(clk_100k)) then
  if (rising_edge(clk_100k)) then
    case curr_state is
      when waiting =>
        milisec <= milisec + 1;
        if (milisec = milisec_max) then
          sec <= sec + 1;
          milisec <= (others => '0');
        end if;
        if (sec = to_unsigned(timeout_sec,8)) then
          next_state <= timeout;
          sec <= (others => '0');
        else
          next_state <= waiting;
        end if;
        wdto_reset_n <= '1';
      when reset =>
        milisec <= (others => '0');
        sec <= (others => '0');
        wdto_reset_n <= '1';
        next_state <= waiting;
      when timeout =>
        milisec <= milisec + 1;
        if (milisec = to_unsigned(100*reset_msec,24)) then
          milisec <= (others => '0');
          next_state <= waiting;
        else
          next_state <= timeout;
        end if;
        wdto_reset_n <= '0';
      end case;
--    curr_state <= next_state;
    end if;
  end process;
end WD_arch;
```

```

vpp_vddq_block
library ieee;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

-- VPP must always be on when VDDQ is on. (VPP >= VDDQ)
-- VPP_EN = (SLP_S4#) OR (30msec_Delayed(VDDQ_PWRGD)) <- The delay is a must only at falling edge.
-- VDDQ_EN = (VPP_PWRGD) AND (SLP_S4#)

entity vpp_vddq_block is
port (
    slp_s4:      in std_logic; -- SLP_S4#
    vddq_pwrqd:  in std_logic; -- Open-drain, internal weak pull-up required
    vpp_pwrqd:   in std_logic; -- Open-drain, internal weak pull-up required
    clk_100k:    in std_logic; -- 100KHz clock, T = 10uSec
    vpp_en:      out std_logic;
    vddq_en:     out std_logic);
end vpp_vddq_block;

architecture vpp_vddq_arch of vpp_vddq_block is
type state_type is (pwrqd, no_pwrqd, delay);
attribute enum_encoding : string;
attribute enum_encoding of state_type : type is "01 00 10"; --<< no_pwrqd is default after FPGA power-on
signal curr_state: state_type := no_pwrqd;
signal delayed_vddq_pwrqd: std_logic := '0';
signal count : unsigned(15 downto 0) := (others => '0');
begin

vpp_en <= '1' when (slp_s4 = '1') or (delayed_vddq_pwrqd = '1')
else '0';

vddq_en <= '1' when (vpp_pwrqd = '1') and (slp_s4 = '1')
else '0';

process (clk_100k) -- 30 mSec delay process: vddq_pwrqd -> delayed_vddq_pwrqd
begin
    if (clk_100k = '1') then
        case curr_state is

            when pwrqd =>
                if ((vddq_pwrqd = '1') and (slp_s4 = '1')) then
                    curr_state <= pwrqd;
                    delayed_vddq_pwrqd <= '1';
                else
                    curr_state <= delay; -- Delay at vddq_pwrqd transition from 1 to 0
                    count <= (others => '0');
                end if;

            when delay => -- According to Skylake / Kabylake PDG and JEDEC DDR4: 30 mSec between VDDQ off to VPP off
                if (count = to_unsigned(3000,16)) then -- 3000 * 10uSec = 30 mSec
                    curr_state <= no_pwrqd;
                    count <= (others => '0');
                else
                    count <= count + 1;
                    curr_state <= delay;
                end if;
                delayed_vddq_pwrqd <= '1';

            when no_pwrqd =>
                if ((vddq_pwrqd = '1') and (slp_s4 = '1')) then
                    curr_state <= pwrqd; -- transition to high can be done without a delay (SLP_S4# is already high)
                    delayed_vddq_pwrqd <= '1';
                else
                    curr_state <= no_pwrqd;
                    delayed_vddq_pwrqd <= '0'; -- delayed_vddq_pwrqd signal will not assert at vddq_pwrqd glitches
                end if;

        end case;
    end if;
end process;
end vpp_vddq_arch;

```

```

vccsa_vr_en_block
library ieee;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

-- CPU VR must ramp up after +12VS, +5VS, +3V3S and VCCIO are OK.
-- VCCSA can ramp up after VCCIO.

entity vccsa_vr_en_block is
port (
    v12s_pwrqd:  in std_logic; --iPU pulls up also if power module is not connected by ribbon
    v5s_pwrqd:  in std_logic;
    v33s_pwrqd:  in std_logic;
    vccio_pwrok:  in std_logic; -- Internal weak pull-up required
    slp_s3:      in std_logic;
    rsmrst_pwrqd: in std_logic;
    clk_100k:     in std_logic; -- 100KHz clock, T = 10uSec <- PROVISION, NOT IN USE
    vr_en:        out std_logic;
    vccsa_en:     out std_logic);
end vccsa_vr_en_block;

architecture vccsa_vr_arch of vccsa_vr_en_block is
signal output: std_logic;
begin

output <= '1' when (v12s_pwrqd = '1') and (v5s_pwrqd = '1') and (v33s_pwrqd = '1') and (vccio_pwrok = '1') and (slp_s3 = '1') and
(rsmrst_pwrqd = '1')
else '0';

vr_en <= output;
vccsa_en <= output;

end vccsa_vr_arch;

```

Table 45-2. CFL Power Sequence Related Power Rails

Name	Source	Destination	Description
VCCRTC	Platform	PCH	3.05-V supplies for PCH RTC Well. This power is not expected to be shut off in any of the sleep states unless the RTC battery is removed or completely drained.
VCCDSW_3p3	Platform	PCH	3.3-V supply for Deep Sx wells. If Deep Sx is not supported on the platform, tie to VCCPRIM_3p3
VCCPRIM_1p0/1p8/ 3p3	Platform	PCH	PCH I/O and Misc rails 1.0/1.8/3.3V (Primary Well)
VCCSPI	Platform	PCH	1.8V/3.3V supply for the SPI IO. This rail must be powered when VCCPRIM is powered.
VCC	Processor	Processor	Processor core rail
VCCST	Processor	Processor	Sustain voltage for processor in Standby modes
VCCPLL	Processor	Processor	CPU PLL power rails
VCCPLL_OC	Processor	Processor	CPU digital PLL power rails
VCCGT	Processor	Processor	Sliced graphics power rail
VCCGTX ¹	Processor	Processor	Unsliced graphics power rail
VCCIO	Processor	Processor	IO power rail
VDDQ	Processor	Processor	CPU Memory power rail, voltage dependent on memory technology
VPP	Processor	Processor	CPU Memory power rail, voltage dependent on memory technology
VCCSA	Processor	Processor	System Agent power rail

Table 228. Tiger Lake Power Sequence Related Power Rails

Name	Source	Destination	Description
Common			
VCCRTC	Platform	PCH	2.0-3.3V +5% supplies for PCH RTC Well. This power is not expected to be shut off in any of the sleep states unless the RTC battery is removed or completely drained.
Common			
VCCDSW_3P3	Platform	PCH	3.3-V supply for Deep Sx wells. If Deep Sx is not supported on the platform, tie to VCCPRIM_3p3
Common			
VCC1P05_OUT_FET	PCH	Platform	FIVR output of PCH to platform 1.05V Power Gates (VCCST/VCCSTG)
Common			
VCCPRIM_1P8/3P3 ²	Platform	PCH	PCH I/O and Misc rails 1.8/3.3V (Primary Well)
Common			
VCC_VNNEXT_1P05 VNN_BYP	Platform	PCH	Optional BYPASS rail for PCH Prime Core Well (760mV in S0/S1 and 1.05V in S states) or reduced power consumption in low power states
Common			
VCC_V1P0SEXT_1P05 V1P05_BYP	Platform	PCH	Optional BYPASS rail for PCH Primary Well (1.05V) for reduced power consumption in low power states
Common			
VCCST	Processor	Processor	Sustain voltage for processor in Standby modes
Common			
VCCSTG	Processor	Processor	Gated version of VCCST
Common			
VPP	Processor	Memory	Memory power rail, voltage dependent on memory technology
Common			
VDDQ	Processor	Processor	CPU Memory power rail, voltage dependent on memory technology
Common			
VCCIN	Processor	Processor	Processor FIVR Input power supply
Common			
VCCIN_AUX	Platform	PCH	PCH FIVR input power supply

¹VCCGTX₁, VCCGTX₂, VCCGTX₃, VCCGTX₄, VCCGTX₅, VCCGTX₆, VCCGTX₇, VCCGTX₈, VCCGTX₉, VCCGTX₁₀, VCCGTX₁₁, VCCGTX₁₂, VCCGTX₁₃, VCCGTX₁₄, VCCGTX₁₅, VCCGTX₁₆, VCCGTX₁₇, VCCGTX₁₈, VCCGTX₁₉, VCCGTX₂₀, VCCGTX₂₁, VCCGTX₂₂, VCCGTX₂₃, VCCGTX₂₄, VCCGTX₂₅, VCCGTX₂₆, VCCGTX₂₇, VCCGTX₂₈, VCCGTX₂₉, VCCGTX₃₀, VCCGTX₃₁, VCCGTX₃₂, VCCGTX₃₃, VCCGTX₃₄, VCCGTX₃₅, VCCGTX₃₆, VCCGTX₃₇, VCCGTX₃₈, VCCGTX₃₉, VCCGTX₄₀, VCCGTX₄₁, VCCGTX₄₂, VCCGTX₄₃, VCCGTX₄₄, VCCGTX₄₅, VCCGTX₄₆, VCCGTX₄₇, VCCGTX₄₈, VCCGTX₄₉, VCCGTX₅₀, VCCGTX₅₁, VCCGTX₅₂, VCCGTX₅₃, VCCGTX₅₄, VCCGTX₅₅, VCCGTX₅₆, VCCGTX₅₇, VCCGTX₅₈, VCCGTX₅₉, VCCGTX₆₀, VCCGTX₆₁, VCCGTX₆₂, VCCGTX₆₃, VCCGTX₆₄, VCCGTX₆₅, VCCGTX₆₆, VCCGTX₆₇, VCCGTX₆₈, VCCGTX₆₉, VCCGTX₇₀, VCCGTX₇₁, VCCGTX₇₂, VCCGTX₇₃, VCCGTX₇₄, VCCGTX₇₅, VCCGTX₇₆, VCCGTX₇₇, VCCGTX₇₈, VCCGTX₇₉, VCCGTX₈₀, VCCGTX₈₁, VCCGTX₈₂, VCCGTX₈₃, VCCGTX₈₄, VCCGTX₈₅, VCCGTX₈₆, VCCGTX₈₇, VCCGTX₈₈, VCCGTX₈₉, VCCGTX₉₀, VCCGTX₉₁, VCCGTX₉₂, VCCGTX₉₃, VCCGTX₉₄, VCCGTX₉₅, VCCGTX₉₆, VCCGTX₉₇, VCCGTX₉₈, VCCGTX₉₉, VCCGTX₁₀₀, VCCGTX₁₀₁, VCCGTX₁₀₂, VCCGTX₁₀₃, VCCGTX₁₀₄, VCCGTX₁₀₅, VCCGTX₁₀₆, VCCGTX₁₀₇, VCCGTX₁₀₈, VCCGTX₁₀₉, VCCGTX₁₁₀, VCCGTX₁₁₁, VCCGTX₁₁₂, VCCGTX₁₁₃, VCCGTX₁₁₄, VCCGTX₁₁₅, VCCGTX₁₁₆, VCCGTX₁₁₇, VCCGTX₁₁₈, VCCGTX₁₁₉, VCCGTX₁₂₀, VCCGTX₁₂₁, VCCGTX₁₂₂, VCCGTX₁₂₃, VCCGTX₁₂₄, VCCGTX₁₂₅, VCCGTX₁₂₆, VCCGTX₁₂₇, VCCGTX₁₂₈, VCCGTX₁₂₉, VCCGTX₁₃₀, VCCGTX₁₃₁, VCCGTX₁₃₂, VCCGTX₁₃₃, VCCGTX₁₃₄, VCCGTX₁₃₅, VCCGTX₁₃₆, VCCGTX₁₃₇, VCCGTX₁₃₈, VCCGTX₁₃₉, VCCGTX₁₄₀, VCCGTX₁₄₁, VCCGTX₁₄₂, VCCGTX₁₄₃, VCCGTX₁₄₄, VCCGTX₁₄₅, VCCGTX₁₄₆, VCCGTX₁₄₇, VCCGTX₁₄₈, VCCGTX₁₄₉, VCCGTX₁₅₀, VCCGTX₁₅₁, VCCGTX₁₅₂, VCCGTX₁₅₃, VCCGTX₁₅₄, VCCGTX₁₅₅, VCCGTX₁₅₆, VCCGTX₁₅₇, VCCGTX₁₅₈, VCCGTX₁₅₉, VCCGTX₁₆₀, VCCGTX₁₆₁, VCCGTX₁₆₂, VCCGTX₁₆₃, VCCGTX₁₆₄, VCCGTX₁₆₅, VCCGTX₁₆₆, VCCGTX₁₆₇, VCCGTX₁₆₈, VCCGTX₁₆₉, VCCGTX₁₇₀, VCCGTX₁₇₁, VCCGTX₁₇₂, VCCGTX₁₇₃, VCCGTX₁₇₄, VCCGTX₁₇₅, VCCGTX₁₇₆, VCCGTX₁₇₇, VCCGTX₁₇₈, VCCGTX₁₇₉, VCCGTX₁₈₀, VCCGTX₁₈₁, VCCGTX₁₈₂, VCCGTX₁₈₃, VCCGTX₁₈₄, VCCGTX₁₈₅, VCCGTX₁₈₆, VCCGTX₁₈₇, VCCGTX₁₈₈, VCCGTX₁₈₉, VCCGTX₁₉₀, VCCGTX₁₉₁, VCCGTX₁₉₂, VCCGTX₁₉₃, VCCGTX₁₉₄, VCCGTX₁₉₅, VCCGTX₁₉₆, VCCGTX₁₉₇, VCCGTX₁₉₈, VCCGTX₁₉₉, VCCGTX₂₀₀, VCCGTX₂₀₁, VCCGTX₂₀₂, VCCGTX₂₀₃, VCCGTX₂₀₄, VCCGTX₂₀₅, VCCGTX₂₀₆, VCCGTX₂₀₇, VCCGTX₂₀₈, VCCGTX₂₀₉, VCCGTX₂₁₀, VCCGTX₂₁₁, VCCGTX₂₁₂, VCCGTX₂₁₃, VCCGTX₂₁₄, VCCGTX₂₁₅, VCCGTX₂₁₆, VCCGTX₂₁₇, VCCGTX₂₁₈, VCCGTX₂₁₉, VCCGTX₂₂₀, VCCGTX₂₂₁, VCCGTX₂₂₂, VCCGTX₂₂₃, VCCGTX₂₂₄, VCCGTX₂₂₅, VCCGTX₂₂₆, VCCGTX₂₂₇, VCCGTX₂₂₈, VCCGTX₂₂₉, VCCGTX₂₃₀, VCCGTX₂₃₁, VCCGTX₂₃₂, VCCGTX₂₃₃, VCCGTX₂₃₄, VCCGTX₂₃₅, VCCGTX₂₃₆, VCCGTX₂₃₇, VCCGTX₂₃₈, VCCGTX₂₃₉, VCCGTX₂₄₀, VCCGTX₂₄₁, VCCGTX₂₄₂, VCCGTX₂₄₃, VCCGTX₂₄₄, VCCGTX₂₄₅, VCCGTX₂₄₆, VCCGTX₂₄₇, VCCGTX₂₄₈, VCCGTX₂₄₉, VCCGTX₂₅₀, VCCGTX₂₅₁, VCCGTX₂₅₂, VCCGTX₂₅₃, VCCGTX₂₅₄, VCCGTX₂₅₅, VCCGTX₂₅₆, VCCGTX₂₅₇, VCCGTX₂₅₈, VCCGTX₂₅₉, VCCGTX₂₆₀, VCCGTX₂₆₁, VCCGTX₂₆₂, VCCGTX₂₆₃, VCCGTX₂₆₄, VCCGTX₂₆₅, VCCGTX₂₆₆, VCCGTX₂₆₇, VCCGTX₂₆₈, VCCGTX₂₆₉, VCCGTX₂₇₀, VCCGTX₂₇₁, VCCGTX₂₇₂, VCCGTX₂₇₃, VCCGTX₂₇₄, VCCGTX₂₇₅, VCCGTX₂₇₆, VCCGTX₂₇₇, VCCGTX₂₇₈, VCCGTX₂₇₉, VCCGTX₂₈₀, VCCGTX₂₈₁, VCCGTX₂₈₂, VCCGTX₂₈₃, VCCGTX₂₈₄, VCCGTX₂₈₅, VCCGTX₂₈₆, VCCGTX₂₈₇, VCCGTX₂₈₈, VCCGTX₂₈₉, VCCGTX₂₉₀, VCCGTX₂₉₁, VCCGTX₂₉₂, VCCGTX₂₉₃, VCCGTX₂₉₄, VCCGTX₂₉₅, VCCGTX₂₉₆, VCCGTX₂₉₇, VCCGTX₂₉₈, VCCGTX₂₉₉, VCCGTX₃₀₀, VCCGTX₃₀₁, VCCGTX₃₀₂, VCCGTX₃₀₃, VCCGTX₃₀₄, VCCGTX₃₀₅, VCCGTX₃₀₆, VCCGTX₃₀₇, VCCGTX₃₀₈, VCCGTX₃₀₉, VCCGTX₃₁₀, VCCGTX₃₁₁, VCCGTX₃₁₂, VCCGTX₃₁₃, VCCGTX₃₁₄, VCCGTX₃₁₅, VCCGTX₃₁₆, VCCGTX₃₁₇, VCCGTX₃₁₈, VCCGTX₃₁₉, VCCGTX₃₂₀, VCCGTX₃₂₁, VCCGTX₃₂₂, VCCGTX₃₂₃, VCCGTX₃₂₄, VCCGTX₃₂₅, VCCGTX₃₂₆, VCCGTX₃₂₇, VCCGTX₃₂₈, VCCGTX₃₂₉, VCCGTX₃₃₀, VCCGTX₃₃₁, VCCGTX₃₃₂, VCCGTX₃₃₃, VCCGTX₃₃₄, VCCGTX₃₃₅, VCCGTX₃₃₆, VCCGTX₃₃₇, VCCGTX₃₃₈, VCCGTX₃₃₉, VCCGTX₃₄₀, VCCGTX₃₄₁, VCCGTX₃₄₂, VCCGTX₃₄₃, VCCGTX₃₄₄, VCCGTX₃₄₅, VCCGTX₃₄₆, VCCGTX₃₄₇, VCCGTX₃₄₈, VCCGTX₃₄₉, VCCGTX₃₅₀, VCCGTX₃₅₁, VCCGTX₃₅₂, VCCGTX₃₅₃, VCCGTX₃₅₄, VCCGTX₃₅₅, VCCGTX₃₅₆, VCCGTX₃₅₇, VCCGTX₃₅₈, VCCGTX₃₅₉, VCCGTX₃₆₀, VCCGTX₃₆₁, VCCGTX₃₆₂, VCCGTX₃₆₃, VCCGTX₃₆₄, VCCGTX₃₆₅, VCCGTX₃₆₆, VCCGTX₃₆₇, VCCGTX₃₆₈, VCCGTX₃₆₉, VCCGTX₃₇₀, VCCGTX₃₇₁, VCCGTX₃₇₂, VCCGTX₃₇₃, VCCGTX₃₇₄, VCCGTX₃₇₅, VCCGTX₃₇₆, VCCGTX₃₇₇, VCCGTX₃₇₈, VCCGTX₃₇₉, VCCGTX₃₈₀, VCCGTX₃₈₁, VCCGTX₃₈₂, VCCGTX₃₈₃, VCCGTX₃₈₄, VCCGTX₃₈₅, VCCGTX₃₈₆, VCCGTX₃₈₇, VCCGTX₃₈₈, VCCGTX₃₈₉, VCCGTX₃₉₀, VCCGTX₃₉₁, VCCGTX₃₉₂, VCCGTX₃₉₃, VCCGTX₃₉₄, VCCGTX₃₉₅, VCCGTX₃₉₆, VCCGTX₃₉₇, VCCGTX₃₉₈, VCCGTX₃₉₉, VCCGTX₄₀₀, VCCGTX₄₀₁, VCCGTX₄₀₂, VCCGTX₄₀₃, VCCGTX₄₀₄, VCCGTX₄₀₅, VCCGTX₄₀₆, VCCGTX₄₀₇, VCCGTX₄₀₈, VCCGTX₄₀₉, VCCGTX₄₁₀, VCCGTX₄₁₁, VCCGTX₄₁₂, VCCGTX₄₁₃, VCCGTX₄₁₄, VCCGTX₄₁₅, VCCGTX₄₁₆, VCCGTX₄₁₇, VCCGTX₄₁₈, VCCGTX₄₁₉, VCCGTX₄₂₀, VCCGTX₄₂₁, VCCGTX₄₂₂, VCCGTX₄₂₃, VCCGTX₄₂₄, VCCGTX₄₂₅, VCCGTX₄₂₆, VCCGTX₄₂₇, VCCGTX₄₂₈, VCCGTX₄₂₉, VCCGTX₄₃₀, VCCGTX₄₃₁, VCCGTX₄₃₂, VCCGTX₄₃₃, VCCGTX₄₃₄, VCCGTX₄₃₅, VCCGTX₄₃₆, VCCGTX₄₃₇, VCCGTX₄₃₈, VCCGTX₄₃₉, VCCGTX₄₄₀, VCCGTX₄₄₁, VCCGTX₄₄₂, VCCGTX₄₄₃, VCCGTX₄₄₄, VCCGTX₄₄₅, VCCGTX₄₄₆, VCCGTX₄₄₇, VCCGTX₄₄₈, VCCGTX₄₄₉, VCCGTX₄₅₀, VCCGTX₄₅₁, VCCGTX₄₅₂, VCCGTX₄₅₃, VCCGTX₄₅₄, VCCGTX₄₅₅, VCCGTX₄₅₆, VCCGTX₄₅₇, VCCGTX₄₅₈, VCCGTX₄₅₉, VCCGTX₄₆₀, VCCGTX₄₆₁, VCCGTX₄₆₂, VCCGTX₄₆₃, VCCGTX₄₆₄, VCCGTX₄₆₅, VCCGTX₄₆₆, VCCGTX₄₆₇, VCCGTX₄₆₈, VCCGTX₄₆₉, VCCGTX₄₇₀, VCCGTX₄₇₁, VCCGTX₄₇₂, VCCGTX₄₇₃, VCCGTX₄₇₄, VCCGTX₄₇₅, VCCGTX₄₇₆, VCCGTX₄₇₇, VCCGTX₄₇₈, VCCGTX₄₇₉, VCCGTX₄₈₀, VCCGTX₄₈₁, VCCGTX₄₈₂, VCCGTX₄₈₃, VCCGTX₄₈₄, VCCGTX₄₈₅, VCCGTX₄₈₆, VCCGTX₄₈₇, VCCGTX₄₈₈, VCCGTX₄₈₉, VCCGTX₄₉₀, VCCGTX₄₉₁, VCCGTX₄₉₂, VCCGTX₄₉₃, VCCGTX₄₉₄, VCCGTX₄₉₅, VCCGTX₄₉₆, VCCGTX₄₉₇, VCCGTX₄₉₈, VCCGTX₄₉₉, VCCGTX₅₀₀, VCCGTX₅₀₁, VCCGTX₅₀₂, VCCGTX₅₀₃, VCCGTX₅₀₄, VCCGTX₅₀₅, VCCGTX₅₀₆, VCCGTX₅₀₇, VCCGTX₅₀₈, VCCGTX₅₀₉, VCCGTX₅₁₀, VCCGTX₅₁₁, VCCGTX₅₁₂, VCCGTX₅₁₃, VCCGTX₅₁₄, VCCGTX₅₁₅, VCCGTX₅₁₆, VCCGTX₅₁₇, VCCGTX₅₁₈, VCCGTX₅₁₉, VCCGTX₅₂₀, VCCGTX₅₂₁, VCCGTX₅₂₂, VCCGTX₅₂₃, VCCGTX₅₂₄, VCCGTX₅₂₅, VCCGTX₅₂₆, VCCGTX₅₂₇, VCCGTX₅₂₈, VCCGTX₅₂₉, VCCGTX₅₃₀, VCCGTX₅₃₁, VCCGTX₅₃₂, VCCGTX₅₃₃, VCCGTX₅₃₄, VCCGTX₅₃₅, VCCGTX₅₃₆, VCCGTX₅₃₇, VCCGTX₅₃₈, VCCGTX₅₃₉, VCCGTX₅₄₀, VCCGTX₅₄₁, VCCGTX₅₄₂, VCCGTX₅₄₃, VCCGTX₅₄₄, VCCGTX₅₄₅, VCCGTX₅₄₆, VCCGTX₅₄₇, VCCGTX₅₄₈, VCCGTX₅₄₉, VCCGTX₅₅₀, VCCGTX₅₅₁, VCCGTX₅₅₂, VCCGTX₅₅₃, VCCGTX₅₅₄, VCCGTX₅₅₅, VCCGTX₅₅₆, VCCGTX₅₅₇, VCCGTX₅₅₈, VCCGTX₅₅₉, VCCGTX₅₆₀, VCCGTX₅₆₁, VCCGTX₅₆₂, VCCGTX₅₆₃, VCCGTX₅₆₄, VCCGTX₅₆₅, VCCGTX₅₆₆, VCCGTX₅₆₇, VCCGTX₅₆₈, VCCGTX₅₆₉, VCCGTX₅₇₀, VCCGTX₅₇₁, VCCGTX₅₇₂, VCCGTX₅₇₃, VCCGTX₅₇₄, VCCGTX₅₇₅, VCCGTX₅₇₆, VCCGTX₅₇₇, VCCGTX₅₇₈, VCCGTX₅₇₉, VCCGTX₅₈₀, VCCGTX₅₈₁, VCCGTX₅₈₂, VCCGTX₅₈₃, VCCGTX₅₈₄, VCCGTX₅₈₅, VCCGTX₅₈₆, VCCGTX₅₈₇, VCCGTX₅₈₈, VCCGTX₅₈₉, VCCGTX₅₉₀, VCCGTX₅₉₁, VCCGTX₅₉₂, VCCGTX₅₉₃, VCCGTX₅₉₄, VCCGTX₅₉₅, VCCGTX₅₉₆, VCCGTX₅₉₇, VCCGTX₅₉₈, VCCGTX₅₉₉, VCCGTX₆₀₀, VCCGTX₆₀₁, VCCGTX₆₀₂, VCCGTX₆₀₃, VCCGTX₆₀₄, VCCGTX₆₀₅, VCCGTX₆₀₆, VCCGTX₆₀₇, VCCGTX₆₀₈, VCCGTX₆₀₉, VCCGTX₆₁₀, VCCGTX₆₁₁, VCCGTX₆₁₂, VCCGTX₆₁₃, VCCGTX₆₁₄, VCCGTX₆₁₅, VCCGTX₆₁₆, VCCGTX₆₁₇, VCCGTX₆₁₈, VCCGTX₆₁₉, VCCGTX₆₂₀, VCCGTX₆₂₁, VCCGTX₆₂₂, VCCGTX₆₂₃, VCCGTX₆₂₄, VCCGTX₆₂₅, VCCGTX₆₂₆, VCCGTX₆₂₇, VCCGTX₆₂₈, VCCGTX₆₂₉, VCCGTX₆₃₀, VCCGTX₆₃₁, VCCGTX₆₃₂, VCCGTX₆₃₃, VCCGTX₆₃₄, VCCGTX₆₃₅, VCCGTX₆₃₆, VCCGTX₆₃₇, VCCGTX₆₃₈, VCCGTX₆₃₉, VCCGTX₆₄₀, VCCGTX₆₄₁, VCCGTX₆₄₂, VCCGTX₆₄₃, VCCGTX₆₄₄, VCCGTX₆₄₅, VCCGTX₆₄₆, VCCGTX₆₄₇, VCCGTX₆₄₈, VCCGTX₆₄₉, VCCGTX₆₅₀, VCCGTX₆₅₁, VCCGTX₆₅₂, VCCGTX₆₅₃, VCCGTX₆₅₄, VCCGTX₆₅₅, VCCGTX₆₅₆, VCCGTX₆₅₇, VCCGTX₆₅₈, VCCGTX₆₅₉, VCCGTX₆₆₀, VCCGTX₆₆₁, VCCGTX₆₆₂, VCCGTX₆₆₃, VCCGTX₆₆₄, VCCGTX₆₆₅, VCCGTX₆₆₆, VCCGTX₆₆₇, VCCGTX₆₆₈, VCCGTX₆₆₉, VCCGTX₆₇₀, VCCGTX₆₇₁, VCCGTX₆₇₂, VCCGTX₆₇₃, VCCGTX₆₇₄, VCCGTX₆₇₅, VCCGTX₆₇₆, VCCGTX₆₇₇, VCCGTX₆₇₈, VCCGTX₆₇₉, VCCGTX₆₈₀, VCCGTX₆₈₁, VCCGTX₆₈₂, VCCGTX₆₈₃, VCCGTX₆₈₄, VCCGTX₆₈₅, VCCGTX₆₈₆, VCCGTX₆₈₇, VCCGTX₆₈₈, VCCGTX₆₈₉, VCCGTX₆₉₀, VCCGTX₆₉₁, VCCGTX₆₉₂, VCCGTX₆₉₃, VCCGTX₆₉₄, VCCGTX₆₉₅, VCCGTX₆₉₆, VCCGTX₆₉₇, VCCGTX₆₉₈, VCCGTX₆₉₉, VCCGTX₇₀₀, VCCGTX₇₀₁, VCCGTX₇₀₂, VCCGTX₇₀₃, VCCGTX₇₀₄, VCCGTX₇₀₅, VCCGTX₇₀₆, VCCGTX₇₀₇, VCCGTX₇₀₈, VCCGTX₇₀₉, VCCGTX₇₁₀, VCCGTX₇₁₁, VCCGTX₇₁₂, VCCGTX₇₁₃, VCCGTX₇₁₄, VCCGTX₇₁₅, VCCGTX₇₁₆, VCCGTX₇₁₇, VCCGTX₇₁₈, VCCGTX

```

library ieee;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity voltage_monitor is
generic (low_voltage_01mv:integer range 0 to 33000 := 10000; -- low threshold of voltage hysteresis
         high_voltage_01mv:integer range 0 to 33000 := 20000); -- high threshold of voltage hysteresis
port (
    adc_cmd_valid:          out std_logic;
    adc_cmd_channel:        out std_logic_vector (4 downto 0);
    adc_cmd_startofpacket:  out std_logic; --altera_adc_control IP ignores this signal
    adc_cmd_endofpacket:   out std_logic; --altera_adc_control IP ignores this signal
    adc_sink_rst_n:         out std_logic;
    mainpwr_ok:             out std_logic; -- RSMRST# output to PCH
-- v5_en:                  out std_logic;
    adc_cmd_ready:          in std_logic;
    adc_per_clk:            in std_logic;
    adc_res_valid:          in std_logic; -- ADC sampling rate: 500KHz
    adc_res_channel:        in std_logic_vector (4 downto 0);
    adc_res_data:           in std_logic_vector (11 downto 0); --for 3.3V ref, the LSB =~ 0.806 mV = 806 uV =~ 0.8mV
    adc_res_startofpacket:  in std_logic; --altera_adc_control IP just passes from the corresponding command signal
    adc_res_endofpacket:   in std_logic); --altera_adc_control IP just passes from the corresponding command signal
end voltage_monitor;

architecture voltage_monitor_arch of voltage_monitor is
type state_type is (counting_low, counting_high, state_vol_ok, state_vol_low);
attribute enum_encoding : string;
attribute enum_encoding of state_type : type is "01 10 11 00"; --<< state_vol_low is default after FPGA power-on
type main_state_type is (main_ok, main_fail, main_low);
attribute enum_encoding2 : string;
attribute enum_encoding2 of state_type : type is "01 10 00"; --<< main_low is default after FPGA power-on
signal main_curr_state : main_state_type := main_low;
signal curr_state: state_type := state_vol_low;
signal next_state: state_type := state_vol_low;
signal count : unsigned(23 downto 0):=(others => '0');
signal main_count : unsigned(23 downto 0):=(others => '0');
signal low_vol_th_01mv : unsigned(15 downto 0);
signal high_vol_th_01mv : unsigned(15 downto 0);
signal vol_ok: std_logic := '0';

signal voltage_01mv : unsigned(15 downto 0);
begin

adc_cmd_valid <= '1';
adc_cmd_channel <= "00001"; -- ADC channel 1
adc_cmd_startofpacket <= '1'; --altera_adc_control IP ignores this signal
adc_cmd_endofpacket <= '0'; --altera_adc_control IP ignores this signal
adc_sink_rst_n <= '1';

process (adc_res_valid) -- ADC data processing
begin
    if (adc_res_valid = '1') then
        voltage_01mv <= to_unsigned((TO_INTEGER("0000" & unsigned(adc_res_data)))*8,16);-- << *16 only for evaluation board. The
        analog signal is divided by 2 using resistors. Normaly use *8.
    end if;
end process;

low_vol_th_01mv <= to_unsigned(low_voltage_01mv, 16);
high_vol_th_01mv <= to_unsigned(high_voltage_01mv, 16);
curr_state <= next_state;

process (adc_res_valid) -- ADC data to PWR_OK signal control process. Mealy state machine.
begin
    if (adc_res_valid = '0') then --- on falling edge, in order to allow voltage_01mv to be calculated
        case curr_state is
            when counting_low =>
                if ( voltage_01mv < low_vol_th_01mv) then
                    if (count = to_unsigned(5,24)) then
                        count <= (others => '0'); -- 5 low samples in a row
                        VOL_OK <= '0';
                        next_state <= state_vol_low;
                    else
                        count <= count + 1;
                        next_state <= counting_low;
                    end if;
                else
                    -- less than 5 low samples in a row -> back to vol OK state
                    VOL_OK <= '1';
                    count <= (others => '0');
                    next_state <= state_vol_ok;
                end if;
            when counting_high =>
                if ( voltage_01mv > high_vol_th_01mv) then
                    if (count = to_unsigned(1000,24)) then -- 5000 high samples in a row = 10msec -- ADC sampling rate:
500KHz
                        count <= (others => '0');
                        VOL_OK <= '1';
                        next_state <= state_vol_ok;
                    else
                        count <= count + 1;
                        next_state <= counting_high;
                    end if;
                end if;
        end case;
    end if;
end process;

```

```

        end if;
    else
        VOL_OK <= '0';
        count <= (others => '0');
        next_state <= state_vol_low;
    end if;

when state_vol_low =>
    if ( voltage_01mv > high_vol_th_01mv) then -- first sample high
        count <= count + 1;
        next_state <= counting_high;
    else
        VOL_OK <= '0';
        count <= (others => '0');
        next_state <= state_vol_low;
    end if;

when state_vol_ok =>
    if ( voltage_01mv < low_vol_th_01mv) then -- first sample low
        count <= count + 1;
        next_state <= counting_low;
    else
        VOL_OK <= '1';
        count <= (others => '0');
        next_state <= state_vol_ok;
    end if;

when others =>
    next_state <= state_vol_ok;

end case;
end if;
end process;

process (adc_res_valid) -- Main process, dealing with RSMRST# and VRs enable ping -- ADC sampling rate: 500KHz
begin
    if (adc_res_valid = '1') then
        case main_curr_state is

when main_ok =>
    if (vol_ok = '0') then
        main_curr_state <= main_fail;
        main_count <= (others => '0');
    else
        main_curr_state <= main_ok;
    end if;
--    v5_en <= '1';
--    mainpwr_ok <= '1';

when main_fail => -- According to Skylake / Kabylake PDG: a minimum of 1 uSec between rsmrst# assertion to power
loss.
    if (main_count = to_unsigned(100,24)) then --20mSec => 10msec -- ADC sampling rate: 500KHz
        main_curr_state <= main_low;
        main_count <= (others => '0');
    else
        main_count <= main_count + 1;
        main_curr_state <= main_fail;
    end if;
--    v5_en <= '1';
--    mainpwr_ok <= '0';

when main_low =>
    if (vol_ok = '1') then
        main_curr_state <= main_ok;
    else
        main_curr_state <= main_low;
    end if;
--    v5_en <= '0';
--    mainpwr_ok <= '0';

end case;
end if;
end process;
end voltage_monitor_arch;

```

```

library ieee;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

-- SLP_S3# asserted with or before PCH_PWROK
-- VCCST_PWRGD is derived from PG of the CPU's VR, with a delay at rising edge after SLP_S3# deassertion of minimum 1 msec (actual 3 msec).
-- VCCST_PWRGD is tied to PCH_PWROK, and should have a hardware resistive divider, to be at 1V domain (CPU input).
-- SLP_S3# assertion to VCCST_PWRGD de-assertion: maximum of 1 usec.
-- VR_VSA_READY is a signal to EC. in the CRB it is connected to VR_READY.

entity pch_pwrok_block is
port (
    slp_s3:          in std_logic; -- SLP_S3#
    vr_ready:         in std_logic; -- Open-drain, internal weak pull-up required
    vccsa_ready:      in std_logic; -- Open-drain, internal weak pull-up required
    clk_100k:         in std_logic; -- 100KHz clock, T = 10uSec
    vccst_pwrqd_3v3: out std_logic;
    pch_pwrok:        out std_logic);
end pch_pwrok_block;

architecture pch_pwrok_block_arch of pch_pwrok_block is
type state_type is (pwrqd, no_pwrqd, delay);
attribute enum_encoding : string;
attribute enum_encoding of state_type : type is "01 00 10"; --<> no_pwrqd is default after FPGA power
signal curr_state: state_type := no_pwrqd;
signal delayed_vr_vccsa_ok: std_logic := '0';
signal vr_vccsa_ok: std_logic;
signal count : unsigned(15 downto 0) := (others => '0');
begin

vr_vccsa_ok <= '1' when (vr_ready = '1') and (vccsa_ready = '1') and (slp_s3 = '1') -- Delay trigger
else '0';

 <= '1' when (delayed_vr_vccsa_ok = '1') and (slp_s3 = '1') -- Output
else '0';

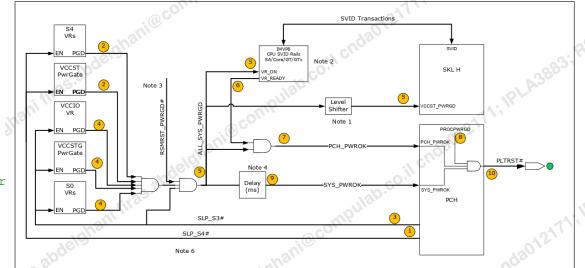
vccst_pwrqd_3v3 <= '1' when (delayed_vr_vccsa_ok = '1') and (slp_s3 = '1')
else '0';

process (clk_100k) -- 5 mSec delay process, delay at pwrok rising edge: vr_vccsa_ok -> delayed_vr_vccsa_ok
begin
    if (clk_100k = '1') then
        case curr_state is
            when pwrqd =>
                if (vr_vccsa_ok = '1') then
                    curr_state <= pwrqd;
                    delayed_vr_vccsa_ok <= '1';
                else
                    curr_state <= no_pwrqd; -- short delay at vr_vccsa ok transition from 1 to 0
                    delayed_vr_vccsa_ok <= '0'; -- delayed vr_vccsa ok signal will not assert at vr_vccsa ok glitches of 1T
                    end if;
            when delay =>
                if (count = to unsigned(3000,16)) then -- 300 * 10uSec = 3 mSec (Changed to 3000: Test)
                    curr_state <= pwrqd;
                    count <= (others => '0');
                else
                    count <= count + 1;
                    curr_state <= delay;
                end if;
                delayed_vr_vccsa_ok <= '0';
            when no_pwrqd =>
                if (vr_vccsa_ok = '1') then
                    curr_state <= delay; -- transition to high can be done without a delay (SLP_S4# is already high)
                    count <= (others => '0');
                    delayed_vr_vccsa_ok <= '0';
                else
                    curr_state <= no_pwrqd;
                end if;
            end case;
        end if;
    end process;
end pch_pwrok_block_arch;

```

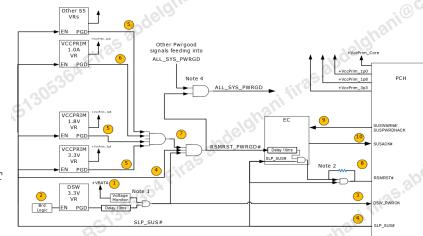
45.2 Sequencing Interface Signals List and Power Rails

Figure 45-1. CFL Flow Diagram for SYS_PWROK/PCH_PWROK Generation



- Notes:**
1. Enable inputs may be qualified by other signals and factors not shown.
 2. Optional configurations. Different configurations may have impact boot latencies differently.
 3. This delay is not explicitly required. For systems supporting traditional PCIe add-in cards that require up to 100ms delay from more or less time is required, then a specific timing delay circuit may be required.
 4. Voltage Monitor circuit to ensure PSU pwr rails are mostly up and stable before the VR is enabled.
 5. Critical based on the system design to ensure PCH_PWROK rising edge is not used as a qualifier to gate PLTRST# deassertion via SYS_PWROK assertion. PSU power rail stability is inferred on the platform thru other means, such as the VM block shown in the diagram.
 6. VCCST_PWRGD must go low during 5x pwr states, regardless of the voltage level of VCCST

Many variations exist on how the various pwyod signals can be generated on a DT platform to balance system robustness with platform cost constraints. This is just one example of possible logic connections. Components and connections in blue are alternative configurations. Components and connection in black show a baseline configuration.



**Table 45-1. CFL Interface Signals List (Sheet 4 of 4)**

Name	Source	Destination	Description
CPU_C10_GATE#	CPU	Platform	Power gating control to turn off VCCSTG, VCCIO and VCCPLL_OC in C10

Table 45-2. CFL Power Sequence Related Power Rails

Name	Source	Destination	Description
VCCRTC	Platform	PCH	3.05-V supplies for PCH RTC Well. This power is not expected to be shut off in any of the sleep states unless the RTC battery is removed or completely drained.
VCCDSW_3p3	Platform	PCH	3.3-V supply for Deep Sx wells. If Deep Sx is not supported on the platform, tie to VCCPRIM_3p3
VCCPRIM_1p0/1p8/3p3	Platform	PCH	PCH I/O and Misc rails 1.0/1.8/3.3V (Primary Well)
VCCSPI	Platform	PCH	1.8V/3.3-V supply for the SPI IO. This rail must be powered when VCCPRIM is powered.
VCC	Platform	Processor	Processor core rail
VCCST	Platform	Processor	Sustain voltage for processor in Standby modes
VCCPLL	Platform	Processor	CPU PLL power rails
VCCPLL_OC	Platform	Processor	CPU digital PLL power rails
VCCGT	Platform	Processor	Sliced graphics power rail
VCCGTx ¹	Platform	Processor	Unsliced graphics power rail
VCCIO	Platform	Processor	IO power rail
VDDQ	Platform	Processor	CPU Memory power rail, voltage dependent on memory technology
VPP	Platform	Processor	CPU Memory power rail, voltage dependent on memory technology
VCCSA	Platform	Processor	System Agent power rail

45.3 Power States

CLH - PDG

Table 45-3. System with M3 State Supported (Sheet 1 of 2)

Rails	S0/M0	S0ix/ M-off ¹⁶	S3/M3	S3/M- off	S4 & S5/M3	S4 & S5/ M-off	Deep S3	Deep S4/ S5	G3 ¹
RTC Well	ON	ON	ON	ON	ON	ON	ON	ON	ON
3.3V_DSW	ON	ON	ON	ON	ON	ON	ON	ON	No Power
VBATA (VDC)	ON	ON	ON	ON	ON	ON	ON	ON	No Power
V5.0A	ON	ON	ON	ON	ON	ON	OFF	OFF	No Power
V3.3A	ON	ON	ON	ON	ON	ON	OFF	OFF	No Power
V1.8A	ON	ON	ON	ON	ON	ON	OFF	OFF	No Power
V1.0A	ON	ON	ON	ON	ON	ON	OFF	OFF	No Power
V3.3M ³	ON	OFF	ON ¹¹	OFF	ON ¹¹	OFF	OFF	OFF	No Power
V1.8M ³	ON	OFF	ON ¹¹	OFF	ON ¹¹	OFF	OFF	OFF	No Power
VDDQ	ON	ON	ON	ON	OFF	OFF	ON	OFF	No Power
V2.5U	ON	ON	ON	ON	OFF	OFF	ON	OFF	No Power
VCCST ¹³	ON	ON	ON ¹⁹	ON ¹⁹	OFF ⁵	OFF ⁵	OFF	OFF	No Power
VCCPLL	ON	ON	ON ^{7, 19}	ON ^{7, 19}	OFF ⁵	OFF ⁵	OFF ⁵	OFF	No Power

**Table 45-3. System with M3 State Supported (Sheet 2 of 2)**

Rails	S0/M0	S0ix/ M-off ¹⁶	S3/M3	S3/M- off	S4 & S5/M3	S4 & S5/ M-off	Deep S3	Deep S4/ S5	G3 ¹
V3.3S	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	No Power
VCCPLL_OC ¹³	ON	OFF ¹⁷	ON ^{8, 19}	ON ^{8, 19}	OFF	OFF	OFF ⁸	OFF	No Power
VCC	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	No Power
VCCGT/VCCGTx	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	No Power
VCCIO	ON	OFF ¹⁷	OFF	OFF	OFF	OFF	OFF	OFF	No Power
VCCSA	ON	OFF ¹⁷	OFF	OFF	OFF	OFF	OFF	OFF	No Power

Notes:

1. The state of the system without RTC well powered can also be considered G3.
2. NA
3. V3.3M and V1.8M are platform rails used by external devices which ME operates during Sx/M3 states. These rails are not used directly by the CPU/PCH, and are not present on non-M3 supported systems
4. NA
5. VCCST, VCCPLL can remain powered during S4 and S5 power states for board VP optimization. VCCST may also remain powered in S4 and S5 for debug purposes. Refer to [Chapter 46, "Platform Debug and Test Hooks"](#).
6. NA
7. VCCPLL is allowed to be OFF in this power state, but it is generally assumed to be ON since it is powered from the same source as VCCST. VCCPLL should never be ON while VCCST is OFF
8. VCCPLL_OC is allowed to be turned off during S3 if it is not powered directly from VDDQ
9. NA
10. For no M3 support on external devices, V3.3M/V1.8M will be OFF in Sx/M3
11. VCCOPC_1p8 may be left on in Sx with minimal leakage.
12. NA
13. Assume SLP_S0# and CPU_C10_GATE# have asserted from the PCH. This supply is expected to be OFF during states where SLP_S0# is asserted. It may be left on during this condition, but the SoC will not achieve its lowest power consumption. Specific power up latencies apply when exiting this state. VCCST should be "on" whenever VCCPLL_OC is "on". VCCPLL_OC must be "off" whenever VCCST is "off", pay special attention particularly for systems supporting DS3.
14. For additional power savings in S3, refer [Section 45.5, "Additional Power Optimizations with Respect to VCCST Rail in S3"](#)

45.4 Power Sequencing Timing Diagrams—Legacy Signals

Table 45-4. Legend for Signals in Transition Waveforms

Color/Legend	Comments
Signal Names	Timing of these signals is set by PCH or processor
Signal Names	Timing of these signals should be met by the platform
Signal Names	Voltage rails or chip-to-chip buses
Grey Highlight	Indicates unstable state

Figure 45-3. Timing Diagram for G3 to S0/M0 [Deep Sx Platform] (Sheet 1 of 2)

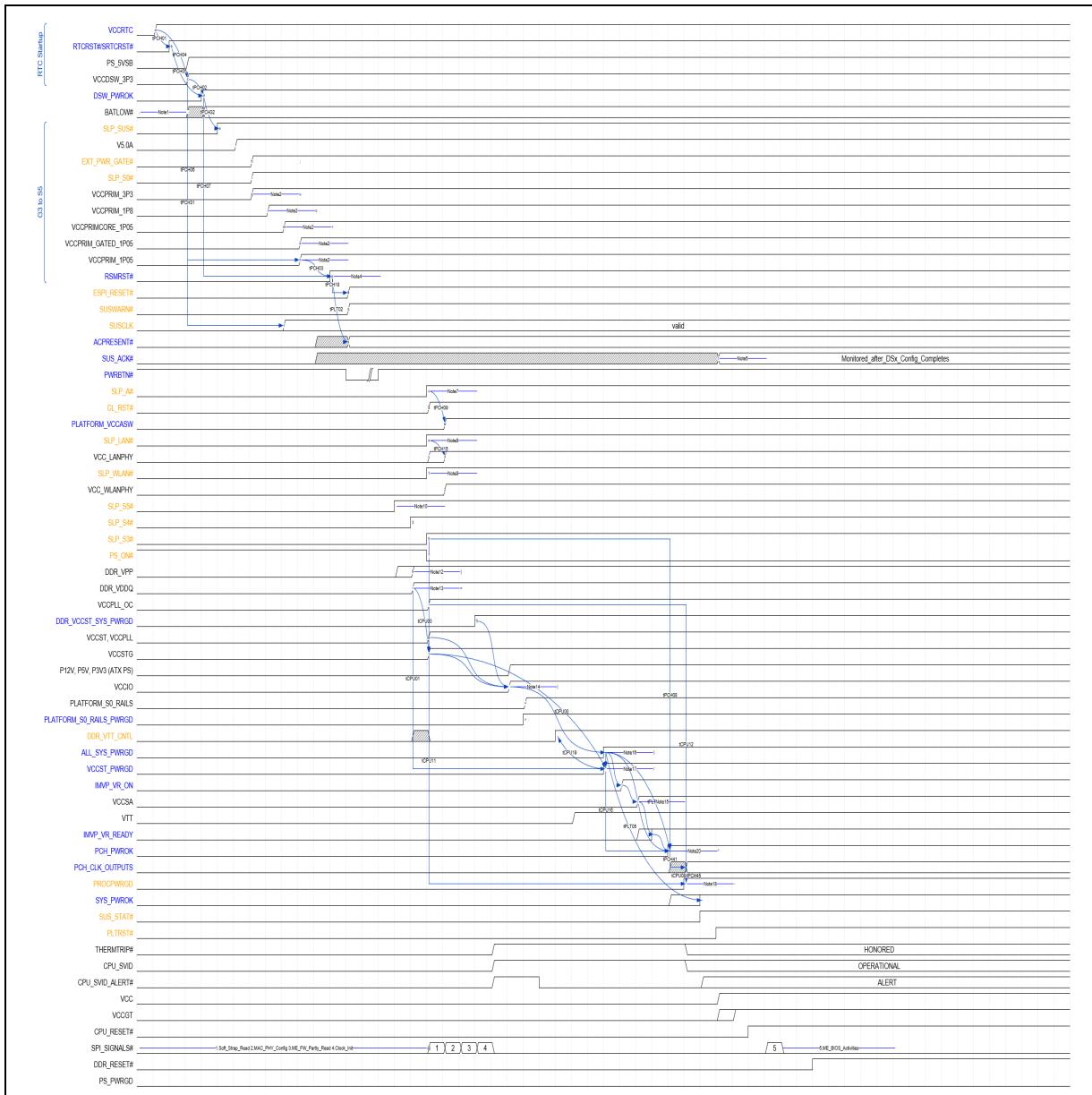


Figure 45-3. Timing Diagram for G3 to S0/M0 [Deep Sx Platform] (Sheet 2 of 2)**Notes:**

1. PCH will sample BATLOW# on the rising edge of DSW_PWROK for DSx Systems
2. Refer section TBD details on PCH prime rail-to-rail power and power down dependencies
3. For a DSx system, PCH will ignore SUSACK# until after SUSWARN# is valid
4. Minimum duration of PWRBTN# assertion = 16mS. PWRBTN# can assert before or after RSMRST# ✓
5. On first exit from G3, SLP_A# de-asserts with SLP_S3# de-assertion
6. High for WoL=1, Low for WoL=0. SLP_LAN# may rise before, but no later than SLP_A#
7. On first exit from G3, SLP_WLAN# de-asserts with SLP_S3# de-assertion
8. Delay between SLP_S5#, SLP_S4#, and SLP_S3# exaggerated for drawing purposes. If the system EC is driving these signals in ESPI mode if the, the minimum delay between SLP_S3#, SLP_S4#, and SLP_S5# is not guaranteed
9. VCCST and VCCPLL can remain powered during S4 and S5 pwr states for board VR optimization. VCCST may also remain powered in S4 and S5 for debug purposes. Refer to debug port design guide for more details.
10. Only required with LPDDR3 and DDR4 memory configurations
11. VDDQ must ramp after VPP on DDR4 and LPDDR3 based systems, thus VDDQ may ramp up after SLP_S3# de-assertion due to VR ramp timing and configuration
12. VCCIO, VCCSA must ramp after VccST and VDDQ have completed their ramps.
13. IMVP VR_ON is recommended to be triggered by ALL_SYS_PWRGD in order to help minimize boot latency.
14. ALL_SYS_PWRGD is assumed to logically AND together the pwrgood signals for the major system power rails
15. VCCST_PWRGD can assert before or equal to PCH_PWROK, but must never lag it. It is recommended that both VCCST_PWRGD and PCH_PWROK include ALL_SYS_PWRGD and / or SLP_S3# in their generation. This ensures during failure events, both signals de-assert at the same time
16. NA
17. When "Power Button" is the trigger for wake or sleep event for the system
18. The Platform should ensure that PCH_PWROK does not glitch when RSMRST# is de-asserted
19. PS has a wide range of specifications, which may affect boot latency
20. Use of DDR_VTT_CNTL to control VTT power gate is optional for additional power savings in Idle power states. If not used, VTT should be controlled by SLP_S3#

power up

Figure 45-4. Timing Diagram for G3 to S0/M0 [Non-Deep Sx Platform] (Sheet 1 of 2)

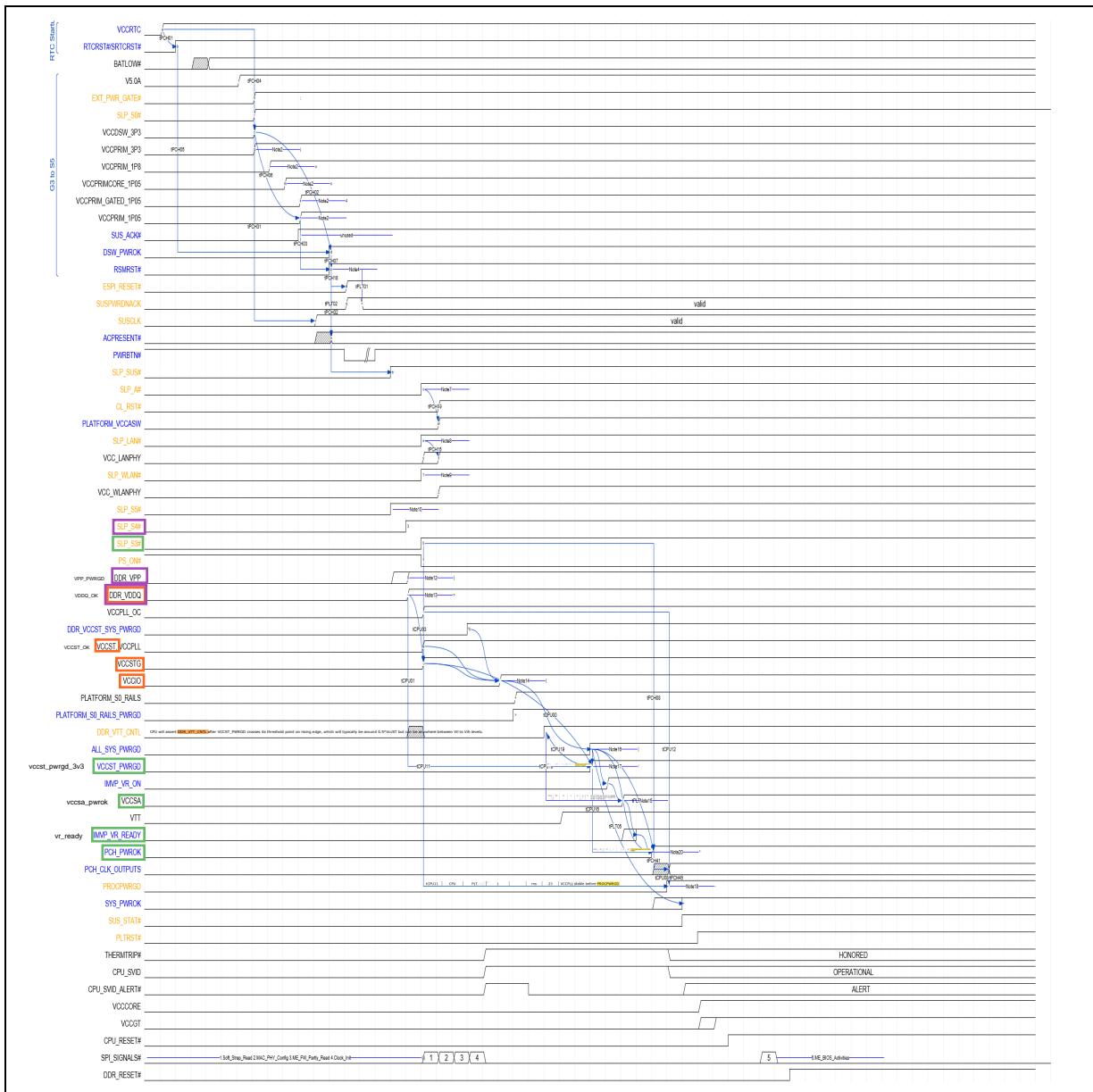


Figure 45-4. Timing Diagram for G3 to S0/M0 [Non-Deep Sx Platform] (Sheet 2 of 2)

Notes:

1. SLP_SUS# is ignored in Non-DSx systems
2. Refer section TBD for details on PCH prime rail-to-rail power and power down dependencies
3. For a non-DeepSx system DSW_PWROK and RSMRST# go high at the same time (connected on board)
4. For a non-DeepSx system SUS_ACK# will rise with prime voltage rail powering the VCCPGPPA power pin due to weak internal pull-up.
5. Minimum duration of PWRBTN# assertion = 16mS. PWRBTN# can assert before or after RSMRST#
6. On first exit from G3, SLP_A# de-asserts with SLP_S3# de-assertion
7. High for WoL=1, Low for WoL=0. SLP_LAN# may rise before, but no later than SLP_A#
8. On first exit from G3, SLP_WLAN# de-asserts with SLP_S3# de-assertion
9. Delay between SLP_S5#, SLP_S4#, and SLP_S3# exaggerated for drawing purposes. If the system EC is driving these signals in ESPI mode if the, the minimum delay between SLP_S3#, SLP_S4#, and SLP_S5# is not guaranteed
10. VCCST and VCCPLL can remain powered during S4 and S5 power states for board VR optimization. VCCST may also remain powered in S4 and S5 for debug purposes. Refer to debug port design guide for more details.
11. Only required with LPDDR3 and DDR4 memory configurations
12. **VDDQ must ramp after VPP on DDR4 and LPDDR3 based systems, thus VDDQ may ramp up after SLP_S3# de-assertion due to VR ramp timing and configuration**
13. **VCCIO, VCCSA must ramp after VccST and VDDQ have completed their ramps**
14. IMVP VR_ON is recommended to be triggered by ALL_SYS_PWRGD in order to help minimize boot latency.
15. ALL_SYS_PWRGD is assumed to logically AND together the pwrgood signals for the major system power rails
16. **VCCST_PWRGD can assert before or equal to PCH_PWROK, but must never lag it. It is recommended that both VCCST_PWRGD and PCH_PWROK include ALL_SYS_PWRGD and / or SLP_S3# in their generation. This ensures during failure events, both signals de-assert at the same time**
17. NA
18. When "Power Button" is the trigger for wake or sleep event for the system
19. The Platform should ensure that PCH_PWROK does not glitch when RSMRST# is de-asserted
20. PS has a wide range of specifications, which may affect boot latency
21. Use of DDR_VTT_CNTL to control VTT power gate is optional for additional power savings in Idle power states. If not used, VTT should be controlled by SLP_S3#

Additional Notes:

The state of the SLP_A# and SUSPWRDNACK signals are used by the EC to determine if PCH requires the suspend-well to stay powered.

- SUSPWRDNACK
 - *Platform not supporting M3* - EC must keep SUS Rails powered ON if: SUSPWRDNACK is de-asserted **OR** System state is S3. Else, EC has an option to do whatever it wants with the SUS Rails
 - *Platform supporting M3* - EC must keep SUS Rails powered ON if: SUSPWRDNACK is de-asserted **OR** System state is S3 **OR** SLP_A# is de-asserted **OR** it is the first 200mS after SUS Rails power has been applied. Else, EC has an option to do whatever it wants with the SUS Rails
- Primary rails and Deep Sx Rails should **never** be active while VccRTC rail is inactive.

Power Down

Figure 45-5. Timing Diagram for S0/M0 to G3 [Deep Sx Platform]

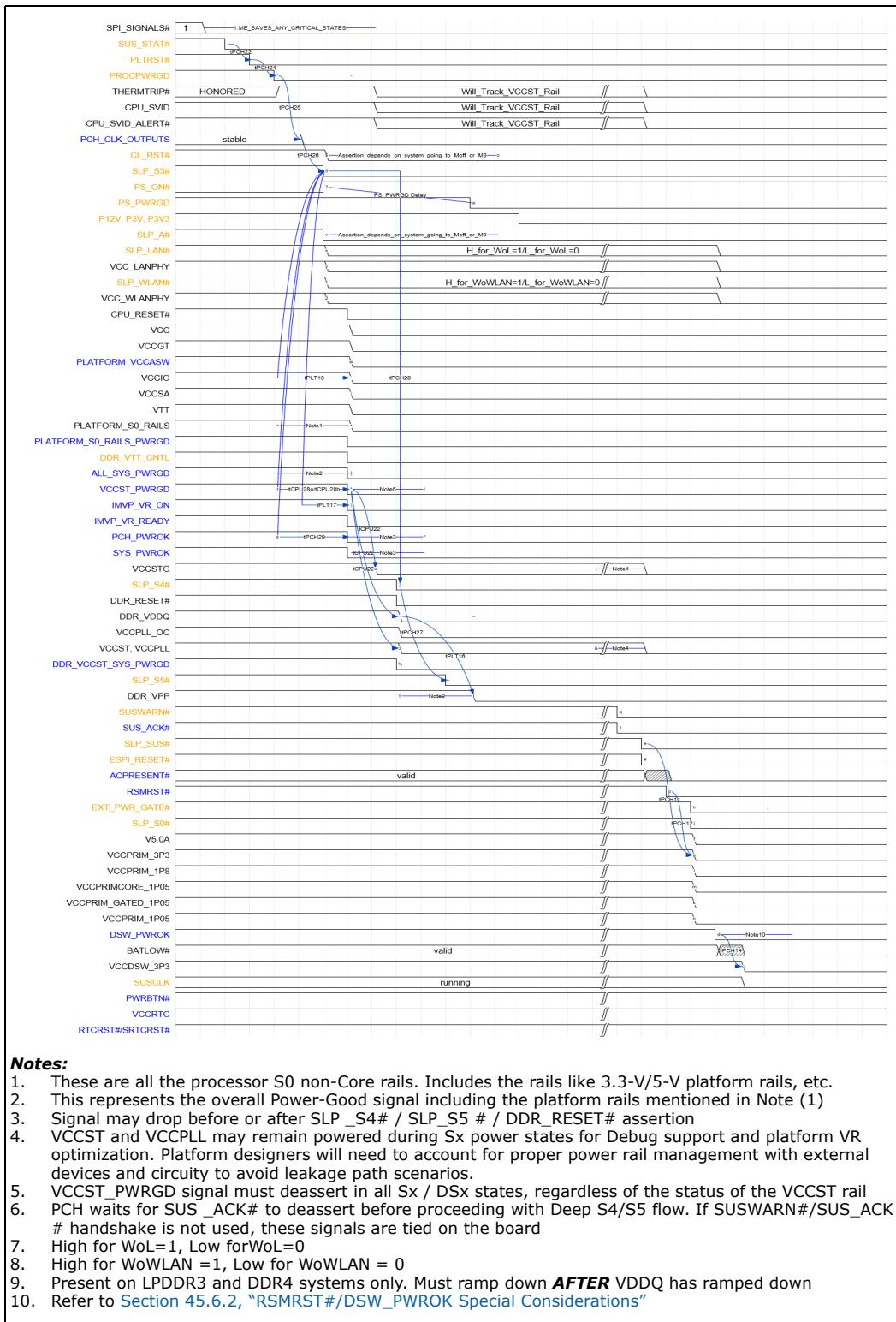


Figure 45-6. Timing Diagram for S0/M0 to G3 [Non-Deep Sx Platform]

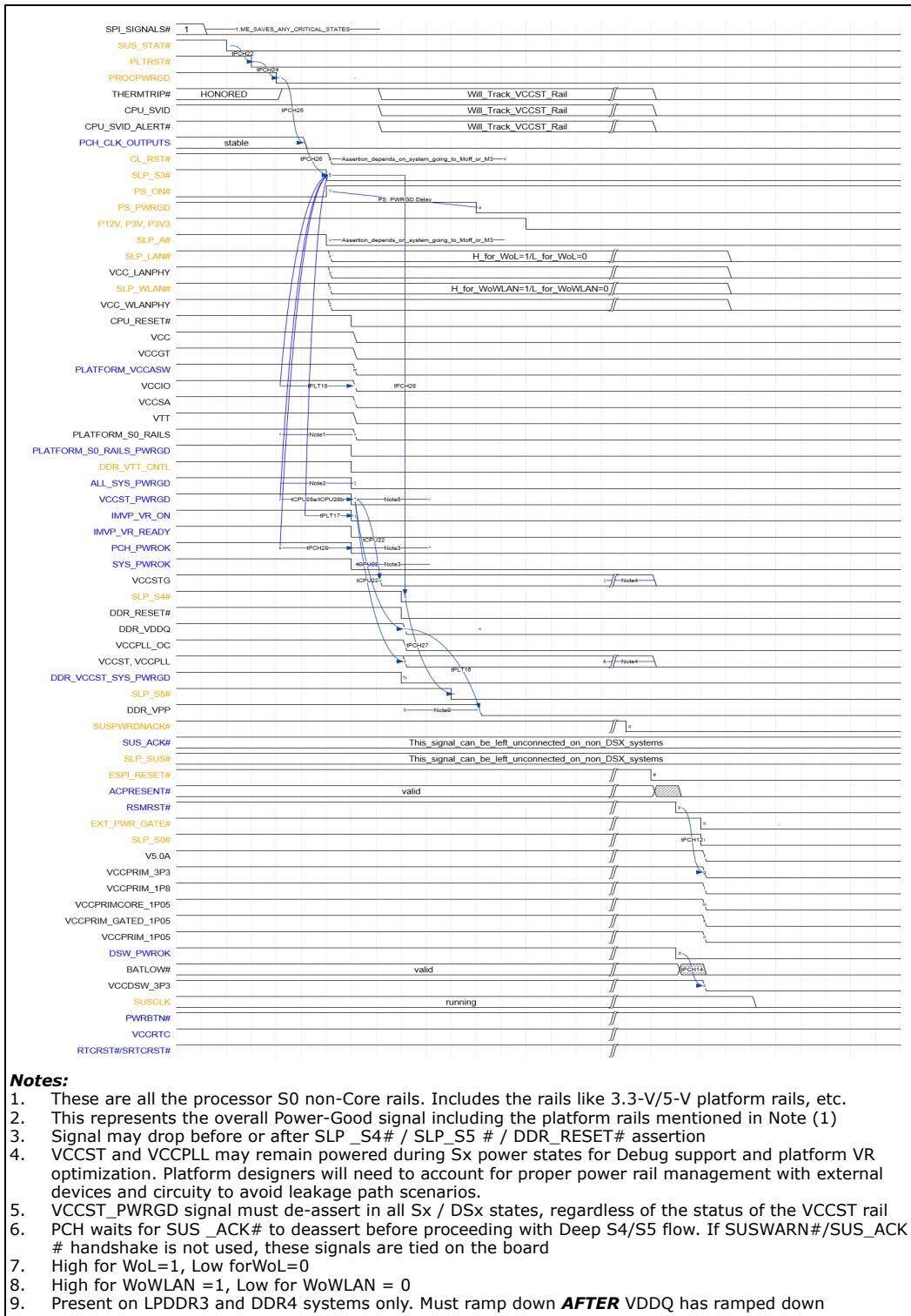


Figure 45-7. Timing Diagram for Warm Reset (Host Partition Reset w/o Power Cycle)

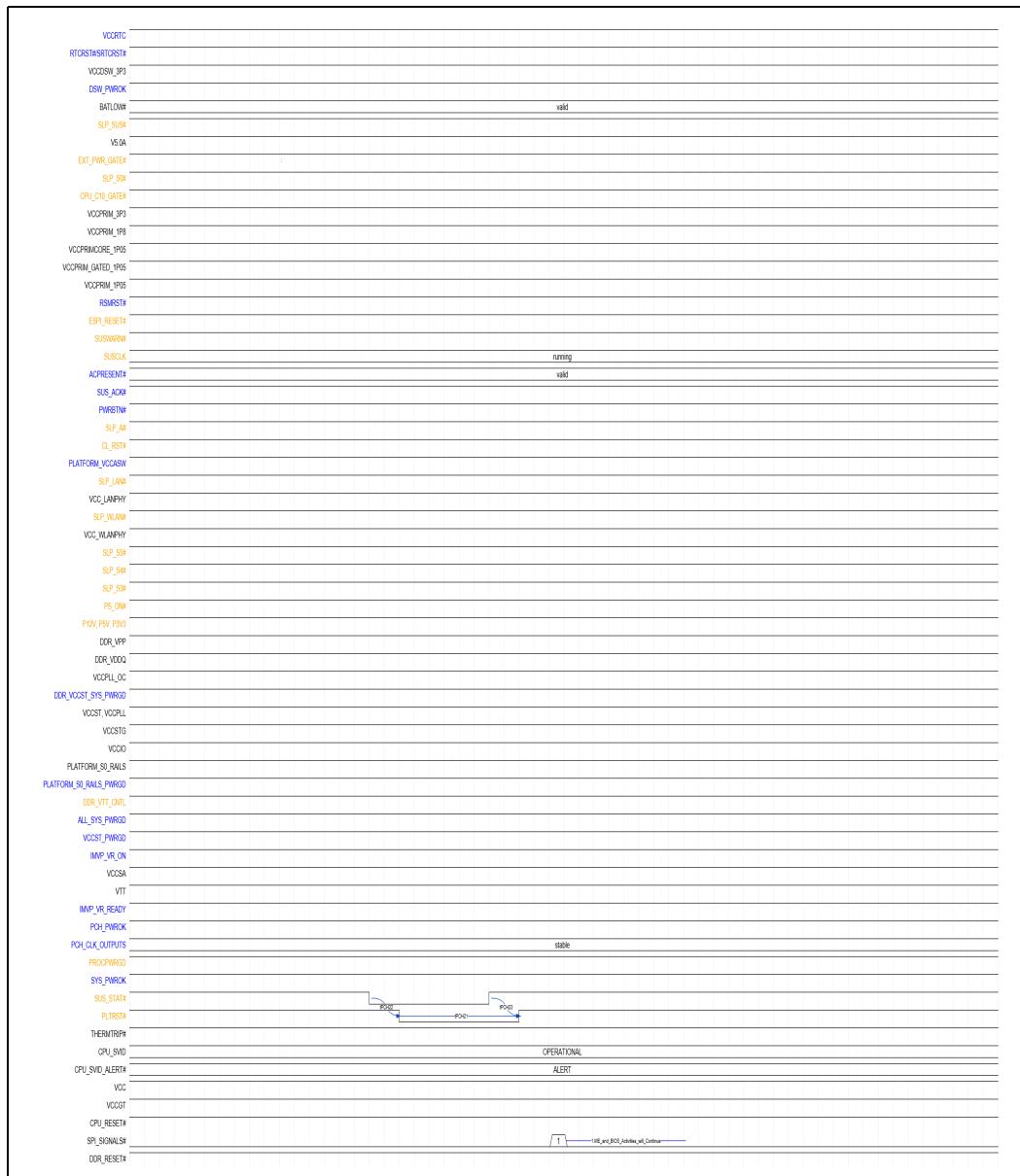
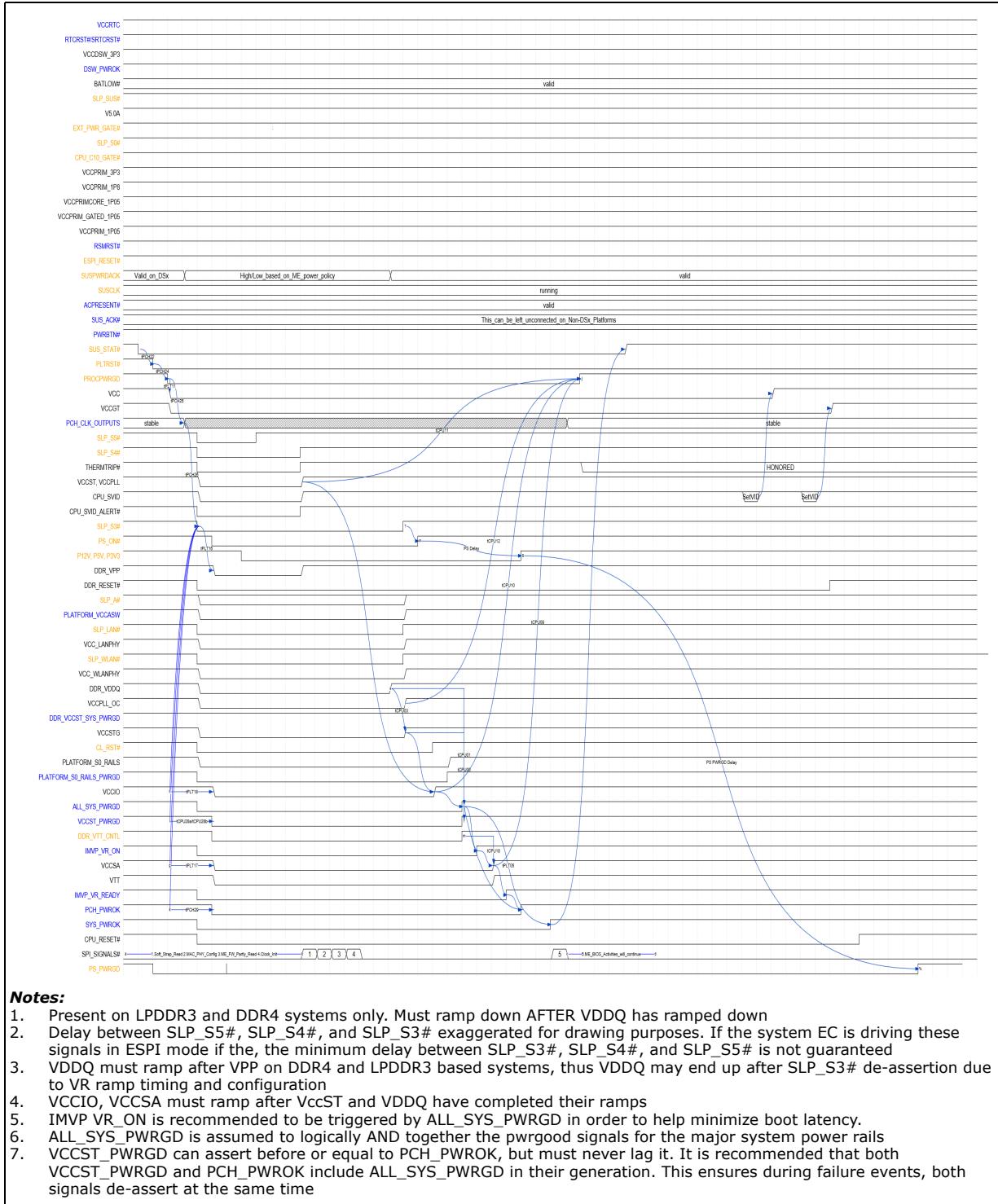


Figure 45-8. Timing Diagram for Cold Reset (Host Partition Reset w/ Power Cycle) and Global Reset [Non-Deep Sx Platform]



45.4.1 Power Sequencing Timing Requirements

The following table defines the timing parameters used in the timing diagrams in previous sections. The timing parameters are put into three categories: tCPU, tPCH, and tPLT. tCPU parameters are mostly required by the processor, tPCH parameters are required or controlled by the PCH, and tPLT parameters are mostly controlled by the platform.

Table 45-5. Platform Sequencing Timing Parameters (Sheet 1 of 7)

Label	Required By	Controlled By	Min.	Max.	Units	Note #	Description
tCPU00	CPU	PLT	1		ms	6, 7, 26	VCCST ramped and stable to VccST_PWRGD assertion
tCPU01	CPU	PLT	1		ms	6, 7	VDDQ ramped and stable to VccST_PWRGD assertion
tCPU02	CPU	PLT		No limit	ms	26, 43	VCCST ramped and stable before VDDQ stable Note: tCPU02 and tCPU03 are mutually exclusive, and only one of the parameters needs to be met depending on the rail order of VCCST/VCCSTG and VDDQ.
tCPU03	CPU	PLT		25	ms	26, 43	VDDQ ramped and stable before VCCST stable Note: tCPU02 and tCPU03 are mutually exclusive, and only one of the parameters needs to be met depending on the rail order of VCCST/VCCSTG and VDDQ.
tCPU04	CPU	PLT	0		ns	26, 31	VCCST must always ramp with or earlier than VCCSTG. VCCST >= VCCSTG at all times during ramp.
tCPU05	CPU	PLT	100		ns		VDDQ ramped and stable before VCCSA/VCCIO ramps
tCPU06	CPU	PLT	100		ns	27	VCCST ramped and stable before VCCSA/VCCIO ramps.
tCPU07	CPU	PLT	No Req	No Req	ns		VCCSA ramped and stable before VCCIO stable Note: this timing is to explicitly call out that there is no timing requirement between VCCSA and VCCIO.
tCPU08	CPU	PCH	1		ms	8	PCH CLK outputs stable and CPU VRs stable/ready to PROCPWRGD assertion
tCPU09	CPU	PLT	1		ms	23	VCCSA stable before PROCPWRGD
tCPU10	CPU	PLT	1		ms	23	VCCIO stable before PROCPWRGD
tCPU11	CPU	PLT	1		ms	23	VCCPLL stable before PROCPWRGD
tCPU12	CPU	PLT	1		ms	23	VCCPLL_OC stable before PROCPWRGD
tCPU16	CPU	PLT	0		ns	32	VCCST_PWRGD assertion to PCH_PWROK assertion
tCPU18	CPU	PLT	0	35	us	34	DDR_VTT_CNTL (was DDR_PG_CTL) assertion to DDR_VTT supplied ramped and stable while PLTRST = H (de-asserted).
tCPU19		CPU	0	100	ns	9	VccST_PWRGD assertion to DDR_VTT_CNTL (was DDR_PG_CTL asserted).
tCPU20	CPU	PLT + PCH		500	ms	10	THERMTRIP# assertion until VCC, VCCGT, VCCSA, VCCIO VRs are disabled and not sourcing power.
tCPU21	CPU	PCH		1	ms		THERMTRIP# assertion until PCH issues global reset and assertion of SLP_SX signals

Table 45-5. Platform Sequencing Timing Parameters (Sheet 2 of 7)

Label	Required By	Controlled By	Min.	Max.	Units	Note #	Description
tCPU22	CPU	PLT	1		us		VCCST_PWRGD de-assertion to either VDDQ, VCCST below specification for normal S0 to Sx transitions. Recommend VCCST_PWRGD goes low with SLP_S3#.
tCPU23	CPU	PCH	TBD		us	45	PROCPWRGD de-assertion to either VCC* below specification for normal S0 to Sx transitions
tCPU26	CPU	PLT	10	65	us	11	CPU_C10_GATE# de-assertion to VCCSTG stable Note: CPU_C10_GATE# de-assertion to VCCST also needs to meet max 65us on cold boot
tCPU27	CPU	PLT	10	240	us	11	CPU_C10_GATE# de-assertion to VCCIO stable
tCPU28a	CPU	PLT		200	us	36	SLP_S3# assertion to VCCST_PWRGD de-assertion
tCPU28b	CPU	PLT	0		us	37, 38	VCCST_PWRGD low to VCCST dropping 5% of nominal value
tCPU29	CPU	PLT		100	mV/us	13, 25	Processor power rail instantaneous slew rate.
tCPU33	CPU	PLT		240	us	11	CPU_C10_GATE# de-assertion to VCCPLL_OC stable
tPCH01	PCH	PLT	9		ms	1, 47, 48	VccRTC = 2.0V to the point in time where voltage on the RTC resets equals 0.65 times the voltage present on the VccRTC rail during ramp. This measurement should be made from VccRTC = 2.0V to the first of RTCRST# or SRTCRST# reaching 0.65 * VccRTC
tPCH02	PCH	PLT	10		ms		VccDSW stable (@95% of full value) to DSW_PWROK high
tPCH03	PCH	PLT	10		ms		VccPrimary stable (@95% of full value) to RSMRST# high
tPCH04		PCH	9		ms	29, 42	VccRTC stable (@90% of full value) to start of VccDSW voltage ramp for systems with coined RTC battery
tPCH48		PCH	30		ms	29, 42	VccRTC stable (@90% of full value) to start of VccDSW voltage ramp for systems with coinless RTC
tPCH05	PCH	PLT	1		us	42	RTCRST# high (voltage above ViH_min) to DSW_PWROK high (when voltage crosses ViL_max such that internally it might be resolved as a logic '1')
tPCH06	PCH	PLT	200		us		VccDSW 3.3 stable (@95% of full value) to VccPrimary 1.05V starting to ramp (for DSx or nonDSx configurations)
tPCH07	PCH	PLT	0		ms		DSW_PWROK high to RSMRST# high
tPCH08	PCH	PLT	1		ms		SLP_S3# de-assertion to PCH_PWROK assertion

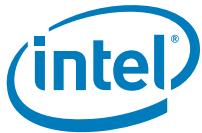


Table 45-5. Platform Sequencing Timing Parameters (Sheet 3 of 7)

Label	Required By	Controlled By	Min.	Max.	Units	Note #	Description
tPCH09	PCH	PLT	2, 4, 8, 16		ms		SLP_A# high to PCH assuming ASW rails are stable (95% of full value)
tPCH10	PCH	PLT	400		ns	4, 5	PCH_PWROK low to VCCIO dropping 5% of nominal value
tPCH11	PCH	PLT	100		ns		SLP_SUS# asserting to VccPRIM dropping 5% of nominal value
tPCH12	PCH	PLT	400		ns	5, 24	RSMRST# asserting to VccPRIM dropping 5% of nominal value
tPCH13	PCH		0		ms	14	RTCRST# asserting to VccRTC dropping 5% of nominal value (this applies only when RTC battery is removed)
tPCH14	PCH	PLT	400		ns	4, 5	DSW_PWROK falling to any of VccDSW, VccPrimary dropping 5% of nominal value
tPCH15	PCH	PLT		20	ms		SLP_LAN# (or LANPHYPC) rising to VccLANPHY high and stable
tPCH18	PCH	PCH	90		us		De-assertion of RSMRST# to De-assertion of eSPI_RESET# on exit from G3 (DSx platforms only), or exit from DSx
		PCH	95		ms		De-assertion of RSMRST# to De-assertion of eSPI_RESET# on exit from G3 (Non-DSx platforms only)
tPCH20	PLT	PCH	-100		ns		DDR_RESET# assertion to SLP_S4# assertion (The negative minimum timing implies that DDR_RESET# must either fall before SLP_S4# or within 100 nS after it).
tPCH21		PCH	Note 39		ms	39	Warm Reset PLTRST# assertion duration time
tPCH22		PCH	210		us		SUS_STAT# active to PLTRST# active
tPCH23		PCH	60		us		SUS_STAT# de-assertion to PLTRST# de-assertion
tPCH24		PCH	30		us		PLTRST# assertion to PROCPWRGD de-assertion
tPCH25		PCH	10		us		PROCPWRGD de-assertion to CLKOUT_BCLK turning OFF. This timing is programmable (Range = 10us, 100us, 1ms, 10ms). Default is set to 10 us.
tPCH26		PCH	1		us		CLKOUT_BCLK turning OFF to SLP_S3# assertion
tPCH27		PCH	30		us	33	SLP_S4# assertion to SLP_S5# assertion
tPCH28		PCH	30		us	33	SLP_S3# assertion to SLP_S4# assertion
tPCH29		PCH	0		ms		SLP_S3# assertion to PCH_PWROK deassertion
tPCH31		PCH		tPCH02 + tPCH32	ms	15, 44	VccDSW3p3 ramped and stable until SUSCLK is stable. SUSCLK will start before the max timing, but will not be guaranteed valid until max timing is met
tPCH32		PCH	95		ms		DSW_PWROK assertion to SLP_SUS# de-assertion (only applicable to Deep Sx supported platforms).

Table 45-5. Platform Sequencing Timing Parameters (Sheet 4 of 7)

Label	Required By	Controlled By	Min.	Max.	Units	Note #	Description
tPCH33		PCH	0, 99		ms	16	PCH_PWROK high to PLTRST# de-assertion. This timing is set by the PCH via Soft strap settings.
tPCH34	PCH	PLT		50	ms		All PCH Primary Rails should ramp up within this window.
tPCH35	PCH	PLT		100	ms	20	All PCH Rails should ramp down within this window.
tPCH36	PCH	PLT		100	mV/us		PCH Power rails instantaneous slew rate
tPCH37	PCH	PLT	5		mV/us	21	MPHY / SRAM Supply instantaneous slew rate
tPCH41	PCH	PCH	1		ms		PCH_PWROK high to PCH clock outputs stable
tPCH42	PCH	PLT		10	mV/us		VCCPRIM_Core slew rate during VID change
tPCH43	PCH	PLT	95		ms	28	DSW_PWROK assertion to PWRBTN# monitored
tPCH44	PCH	PLT	500		us		tPCH09 expiring to CL_RST# high
tPCH45		PCH	1, 5, 50, 100		ms	41	Clock outputs stable to PROCPWRGD assertion to processor. Timing set by PCH via Soft Strap settings
tPCH46		PCH	1, 2, 5		ms	41	PROCPWRGD and SYS_PWROK High to SUS_STAT# de-assertion. NOTE: Timing can be adjusted through the FIT tool
tPCH47	PCH	PLT	0		ms	46	RCIN# de-asserted to PCH_PWROK assertion
tPLT01		PCH	200		ms	2	RSMRST# de-assertion to SUSPWRDNACK valid. Timing set by PCH.
tPLT02			0	90	ms		RSMRST# de-assertion to ACPRESENT valid (not floating). NOTE: This is only for platforms not supporting Deep Sx state
				0	ms		RSMRST# de-assertion to ACPRESENT valid (not floating) [For platforms supporting Deep Sx state]. NOTE: ACPRESENT is a powered input to the PCH while in the Deep Sx state (that is, when DSW_PWROK=1 and RSMRST#=0). The PCH implements a weak pull-down on ACPRESENT while RSMRST#=0 such that the input is observed as 0 if not driven actively by the board during this time. If driven actively by the board to 1 while in Deep Sx and if the PCH is enabled to treat the ACPRESENT indication as a wake event from Deep Sx, then the PCH will power up the Primary well.
tPLT04	CPU/PCH	PLT	1		ms	3, 19	ALL_SYS_PWRGD assertion to PCH_PWROK. This timing must be controlled on the platform.
tPLT05		PLT	Platform dependent	No limit		18	ALL_SYS_PWRGD assertion to SYS_PWROK. This timing must be controlled on the platform. SYS_PWROK provides a platform/EC mechanism to stall the PCH de-assertion of PLTRST# to the platform. Different platform components may have different timing requirements from when their power rails are stable until their respective reset signal can de-assert. Platform designer should adjust this timing based on their specific platform requirements.



Table 45-5. Platform Sequencing Timing Parameters (Sheet 5 of 7)

Label	Required By	Controlled By	Min.	Max.	Units	Note #	Description
tPLT11	CPU	PLT		500	ms		SLP_S3# assertion to VCC, VCCGT, VCCIO and VCCSA <100mV.
tPLT14		PCH	4		s		Power cycle duration time; programmable via PM_CFG.PWR_CYC_DUR bit.
tPLT15	PLT (MEM)	PLT		200	us	40	SLP_S4# assertion to VDDQ EN Low (VDDQ VR disabled). Memory dependent, refer JEDEC requirements
tPLT16	PLT (MEM)	PLT	30		ms	30	VDDQ ramp down to start of VPP ramp down when entering S4 and lower. This is not SOC requirement but a recommended timing to help meet the JEDEC sequencing requirements. Platform designers are responsible for making sure their designs meet the power sequencing requirements for the memory technology used in their designs.
tPLT20	PLT (MEM)	PLT	2.5		ms	30	VPP ramped to VDDQ start of ramp when entering S0 at power up. This is not SOC requirement but a recommended timing to help meet the JEDEC sequencing requirements. Platform designers are responsible for making sure their designs meet the power sequencing requirements for the memory technology used in their designs.
tPLT17	CPU	PLT		200	us	35	SLP_S3# assertion to IMVP VR_ON de-assertion
tPLT18	CPU	PLT		200	us	35	SLP_S3# assertion to VCCIO VR disabled
tPLT19	PLT	PLT		10	us	22	RSMRST_PWRGD# or SLP_SUS# assertion to RSMRST# assertion

**Table 45-5. Platform Sequencing Timing Parameters (Sheet 6 of 7)**

Label	Required By	Controlled By	Min.	Max.	Units	Note #	Description
Notes:							
<ol style="list-style-type: none"> 1. PCH Primary Rails must never be active while VCCRTC is OFF 2. RSMRST# should be de-asserted for at least 200ms before PCH will consider SUSPWRDNACK valid. 3. PCH can delay de-assertion of PLTRST# with a soft strap setting. Refer tPCH08 and tPCH09 and the SPI Programming Guide for more details. 4. For catastrophic/surprise power failures only. 5. For surprise power down cases, if DSW_PWROK is de-asserted (tPCH14) before DSW3.3 and any other Prim rails drop out of spec, there is no risk of RTC corruption (assuming VCCRTC was previously powered and VCCRTC# is de-asserted) and this spec can be ignored 6. VCCST_PWRGD has no edge rate requirement, but edges must be monotonic. 7. VCCST_PWRGD must accurately reflect the state of VCCST and must not glitch when VCCST, or VDDQ power is applied. Additionally, VCCST_PWRGD must track to the state of PCH_PWROK on the platform. When PCH_PWROK de-asserts during S0 --> Sx transitions, then VCCST_PWRGD must also de-assert. 8. Processor's PROCPWRGD is not expected to be used externally on the platform, but is available for monitoring. 9. CPU will assert DDR_VTT_CNTL after VCCST_PWRGD crosses its threshold point on rising edge, which will typically be around 0.5*VccST but can be anywhere between Vil to Vih levels. 10. It is strongly recommended that the SLP_S3# be a qualifying input signal to ALL_SYS_PWRGD logic, which drives IMVP VR_ON inputs. Additionally, it is recommended that SLP_S3# also qualify the EN control to the VCCIO power supply 11. NA 12. NA 13. Applies to all CPU power supply rails 14. There are no perceived functional failures if this minimum is violated since all data is lost anyway when VccRTC drops. The only real requirement is to avoid driving a high signal level into the powered down device for an extended period of time, assuming an external device(s) on a different power supply than VCCRTC is driving RTCRST# or any other RTC well input signal. 15. SUSCLK is now powered in DSW well. 16. PCH can delay de-assertion of PLTRST# with a soft strap setting. If tPCH33 is not enabled, the platform is responsible for controlling the assertion timing of PCH_PWROK and SYS_PWROK in such a way that it satisfies platform component timing requirements of power stable to reset de-assertion. Refer SPI Programming Guide for more details. On a platform level this timing doesn't necessarily need to meet 99ms, PCH soft straps guarantees a min of 99ms. 17. NA 18. Example, if the platform only has mini-PCIe* devices requiring a 1 ms delay from power rails stable to PCIe* reset de-assertion, then the minimum value for ALL_SYS_PWRGD assertion to SYS_PWROK can be reduced to 1 ms. If SYS_PWROK asserts before PCH_PWROK, no additional delay will be added by the PCH to delay PLTRST# de-assertion and the final timing value will be subject to internal PCH timing parameters. 19. PCH_PWROK assertion assumes all the following CPU and PCH voltage rails are ramped and stable except for: VCC, VCCGT, VCCGTX, VCCOPC, VCCEOPIO 20. No Ramp down requirement between rails. All VRs should stop regulation based on SLP_SUS# signal, or another common shutdown signal 21. NA 22. 10us max limit is an estimation and will vary based on platform VR and EC implementation details. Platform designers are responsible for ensuring that tPCH12 is not violated during normal DSx entries based on EC and VR response times to assertion of SLP_SUS# and/or RSMRST_PWRGD# 23. PCH will have a minimum of a 1ms delay from PCH_PWROK to assertion of PROCPWRGD. Therefore, the platform is only responsible to have these rails ready a minimum of 0ms before PCH_PWROK assertion 24. Applicable to all power down sequences where Prime well rails are turned off and to emergency power loss. Refer RSMRST#/DSW_PWROK Special Considerations section for more details. 25. Only applicable to SKUs with OPC support 26. If VCCSTG and VCCIO supplies are merged together as a single supply, then VCCSTG/VCCIO supply must also satisfy this requirement 27. If VCCSTG and VCCIO supplies are merged together as a single supply, then the timing requirement is between VCCST/VCCSTG/VCCIO and VCCSA 							



Table 45-5. Platform Sequencing Timing Parameters (Sheet 7 of 7)

Label	Required By	Controlled By	Min.	Max.	Units	Note #	Description
Notes: (cont)							
28.							Applicable to all G3 exits where GEN_PMCN_B.AG3E=1. The PWRBTN# must assert for at least 16ms after the minimum tPCH43 timing before PCH will detect PWRBTN# as a wake event
29.							For scenarios where the coin-cell is dead or missing, the PCH can electrically tolerate the VCCRTC and VCCDSW/VCCPRIM 3.3V supplies ramping simultaneously with VCCDSW/VCCPRIM 3.3V leading VCCRTC by ~0.2 – 0.3V (i.e. voltage drop associated with the coin cell Schottky diode circuit). These scenarios are not allowed as typical power sequencing scenarios, and are expected to be very infrequent. In these scenarios, glitch-free signal operation on DSW signals can not be guaranteed when the platform is configured for DSx mode. Likewise, glitch-free signal operation on DSW and PRIM well signals can not be guaranteed when the platform is configured for non-DSx mode
30.							Generally, JEDEC specifications require VPP > = VDDQ for both power up and power down sequences. This timing parameter is a suggested value as ensure this requirement is met on power down, but systems designer must account for other platform level considerations such as output decoupling, discharge circuits, etc that can alter this relationship.
31.							VCCST supply is typically controlled by SLP_S4#, and VCCSTG supply is typically controlled by SLP_S0# AND SLP_S3#. Since the timing delay between SLP_S3# and SLP_S4# deassertion during a S4/S5 à S0 transition can be small (defined by tPCH28), OEMs may need to take extra steps to ensure this timing is met. Example, power gates of equivalent slew rate may be required on both VCCST and VCCSTG supplies. Alternatively, a small timing delay path might be required on the SLP_S3# path of the VCCSTG power gate enable
32.							VCCST_PWRGD should start to assert no later than when PCH_PWROK asserts; however, VCCST_PWRGD may lag completing its ramp with respect to PCH_PWROK by up to 20us
33.							Not applicable to eSPI mode, refer to the eSPI Compatibility Specification (RDC#508740) for eSPI implementations
34.							Only applies to configurations that use DDR_VTT_VCTRL signal to enable/disable VTT VR and only applies after PLTRST# has de-asserted (ie, S0). VTT must be completely ramped and ready for full load at or before the de-assertion of PLTRST# on Sx -> S0 transitions.
35.							Timing to VR being disabled, not until the VR is fully ramped down
36.							S0 to S3 transition with VCCST powered in S3 state
37.							S0 to Sx transition with VCCST unpowered in Sx
38.							Recommend not to exceed 200us delay with respect to SLP_S3#
39.							During a warm reset sequence, the PCH sequencing flows will internally re-execute several timing parameters and tPCH45 and tPCH46 are two of the parameters that are re-executed. tPCH45 and tPCH46 are primarily for G3/DSx/Sx --> S0 flows, but they are executed for cold boot, cold reset, and warm reset. The nominal setting for these timings is partially controlled by the setting of SPI soft straps associated with tPCH45 and tPCH46. The minimum possible PLTRST# assertion time will be equal to the soft strap settings for tPCH45 + tPCH46. The maximum PLTRST# assertion time is not guaranteed across all warm reset cycles, but the typical worst case assertion time is approximately equal to (tPCH45 soft strap setting) + (tPCH46 soft strap setting) + 20ms.
40.							This is a platform timing recommendation to help ensure that memory device power down sequencing requirements between VDDQ and VPP are likely to be met (refer tPLT16). This is not a requirement for the CFL SoC itself. Refer to the JEDEC LPDDR3 and DDR4 power down sequencing requirements for more details
41.							This parameter is programmable, refer to the Flash Descriptor Record in the SPI Programming Guide in the ME firmware package
42.							For the dead RTC coin cell and coin cell-less with depleted main battery RTC cases where the 3.3DSW ramps and powers the RTC well, the board designer should set the 3.3DSW rail stable to DPWROK assertion delay (tPCH02) for = 30ms. This is required to ensure that SRTCRST# and RTCRST# de-assert after VCCRTC is stable, but before DPWROK assertion. Failure to meet this requirement may result in DPWROK asserting with, or before, SRTCRST# and RTCRST# reach VIH, which is a sequencing violation and can result in a non-booting system scenario. Refer Section 30.2 (Real Time Clock Topology Guidelines) and Section 59.3.13 (RTC Circuit) for SRTCRST# and RTCRST# RC timing network details
43.							tCPU02 and tCPU03 are mutually exclusive, only tCPU02 or tCPU03 needs to be met
44.							SUSCLK stable means the clock is toggling and is within its defined parameters. This timing spec is applicable to all G3 exits, excluding the G3 exit immediately following the ramp of VCCRTC, which may have a longer delay
45.							This does not concern platform design, this is internal to SOC.
46.							RCIN# is expected to be low in S3 and lower, but can be kept high in S3 through S5 without issue. RCIN# low in S3 and lower will not cause an INIT#. This requirement does not preclude the platform for asserting RCIN# after PLTRST# de-assertion when entering S0.
47.							C3 and C4 Capacitors used in the RTC external circuit should be evaluated with regards to aging, voltage and temperature characteristics to ensure reliable operation in the intended operating environment. See section 13.1.3 RTC External RTRST# Circuit.
48.							For measurement details, reference RTC Reset Timing Technical Advisory - RDC#610459.

Additional Notes:

- Unless otherwise noted, all specifications in this table apply to all processor frequencies.
- DDR_RESET# behavior does not directly track the state of VDDQ rail. For DDR_RESET# to assert properly on surprise power failure, the VDDQ VR pwrgood output should be an input into the logic generating PCH_PWROK

45.5 Additional Power Optimizations with Respect to VCCST Rail in S3

Throughout the PDG, it is generally noted that the VCCST rail should be ON during S3 power states, and is thus generally shown as being controlled directly or indirectly by the SLP_S4# signal. To save additional power during S3 power state, it is allowable to turn the VCCST rail OFF during S3 given that the following requirements and considerations are met. First, the following table below must be met. To support this table, an additional power gate would be required on the VCCPLL_OC in order to isolate this supply and turn it off during S3. Second, VCCPLL_OC must meet all the same timing requirements and rail-to-rail ramp requirements as VDDQ relative to all other rails of interest, specifically tCPU01, tCPU02, tCPU03 and tCPU05. All various figures and tables throughout the PDG such as in this chapter and other associated documents such as the Debug Port Design Guide and Coffee Lake Platform Power Design Guidelines will not be updated to explicitly reflect this option of turning VCCST off during S3. Also, it is assumed that VCCST and VCCPLL are supplied from the same power gate on the platform as documented in the Platform Architecture Guide, thus references VCCST in this description also include VCCPLL.

Table 45-6. Allowed VCCST and VCCPLL_OC Power State Combinations

VCCST	VCCPLL_OC	State
OFF	OFF	- VALID
OFF	ON	- INVALID
ON	OFF	- VALID
ON	ON	- VALID

Note: this table reflects valid and invalid power states for static, steady state conditions for leakage and long term reliability considerations

45.6 Rail-to-Rail Power Sequencing Requirements

45.6.1 Rail-to-Rail Sequencing For Various Supplies

The following diagrams show the recommended rail-to-rail sequencing requirement for Deep Sx configuration (Figure 45-3) and Non-Deep Sx configuration (Figure 45-4).

In general, Cannon Lake PCH architecture implements new capability to help minimize rail-to-rail sequencing related problems. As long as the RTC 3.3 V supply is powered up properly in advance of DSW and Primary rails, new isolation logic is active and will help achieve proper isolation between power rails during power up and power down sequences. The I/O interfaces will achieve proper isolation between the Primary 1.0 V supply and the higher 1.8 V and 3.3 V Primary / DSW supplies as well as achieving glitch-less I/O behavior for signals that support this capability.

During scenarios where the RTC supply is not powered such as a dead coin cell scenario and it is instead ramping along the Primary/DSW supplies through external platform circuitry, proper isolation between the Primary 1.0 V supply and the higher 1.8 V and 3.3 V Primary / DSW supplies cannot be guaranteed. It is thus possible that current inrush/back drive events could exist in these cases, but these scenarios are not a reliability risk to the PCH. It is, however, assumed that for general operation of Cannon Lake PCH, the RTC supply should be ramped and valid before the DSW and Primary rails ramp. Ramping the RTC supply simultaneously with the VCCDSW_3p3 rail for



Deep Sx systems and VCCPRIM_3p3/VCCDSW_3p3 rails for Non-Deep Sx systems as a standard power up sequence for every G3 exit is not a valid configuration and is not allowed.

Figure 45-9. CFL H Rail to Rail Sequencing Requirement for Deep Sx Configured System

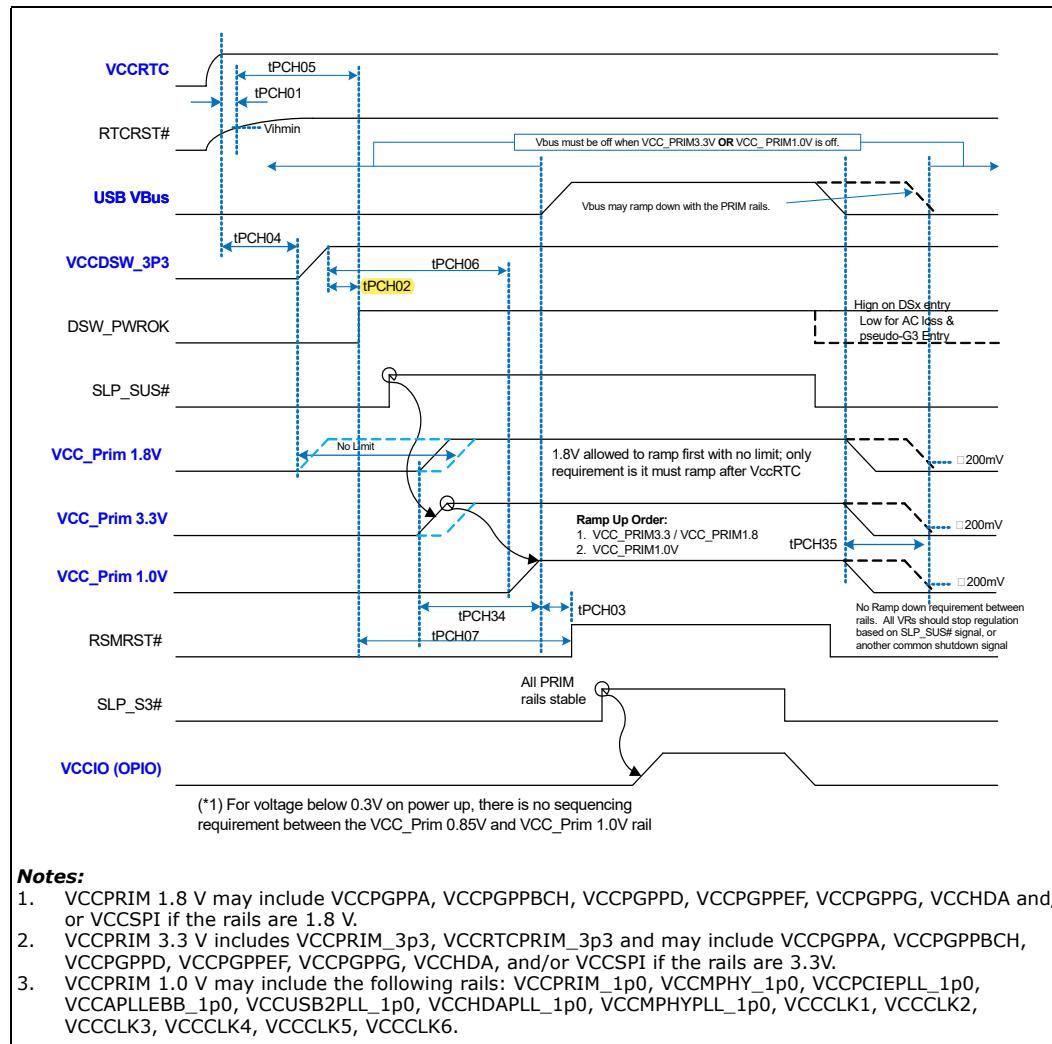
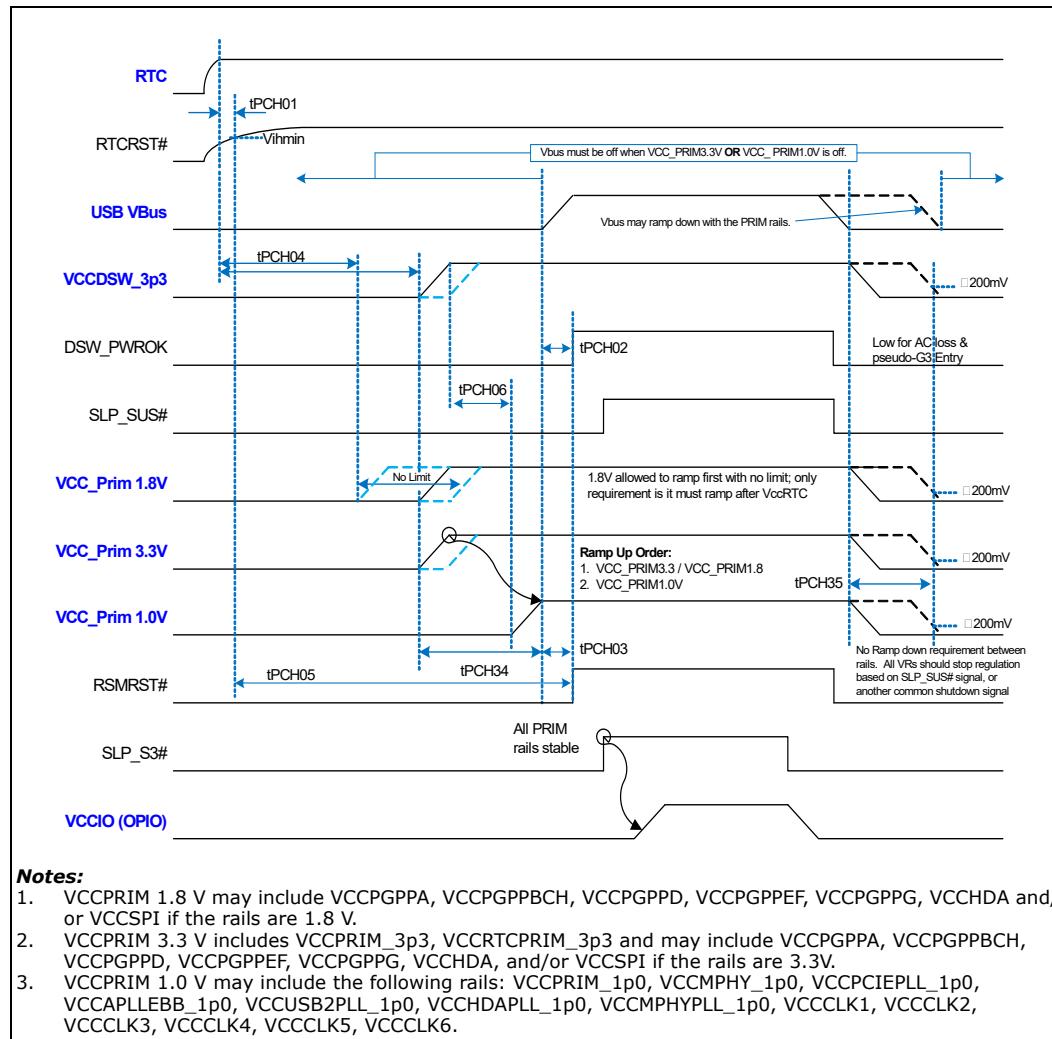


Figure 45-10.CFL H Rail to Rail Sequencing Requirement for Non-Deep Sx Configured System



45.6.1.1 General Rail-to-Rail Sequencing

For power up, all Primary rails should ramp within 50 ms of each other

For power down, there are no explicit timing requirements/relationships between the various Primary Well voltages on power down, but it is required that all Primary Well voltage regulators stop regulation simultaneously based on SLP_SUS# assertion. The power down sequencing should not be staggered from one VR to the next. Natural variance between VRs disabling due to differences in voltage regulator disable time is acceptable on the order of tens of microseconds. It is required that all rails ramp down within ~100 ms.

For surprise power down requirements, refer [Section 45.6.3](#) for assumptions.

**Table 45-7. Rail-to-Rail Sequencing Requirements**

Rail 1	Rail 2	Ramp Up	Ramp Down
VCC_PRIM 3.3 V/ VCC_PRIM 1.8 V	VPRIM_CORE/ VCC_PRIM 1.0 V	3.3 V Primary/1.8 V Primary rails ramp in advance of the 1.0 V Primary and VCCPRIM_Core rails	All rails must ramp down within 100 ms. Refer Section 45.6.1.1 .
VCC_PRIM_3p3/VCC_PRIM 1.0V	Vbus	Vbus ramp after VCC_PRIM 3.3V and VCC_PRIM 1.0V have both reached 95% of their final nominal voltage.	Vbus must ramp down with or before the PRIM rails.

Notes:

1. VCC_PRIM 1.0V may include the following rails: VCCPRIM_1p0, VCCSRAM_1p0, VCCMPHYAON_1p0, VCCMPHYGT_1p0, VCCMPHYPLL_1p0, VCCAPLL_1p0, VCCCLK1, VCCCLK2, VCCCLK3, VCCCLK4, VCCCLK%, VCCCLK6

45.6.1.2 Primary Rails and External USB Vbus Power Sequencing

The Vbus provided to external USB ports can be used by USB devices to power their USB speed detect pull-up resistors. Some devices generate a local 3.3V power supply and pull up the D+ / D- lines with a 1.5k resistor. Other devices, per more recent USB ECRs, are allowed to pull up the D+/D- lines with Vbus voltage directly using ~7.5k resistor; implementation varies with device design.

If Vbus is powered while VCCPRIM_3p3 is not powered, and a device pulls either data line to 3.3V via its speed select pull-up resistor, the PCH will be exposed to leakage current through its un-powered USB 2.0 buffers. These leakage paths potentially impact both Non-Deep Sx and Deep Sx board designs alike.

The leakage paths exist only when Vbus is powered while VCCPRIM_3p3 and VCCPRIM_1.0V are both unpowered.

Powering Vbus while the PRIM rails are not powered is permitted, but designers should be aware that leakage through the PCH may occur.

45.6.2 RSMRST#/DSW_PWROK Special Considerations

When the system is powered off (G3), DSW_PWROK and RSMRST# must not glitch from their Low states while the corresponding PCH rails are not powered or are below normal operating voltage specifications to ensure RTC corruption does not happen.

When a system is in S0-S5 state and not entering a Deep Sx or G3 state, the RSMRST# may only be driven low if the DSW_PWROK is also driven low (by an external controller such as EC). Failure to meet this requirement may result in unexpected PCH behavior, including failure to boot, which may only be recovered through a G3 cycling. This requirement does not apply to systems that tie the DSW_PWROK and RSMRST# together.

RSMRST# must always be driven low before any of the PRIMARY rails fall below the lower end of their tolerance band (Taking RSMRST low without taking DSW_PWROK low is not permitted when not entering DSx states). This is true for all power states transitions including emergency power loss.

45.6.3 Surprise Power Down Sequencing Considerations

Surprise power down events will be treated slightly differently on Cannon Lake PCH compared to past generation platforms. The main goal of the various power down timing specification such as tPCH10, tPCH12, tPCH14, etc., is to ensure proper isolation between the associated power well and the RTC well to guarantee that RTC contents are not accidentally corrupted. There are many events that could cause a surprise power down. The following is a short list of some events, but is not exhaustive:

- VR failure (over current, overvoltage, IC failure, etc.)
- AC removal with no DC Battery present
- Removal of the primary battery

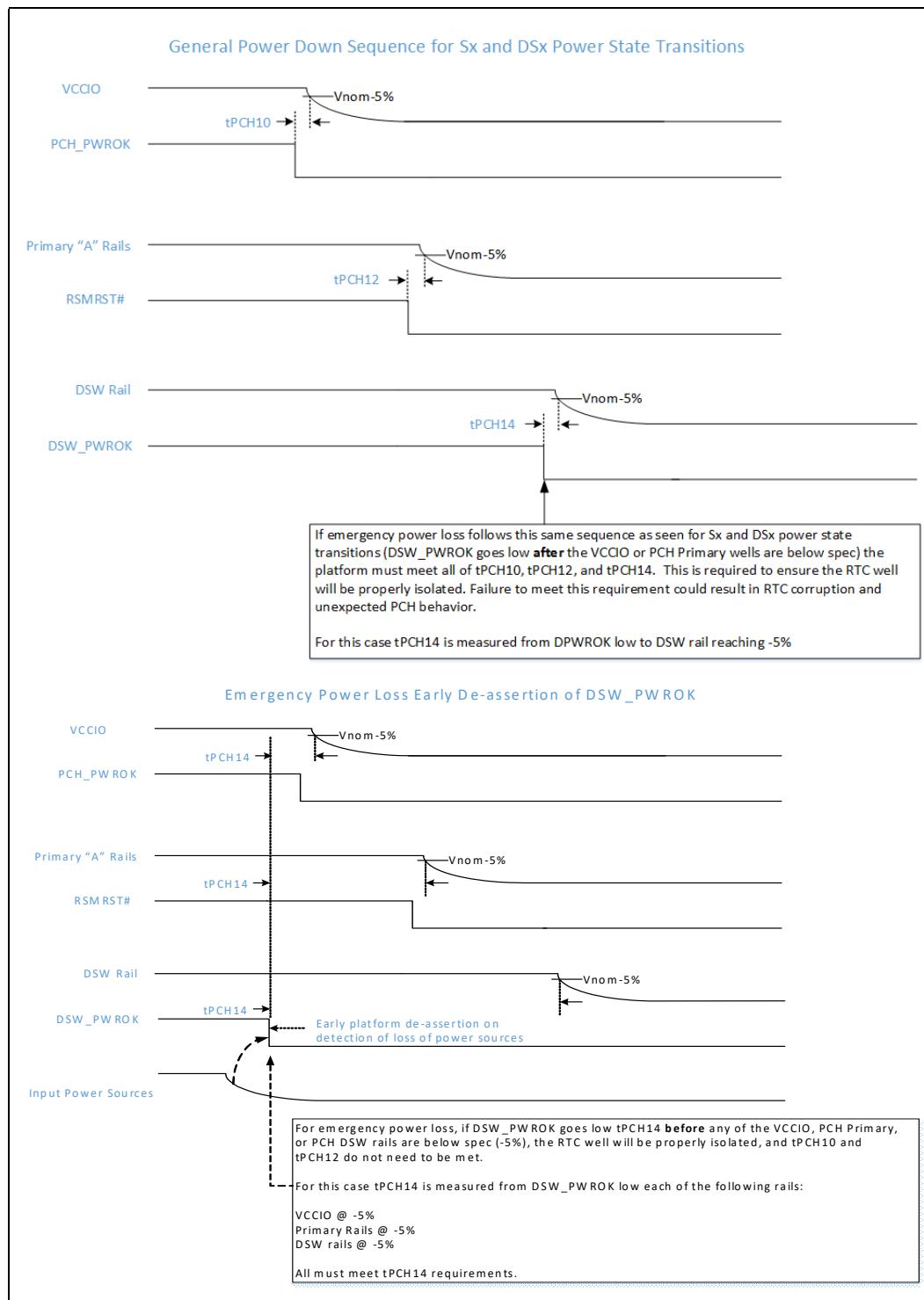
Properly designed platforms generally should not be experiencing VR failures of any kind; therefore the focus of this section is on the unexpected power removal caused by user interaction, which could be an end user, factory technicians, etc.

To ensure RTC is not corrupted, the platform must de-assert the appropriate power good signals BEFORE the rails go out of their defined tolerance range. This implies that the platform should monitor the highest voltage available which is usually the main power supply like the battery voltage to determine when it has dropped too low and VR failure/shutdown is eminent. At that point, the PCH power good signals (PCH_PWROK, RSMRST#, DSW_PWROK) should be driven Low before their associated rails turn off and droop below the defined tolerance. Note, for these cases, de-assertion of just the DSW_PWROK signal is sufficient to activate isolation logic for ALL power wells, thus guaranteeing RTC corruption cannot happen.



Refer Figure 45-7 below for more details.

Figure 45-11.DSW_PWROK Requirement for Power Loss





45.6.4 eSPI Considerations for Sequencing

In general, eSPI support does not have any major impact to power sequencing requirements. However, there are a few behavioral differences worth noting that could have side effects to platform behavior that should be considered.

With eSPI, the EC may or may not take in or drive physical pins that historically would have been supported by the PCH. The values of these signals (ex. SLP_S4#) are tunneled over the eSPI interface between the PCH / EC like virtual wires. If the EC is driving the physical version of these virtual wire signals instead of the PCH driving the signal natively, the PCH cannot guarantee the timing relationships between various tunneled sequencing signals is maintained due to inherent bus latencies.

Only the physical signals directly on the PCH are guaranteed to refer the PCH-defined timing relationships. Example, SLP_S4# de-assertion → SLP_S3# de-assertion relationship is defined as 30 us for the physical pins but could be shorter for the virtual wire relationship on the EC.

45.6.5 Virtual wire SUSWARN deassertion delay during global reset

1. For eSPI enabled platforms, the platform SUS_WARN# de-assertion can occur much later compared to LPC based previous platforms.
2. Hence it is recommended that eSPI based platforms should account for at least 5-6 seconds of delay between eSPI_RESET# de-assertion and SUSWARN# eSPI VW de-assertion from SoC/PCH to EC.
3. This additional delay between above mentioned PCH signals is applicable for Global reset flows only but not for G3 exit flows (or) Sx cycles which do not have the long delay between eSPI_RESET# and SUSWARN.
4. Note that the behavior above does not change the total time for Global reset exit.

§ §

10.12.3 Sequencing Interface Signals List and Power Rails

Table 227. Tiger Lake Platform Sequencing Signals List

Name	Source	Destination	Description
RTCRST#	Platform	PCH	When asserted, this signal resets the register bits in the RTC well.
SRTCST#	Platform	PCH	This signal resets the manageability register bits in the RTC well when the RTC battery is removed
DSW_PWROK	Platform	PCH	Indication to the PCH that VCCDSW_3p3V rail is stable. This signal must be asserted no earlier than 10ms after the DSW power wells are valid. <i>Note:</i> In TGL generation platforms DSW_PWROK and RSMRST# are always separate power ok signals. Refer DSW/PRIM Rail Architecture in DSx and Non-DSx Designs on page 432
RSMRST#	Platform	PCH	This signal is used for resetting the Primary power plane logic. This signal must be asserted for at least 10 ms after the Primary power wells are valid. When de-asserted, this signal is an indication that the power wells are stable. <i>Note:</i> there are special requirements around RSMRST# assertion when NOT entering DSx power states. Refer RSMRST#/DSW_PWROK Special Requirements on page 461 for details.
SUSWARN#	PCH	Platform	This function is only applicable to platforms supporting Deep Sleep Wells This pin asserts low when the PCH is planning to enter the Deep Sx power state and remove Primary power (using SLP_SUS#). The EC/motherboard control logic must observe edges on this pin, preparing for Primary well power loss on a falling edge and preparing for Prim well related activity (host/Intel ME wakes and runtime events) on a rising edge. SUSACK# must be driven to match SUSWARN# once the above preparation is complete. SUSACK# should be asserted within a minimal amount of time from SUSWARN# assertion as no wake events are supported if SUSWARN# is asserted but SUSACK# is not asserted. Platforms supporting Deep Sx, but not wishing to participate in the handshake during wake and Deep Sx entry may tie SUSACK# to SUSWARN#. <i>Note:</i> In eSPI mode this signal is a virtual wire on the eSPI interface and the hard wire signal from PCH is disabled. Refer eSPI Compatibility Specification (# 508740) and Tiger Lake Platform Controller Hub-LP External Design Specification for details.
SUSPWRDNACK	PCH	Platform	This function is only applicable to platforms NOT supporting Deep Sleep States. Asserted high by PCH when it does not require Primary well to be powered. No longer requires a 10-K pull-up to VCCSUS (3.3 V). This signal is required to be connected to EC for platforms with or without M3 support that do not support Deep Sx. This signal gives an added flexibility for the EC to turn OFF the Prim Rails when not needed by the PCH. SUSPWRDNACK can be High/Low in Sx/Moff based on the ME power policy selected. On Platforms that do not support both Deep Sx and M3 EC must keep Prim rails powered ON If, 1. SUSPWRDNACK is de-asserted low Else, EC has the option to turn-off the Prim rails On Platforms that do not support Deep Sx, but supports M3 EC must keep Prim rails powered ON If,

continued...

Name	Source	Destination	Description
			<p>1. SUSPWRDNACK is de-asserted low OR 2. SLP_A# is de-asserted high OR 3. It's the first 200 ms after Prim rails power has been applied. Else, EC has the option to turn-off the Prim rails.</p> <p>Notes: 1. The polarity of SUSPWRDNACK is the opposite of SUS_WARN#. SUS_WARN# will assert low when Primary well power will be turning off, however SUSPWRDNACK will assert high when Primary well power can be turned off. 2. In eSPI mode this signal is a virtual wire on the eSPI interface and the hard wire signal from PCH is disabled. Refer eSPI Compatibility Specification (# 508740) and Tiger Lake Platform Controller Hub-LP External Design Specification for details.</p>
SUSCLK	PCH	Platform	This clock is an output of the RTC generator circuit for use by other chips for refresh clock. SUSCLK is powered from DSW.
ACPRESENT	Platform	PCH	Used on mobile systems to determine presence of AC power or battery power. In addition to previous Intel® ME to EC communication, PCH uses this signal to implement Deep Sx policies. For example, the platform may be configured to enter Deep Sx only on battery and not AC.
SUS_ACK#	Platform	PCH	<p>For platform supporting Deep Sx state, this signal is driven from the platform EC to PCH to acknowledge that EC has received the SUSWARN# signals and it is preparing to go into DeepSx mode.</p> <p>For non-DSW platforms, this signal can be left unconnected. Due to the internal pull-up on this signal it will be pulled high in order for the boot sequence to proceed. It does not matter if this assertion happens before or after SUSPWRDNACK assertion.</p> <p>Note: In eSPI mode this signal is a virtual wire on the eSPI interface and the hard wire signal from PCH is disabled. Refer eSPI Compatibility Specification (# 508740) and Tiger Lake PCH-LP External Design Specification, Volume 1 for details.</p>
SLP_SUS#	PCH	Platform	<p>For platforms supporting Deep Sx state, a low on this signal indicates that PCH is in Deep Sx state and that EC/platform logic does not need to keep the Primary Rails ON.</p> <p>If high means EC must keep Primary rails ON.</p> <p>Unlike previous generation platforms, in TGL SLP_SUS# is used in both DSx and Non-DSx platforms. Refer DSW/PRIM Rail Architecture in DSx and Non-DSx Designs on page 432 for details.</p> <p>Note: In eSPI mode this signal is a hard wire only and not a virtual wire.</p>
PWRBTN#	Platform	PCH	Signal driven from EC to PCH indicating a system request to go into Sleep State OR if the system is already in the Sleep State then it will cause a wake event.
SLP_A#	PCH	Platform	<p>This signal is used to control power to devices on the platform in conjunction with the Intel® ME sub-system. This signal will be asserted in M-off state. If M3 is not supported then SLP_A# will have the same timings as SLP_S3#.</p> <p>On TGL UP3 Platform, SLP_A#/GPD6 is a physical pin (package ball DR41)</p> <p>On TGL UP4 Platform, SLP_A# functionality is implemented on GPP_E4 /SATA_DEVSLP0 pin (package ball DG8). This pin is dedicated for SLP_A# functionality and controlled / owned by the Intel CSME. Therefore, GPP_E4 or SATA_DEVSLP0 function is not available on this pin on TGL UP4 Platform. This pin will behave as SLP_A# once the CSME FW loaded. Prior to that point, this pin will be undriven as it defaults to GPI.</p>

continued...

Name	Source	Destination	Description
			<p><i>Note:</i> In eSPI mode this signal is also virtual wire on the eSPI interface, in addition to the hard signal from PCH. Refer eSPI Compatibility Specification (# 508740) and Tiger Lake Platform Controller Hub-LP External Design Specification for details.</p>
SLP_LAN#	PCH	Platform	<p>LAN Sub-System Sleep Control. Controls power to the LAN PHY. When "low", indicates that power can be shut off to the external wired LAN (GbE) PHY.</p> <p><i>Note:</i> In eSPI mode this signal is also virtual wire on the eSPI interface, in addition to the hard signal from PCH. Refer eSPI Compatibility Specification (508740) and Tiger Lake Platform Controller Hub-LP External Design Specification for details.</p>
SLP_S5#	PCH	Platform	<p>This signal is for power plane control. When asserted (low), it will shut-off power to all non-critical systems in S5 (Soft Off) states.</p> <p><i>Note:</i> In eSPI mode this signal is also virtual wire on the eSPI interface, in addition to the hard signal from PCH. Refer eSPI Compatibility Specification (508740) and Tiger Lake Platform Controller Hub-LP External Design Specification for details.</p>
SLP_S4#	PCH	Platform	<p>S4 Sleep Control. This signal is for power plane control. When asserted (low), it will shut-off power to all non-critical systems in S4 (Suspend to Disk) and lower (S5).</p> <p><i>Note:</i> In eSPI mode this signal is also virtual wire on the eSPI interface, in addition to the hard signal from PCH. Refer eSPI Compatibility Specification (508740) and Tiger Lake Platform Controller Hub-LP External Design Specification for details.</p>
SLP_S3#	PCH	Platform	<p>S3 Sleep Control. This signal is for power plane control. When asserted (low), it will shut-off power to all non-critical systems in S3 (Suspend to RAM) and lower (S4, S5).</p> <p><i>Note:</i> In eSPI mode this signal is also virtual wire on the eSPI interface, in addition to the hard signal from PCH. Refer eSPI Compatibility Specification (508740) and Tiger Lake Platform Controller Hub-LP External Design Specification for details.</p>
CL_RST#	PCH	Platform	This signal connects to the Wireless LAN device supporting Intel® AMT.
VDDQ_PWRGD	Platform VR	Platform Logic	Indicates that the DRAM power supply is stable and within specification.
Platform S0 Rails	Platform	Processor/PC H	These are all the non-core platform rails.
RSMRST_PWRGD#	Platform	Platform	The signal represents power good for VCCPRIM rails and other S5 rails.
ALL_SYS_PWRGD	Platform	Platform	This signal represents the power good for all the rest of platform voltage rails.
IMVP VR_READY	Platform	Platform Logic/SoC	IMVP VR_READY is an active-high output that indicates that the IMVP9 start-up sequence is complete. Assertion of VR_READY indicates that the IMVP9 VR is ready to accept an SVID commands and is operating properly. Refer to the VR_READY definition in the IMVP9 spec for additional details.
PCH_PWROK	Platform	PCH	When asserted, PCH_PWROK indicates that all the main PCH Primary rails and all the CPU rails are up.
SYS_PWROK	Platform	PCH	Generic power good input to the PCH is driven and utilized in a platform-specific manner. Informs PCH that power is stable to some other system component(s) and the system is ready to start the exit from reset (de-asserts PLT_RST# to the processor).

continued...

Name	Source	Destination	Description
			<p>Note: PCH_PWROK and SYS_PWROK both needs to be high to exit reset, but either signal can come up first. PCH does not monitor SYS_PWROK until after PCH_PWROK is asserted. SYS_PWROK may be tied to PCH_PWROK if the platform does not need the use of SYS_PWROK.</p>
DDR_RESET#	PCH	Processor	Controls reset to the memory subsystems (DDR4/LPDDR4)
PROCPWRGD	PCH	Processor	<p>Indicates that VCCST, VCCSTG, VCCIN (VCCIO, VCCSA), VDDQ power supplies and clocks are stable. This signal will be asserted only after PCH_PWROK assertion.</p> <p>VCCIO/VCCSA are FIVR based and are internal to SOC in UP3 and UP4 line.</p> <p>Also, in UP3 and UP4 line SOCs PROCPWRGD signal is internal to the package, but may be observable by designer for debug purposes.</p>
SUS_STAT#	PCH	Platform	<p>This signal is asserted by the PCH to indicate that the system will be entering a low power state soon. This can be monitored by device with memory that need to switch from normal refresh to suspend refresh mode. It can also be used by other peripherals as an indication that they should isolate their outputs that may be going to powered-off planes.</p> <p>Note: In eSPI mode this signal is a virtual wire on the eSPI interface and the hard wire signal from PCH is disabled. Refer eSPI Compatibility Specification (508740) and Tiger Lake Platform Controller Hub-LP External Design Specification (# 576591) for details.</p>
PLTRST#	PCH	Processor	The PCH asserts PLTRST# to reset device on the platform (e.g., SIO, FWH, LAN etc.). Asserted during power-up and when S/W initiates a hard reset sequence through the Reset Control register.
SPI	PCH	Flash Device	<p>Serial Peripheral Interface between PCH and BIOS Flash Device.</p> <p>Note: In eSPI mode this signal is also virtual wire on the eSPI interface, in addition to the hard signal from PCH. Refer eSPI Compatibility Specification (508740) and Tiger Lake Platform Controller Hub-LP External Design Specification (# 576591) for details.</p>
SLP_WLAN#	PCH	WLAN	WLAN Sub-System Sleep Control: When SLP_WLAN# is asserted, power can be shut off to the external wireless LAN device. SLP_WLAN# will always be de-asserted in S0.
LAN_WAKE#	PHY	PCH	<p>Can be used by the LAN PHY as a wake signal.</p> <p>Note: In eSPI mode this signal is also virtual wire on the eSPI interface, in addition to the hard signal from PCH. Refer eSPI Compatibility Specification (508740) and Tiger Lake Platform Controller Hub-LP External Design Specification (# 576591) for details.</p>
ESPI_RESET#	PCH	Platform	Controls reset to eSPI
VCCST_PWRGD	Platform	Processor	Indication that the VCCSTG\VCCST\VDDQ power supplies are stable and within specification
VCCST_OVERRIDE	PCH	Platform/Processor	Signal that allows PCH to keep VCCST powered ON (in case VCCST is powered down) for Type C wake capability.
VCCST_PWRGD_TCS	PCH	Processor	Power good signal to Processor TCSS block for Type C wake capability.
DDR_VTT_CNTL	CPU	VTT VR	Enable signal for the DDR VTT VR
SLP_S0#	PCH	Platform	S0 Sleep Control. When PCH is idle and processor is in C10 state, this pin will assert indicate VR controller can go into a light load mode. This signal can also be connected to EC for other power management related optimizations.

continued...

Name	Source	Destination	Description
			<i>Note:</i> In eSPI mode this signal is a hard wire only and not a virtual wire.
CPU_C10_GATE#	CPU	Platform	Power gating control to turn off VCCSTG in C10 and lower. <i>Note:</i> In eSPI mode this signal is a hard wire only and not a virtual wire.

Table 228. Tiger Lake Power Sequence Related Power Rails

	Name	Source	Destination	Description
Common	VCCRTC	Platform	PCH	2.0-3.3V +5% supplies for PCH RTC Well. This power is not expected to be shut off in any of the sleep states unless the RTC battery is removed or completely drained.
Common	VCCDSW_3P3	Platform	PCH	3.3-V supply for Deep Sx wells. If Deep Sx is not supported on the platform, tie to VCCPRIM_3p3
Common	VCC1P05_OUT_FET	PCH	Platform	FIVR output of PCH to platform 1.05V Power Gates (VCCST/VCCSTG)
Common	VCCPRIM_1P8/3P3 ²	Platform	PCH	PCH I/O and Misc rails 1.8/3.3V (Primary Well)
	VCC_VNNEXT_1P05 VNN_BYP	Platform	PCH	Optional BYPASS rail for PCH Prime Core Well (760mV in S0ix and 1.05V in Sx states) or reduced power consumption in low power states
	VCC_V1P05EXT_1P05 V1P05_BYP	Platform	PCH	Optional BYPASS rail for PCH Primary Well (1.05V) for reduced power consumption in low power states
Common	VCCST	Platform	Processor	Sustain voltage for processor in Standby modes
Common	VCCSTG	Platform	Processor	Gated version of VCCST
Common	VPP	Platform	Memory	Memory power rail, voltage dependent on memory technology
Common	VDDQ	Platform	Processor	CPU Memory power rail, voltage dependent on memory technology
	VCCIN	Platform	Processor	Processor FIVR input power supply
	VCCIN_AUX	Platform	PCH	PCH FIVR input power supply

10.12.4 Power States

VCC_VNNEXT_1P05 (VNN_BYP) and VCC_V1P05EXT_1P05 (V1P05_BYP) are two dedicated voltage regulators that save platform power during low power states (Sx/S0ix).

Both VCC_V1P05EXT_1P05 and VCC_VNNEXT_1P05 are VIDed rails that operate at two voltage levels each. The VID control for this VR is the SLP_S3#, V1P05_CTRL, VNN_CTRL (optional) signals.

Table 229. System with M3 State Supported

Rails	SKU ^s	S0/M0 ³	C10 ²	S0ix/M-off ⁴	S4 and S5/M3	S4 and S5/M-off	Deep S4/S5	G3 ¹
VCCRTC	All	ON	ON	ON	ON	ON	ON	ON
VCCDSW_3P3	All	ON	ON	ON	ON	ON	ON	No Power
VBATA (VDC)	All	ON	ON	ON	ON	ON	ON	No Power
V5.0A	All	ON	ON	ON	ON	ON	OFF	No Power
VCCPRIM_3P3	All	ON	ON	ON	ON	ON	OFF	No Power
VCCPRIM_1P8	All	ON	ON	ON	ON	ON	OFF	No Power
VCC_VNNEXT_1P05	All	ON	ON	ON	ON	ON	OFF	No Power
VCC_V1P05EXT_1P05	All	ON	ON	ON	ON	ON	OFF	No Power

continued...

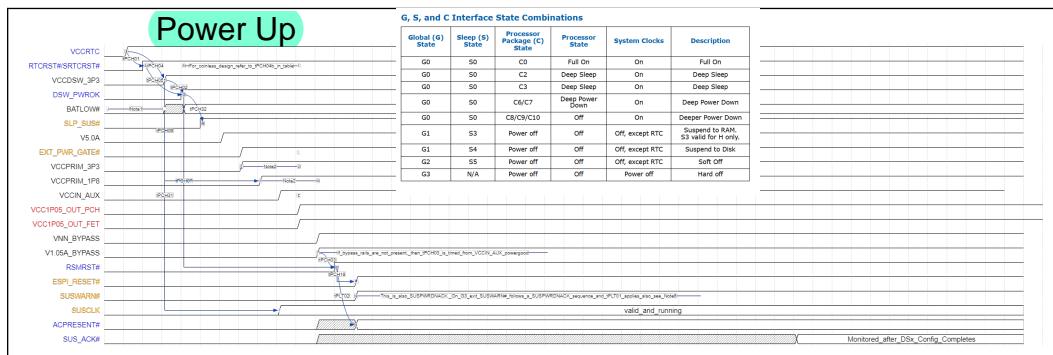
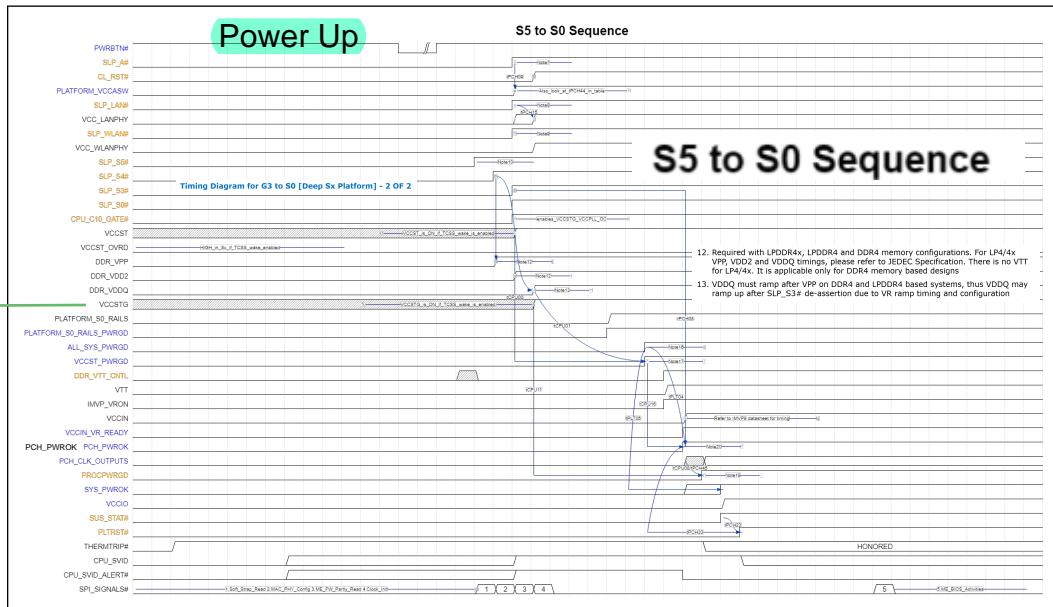
Rails	SKU's	S0/M0 ³	C10 ²	S0ix/M-off ⁴	S4 and S5/M3	S4 and S5/M-off	Deep S4/S5	G3 ¹
V3.3M ⁵	All	ON	ON	OFF	ON ¹⁰	OFF	OFF	No Power
V1.8M ⁵	All	ON	ON	OFF	ON ¹⁰	OFF	OFF	No Power
VDDQ	All	ON	ON	ON	OFF	OFF	OFF	No Power
V2.5U (VPP)	All	ON	ON	ON	OFF	OFF	OFF	No Power
VCCST	All	ON	ON	ON	OFF ⁶	OFF ⁶	OFF	No Power
VCCSTG	All	ON	OFF ²	OFF	OFF	OFF	OFF	No Power
VCC1P8A ¹⁵	H	ON	OFF	OFF	OFF	OFF	OFF	No Power
V3.3S	All	ON	ON	ON	OFF	OFF	OFF	No Power
VCCIN	All	ON	ON	ON ¹¹	OFF	OFF	OFF	No Power
VCCIN_AUX ¹³	All	ON	ON	ON ¹¹	OFF ¹⁴	OFF ¹⁴	OFF	No Power

- Notes:
1. The state of the system without RTC well powered can also be considered G3.
 2. VCCSTG can be turned off when the processor is in C10
 3. S0/M0 state includes all Package C-states from C0-C10
 4. Assume SLP_S0# and CPU_C10_GATE# have asserted from the PCH
 5. V3.3M and V1.8M are platform rails used by external devices which ME operates during Sx/M3 states. These rails are not used directly by the CPU/PCH, and are not present on non-M3 supported systems
 6. VCCST and VCCSTG can remain powered during S4 and S5 power states for board cost optimization. VCCST, VCCSTG may also remain powered in S4 and S5 for debug purposes. Refer to Platform Debug and Test Hooks chapter for more details.
 7. NA
 8. NA
 9. VCCSTG is allowed to be ramped to 0V during S0 only when CPU_C10_GATE# is asserted. Specific exit latency targets are required when this feature is implemented. If VCCSTG power gating is not supported on the platform (shared with VCCST), VCCSTG is allowed to stay ON during S0ix states. Note that merging power rails may reduce power optimization opportunities on the platform.
 10. For no M3 support on external devices, V3.3M/V1.8M will be OFF in Sx/M3
 11. This supply is expected to be 0V during states where SLP_S0# is asserted. It may be left on during this condition, but the SoC will not achieve its lowest power consumption. Specific power up latencies apply when exiting this state. Applicable to form factors with battery only (ie. AIO)Optional depending platform design; ON if AC is present
 12. NA
 13. Leakage on VCCIN_AUX is expected behavior when CORE_VID[1:0]=00; this leakage voltage may be as high as 1.15V during Sx and S0ix states.
 14. VCCIN_AUX may be ON in these power states if required by the PCH.
 15. VCC1P8A of Processor rail can be either merged with VCCPRIM1P8 rail of PCH or enabled by CPU_C10_GATE# using a power switch. Power gating option is preferred since additional power saving in C10 state is possible. All timing diagrams are drawn under the assumption power gating for VCC1P8 is used.

10.12.5 Power Sequencing Timing Diagrams Legacy Signals

Table 230. Legend for Signals in Transition Waveforms

Color/Legend	Comments
Signal Names	Timing of these signals is set by PCH or processor
Signal Names	Timing of these signals should be met by the platform
Signal Names	Voltage rails or chip-to-chip buses
Grey Highlight	Indicates unstable state

Figure 248. Timing Diagram for G3 to S0 [Deep Sx Platform] - 1 OF 2

Figure 249. Timing Diagram for G3 to S0 [Deep Sx Platform] - 2 OF 2


NOTE

General Note: Some of legacy signals shown in these diagrams are not available as hard signals when eSPI is used.

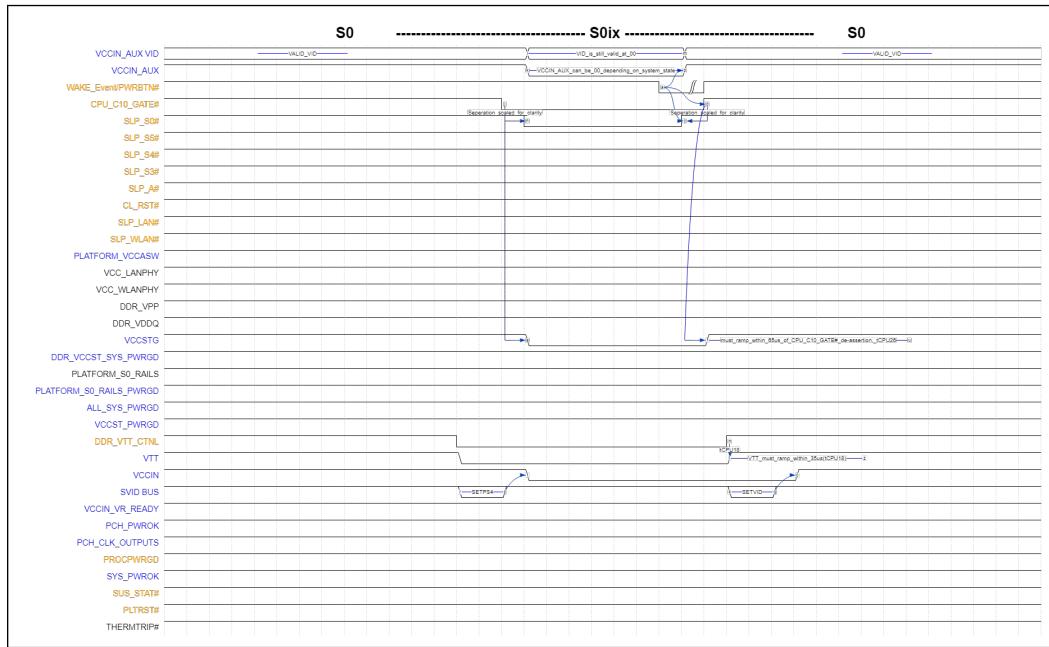
1. PCH will sample BATLOW# on the rising edge of DSW_PWROK for DSx Systems
2. Refer Rail-to-Rail Power Sequencing Requirement section on PCH prime rail-to-rail power and power down dependencies
3. NA
4. NA
5. For a DSx system, PCH will ignore SUSACK# until after SUSWARN# is valid.
6. EC must ignore SUSWARN pin even in DSx system on G3 exit.
7. SLP_A# always goes high with or before SLP_S3#. Depending on PCH settings SLP_A# may go high, then low, then high again all before SLP_S3#, but will go high no later than SLP_S3# on an Sx to S0 transition, or the Sx to S0 portion of G3 exit, Global Reset, and Deep Sx exit.

In the event of a global reset after SLP_A# is de-asserted, during the power up sequence, SLP_A# will assert and the controller monitoring the sleep signals should reset its timeouts.

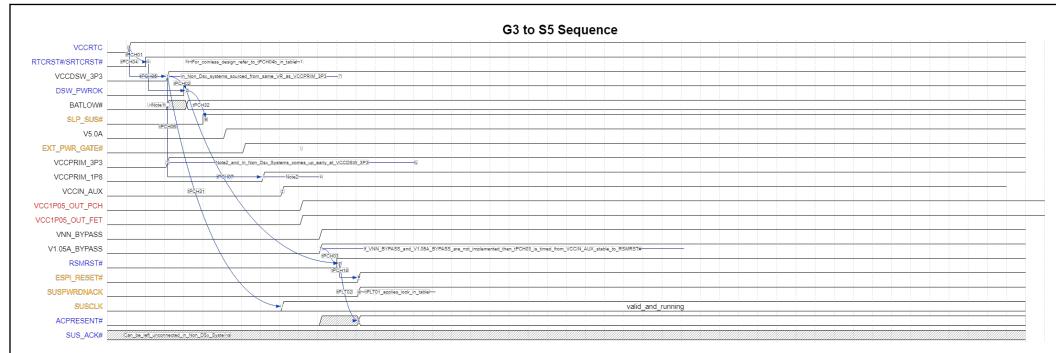
8. High for WoL=1, Low for WoL=0. SLP_LAN# may rise before, but no later than SLP_A#.
9. On first exit from G3, SLP_WLAN# de-asserts with SLP_S3# de-assertion
10. Delay between SLP_S5#, SLP_S4#, and SLP_S3# exaggerated for drawing purposes. If the system EC is driving these signals in ESPI mode the minimum delay between SLP_S3#, SLP_S4#, and SLP_S5# is not guaranteed
11. VCCST, and VCCSTG can remain powered during S4 and S5 pwr states for board VR optimization. VCCST, VCCSTG may also remain powered in S4 and S5 for debug purposes. Refer Platform Debug and Test Hooks chapter for more details. VCCSTG should only ramp up equal to or after VCCST.
12. Required with LPDDR4x, LPDDR4 and DDR4 memory configurations. For LP4/4x VPP, VDD2 and VDDQ timings, please refer to JEDEC Specification. There is no VTT for LP4/4x. It is applicable only for DDR4 memory based designs
13. VDDQ must ramp after VPP on DDR4 and LPDDR4 based systems, thus VDDQ may ramp up after SLP_S3# de-assertion due to VR ramp timing and configuration common
14. NA
15. NA
16. ALL_SYS_PWRGD is assumed to logically AND together the PWRGD signals for the major system power rails
17. VCCST_PWRGD can assert before or equal to PCH_PWRGD, but must never lag it. It is recommended that both VCCST_PWRGD and PCH_PWRGD include ALL_SYS_PWRGD in their generation. This ensures during failure events, both signals de-assert at the same time
18. PROCPWRGD is used only for power sequence debug and is not required to be connected to anything on the platform.
19. NA

20. The Platform should ensure that PCH_PWROK does not glitch when RSMRST# is de-asserted

Figure 250. Timing Diagram for S0-S0ix-S0



power-up

Figure 251. Timing Diagram for G3 to S0[Non-Deep Sx Platform]

S5 to S0 sequence is same as DSx Sequence (refer to DSx S5-S0 sequence diagram).

NOTES

1. PCH will sample BATLOW# on the rising edge of DSW_PWROK for DSx Systems
2. Refer Rail-to-Rail Power Sequencing Requirement section for details on PCH prime rail-to-rail power and power down dependencies.

Additional Notes:

Some of legacy signals shown in these diagrams, like SUSPWRDNACK, are not available as hard signals when eSPI is used, they are Virtual Wires.

The state of the SLP_A# and SUSPWRDNACK signals are used by the EC to determine if PCH requires the suspend-well to stay powered.

- SUSPWRDNACK
 - *Platform not supporting M3* - EC must keep SUS Rails powered ON if: SUSPWRDNACK is de-asserted. Else, EC has an option to do whatever it wants with the SUS Rails
 - *Platform supporting M3* - EC must keep SUS Rails powered ON if: SUSPWRDNACK is de-asserted **OR** SLP_A# is de-asserted **OR** it is the first 200mS after SUS Rails power has been applied. Else, EC has an option to do whatever it wants with the SUS Rails
- Primary rails and Deep Sx Rails should **never** be active while VccRTC rail is inactive.

Figure 252. Timing Diagram for S0/M0 to G3 [Deep Sx Platform]

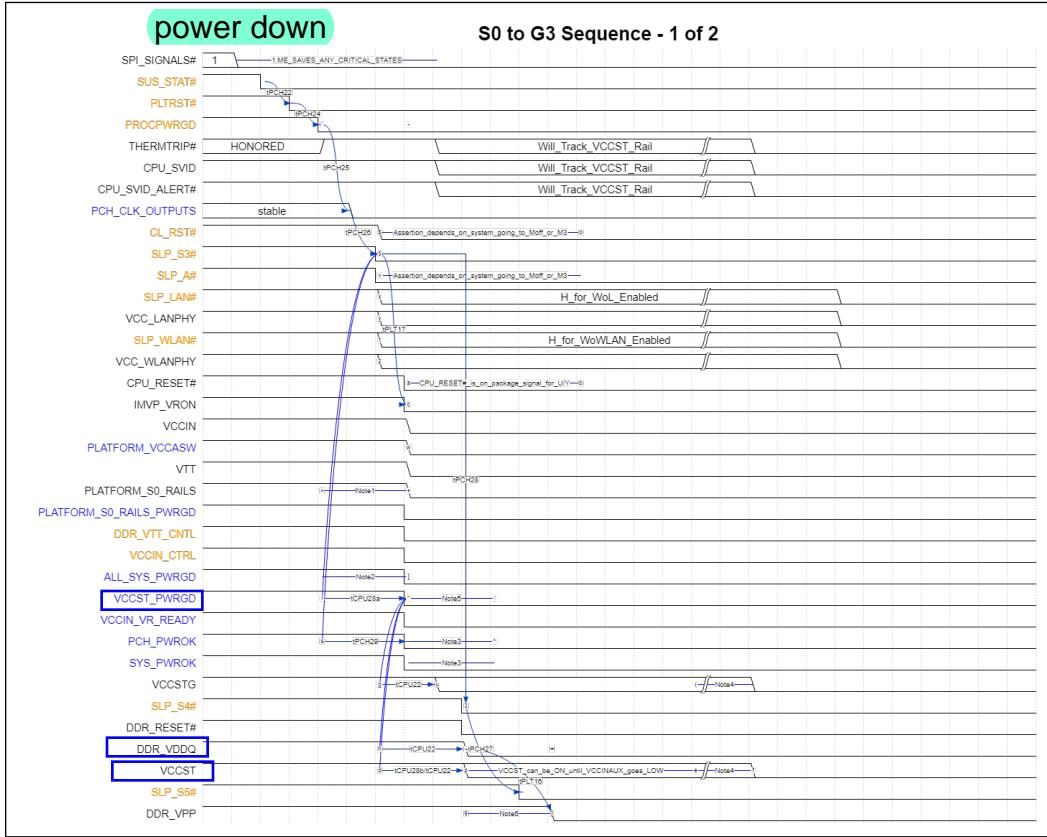


Figure 253. Timing Diagram for S0/M0 to G3 [Deep Sx Platform]

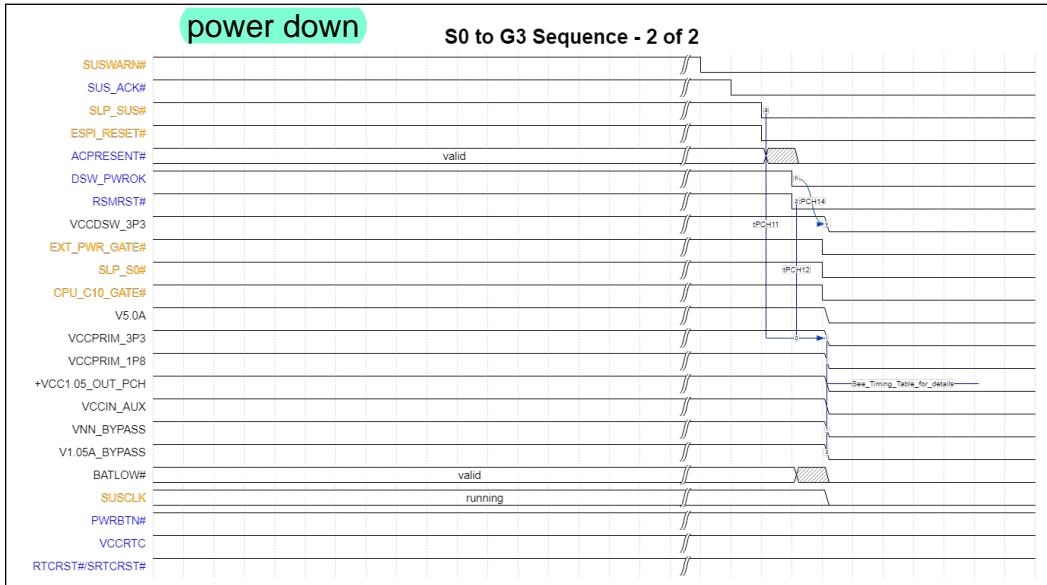


Figure 254. Timing Diagram for S0/M0 to G3 [Non Deep Sx Platform]

For S0 to G3 Sequence 1 of 2 Refer to DSx Sequence

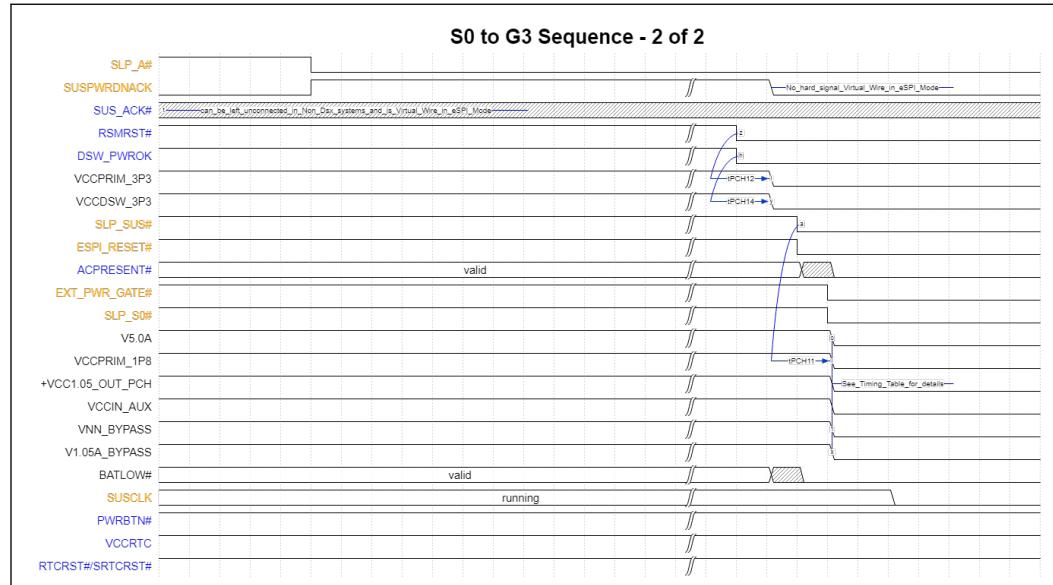
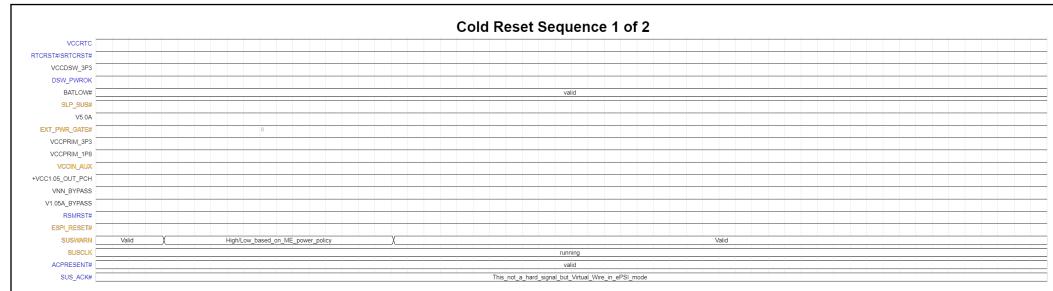
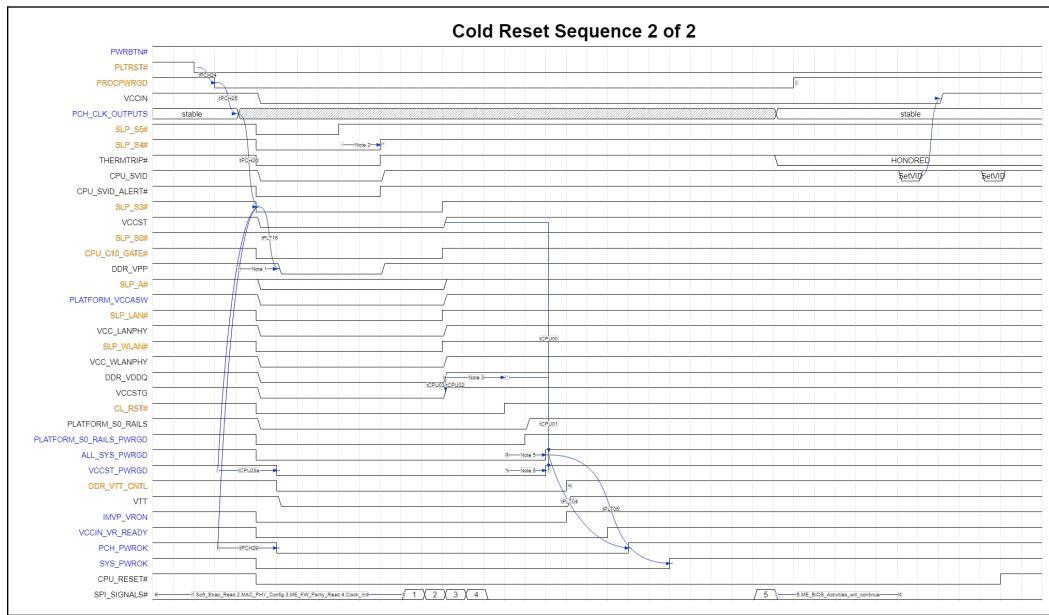


Figure 255. Timing Diagram for Cold Reset [Deep Sx Platform]





NOTES

1. Must ramp down AFTER VDDQ has ramped down
2. If the system EC is driving these signals in ESPI mode, based on the state of eSPI SLP Virtual Wires, the minimum delay between SLP_S3#, SLP_S4#, and SLP_S5# is not guaranteed
3. VDDQ must ramp after VPP on DDR4 and LPDDR4 based systems, thus VDDQ may end up after SLP_S3# de-assertion due to VR ramp timing and configuration
4. NA
5. ALL_SYS_PWRGD is assumed to logically AND together the PWRGD signals for the major system power rails
6. VCCST_PWRGD can assert before or equal to PCH_PWROK, but must never lag it. It is recommended that both VCCST_PWRGD and PCH_PWROK include ALL_SYS_PWRGD in their generation. This ensures during failure events, both signals de-assert at the same time IMVP VR_ON is recommended to be triggered by ALL_SYS_PWRGD in order to help minimize boot latency.

Figure 256. Timing Diagram for Warm Reset [Deep Sx Platform]

Power Sequencing Timing Requirements

The following table defines the timing parameters used in the timing diagrams in previous sections. The timing parameters are put into three categories: tCPU, tPCH, and tPLT. tCPU parameters are mostly required by the processor, tPCH parameters are required or controlled by the PCH, and tPLT parameters are mostly controlled by the platform.

The timing parameters are defined by Min, Max and Typical specifications. The Min and Max timings refer to the minimum or maximum timings allowed between the first and second signals in the Description column, as are the timing boundaries that must be followed. The Typical column refer to the typical timing values measured on Intel boards during validation, which do not imply a requirement but can be used as a reference.

Table 231. Platform Sequencing Timing Parameters

Label	Applicable SKU	Required By	Controlled By	Min.	Max.	Typical	Units	Note #	Description
tCPU00	All	CPU	PLT	2			ms	6, 7	VCCST, VCCSTG ramped and stable to VccST_PWRGD assertion
tCPU01	All	CPU	PLT	1			ms	6, 7	VDDQ ramped and stable to VccST_PWRGD assertion
tCPU02	All	CPU	PLT		No Limit		ms	43	VCCST, VCCSTG ramped and stable before VDDQ stable <i>Note:</i> tCPU02 and tCPU03 are mutually exclusive, and only one of the parameters needs to be met depending on the rail order of VCCST/VCCSTG and VDDQ

continued...



Label	Applicable SKU	Required By	Controlled By	Min.	Max.	Typical	Units	Note #	Description
tCPU03	All	CPU	PLT		No Limit		ms	43	VDDQ ramped and stable before VCCST, VCCSTG stable Note: tCPU02 and tCPU03 are mutually exclusive, and only one of the parameters needs to be met depending on the rail order of VCCST/VCCSTG and VDDQ
tCPU04	All	CPU	PLT	0			ns	31	VCCST must always ramp with or earlier than VCCSTG. VCCST >= VCCSTG at all times during ramp.
tCPU08	All	CPU	PCH	1			ms	8	PCH CLK outputs stable and CPU VRs stable/ready to PROCPWRGD assertion
tCPU16	All	CPU	PLT	0			ns	32	VCCST_PWRGD assertion to PCH_PWROK assertion
tCPU18	All	CPU	PLT	0	35		us	34	DDR_VTT_CNTL (was DDR_PG_CTL) assertion to DDR_VTT supplied ramped and stable while PLTRST = H (de-asserted).
tCPU19	All		CPU	0	100		ns	9	VccST_PWRGD assertion to DDR_VTT_CNTL (was DDR_PG_CTL) asserted.
tCPU20	All	CPU	PLT + PCH		500		ms		THERMTRIP# assertion until VCCIN VR is disabled and not sourcing power
tCPU21	All	CPU	PCH		1		ms		THERMTRIP# assertion until PCH issues global reset and assertion of SLP_SX signals
tCPU22	All	CPU	PLT	1			us	36, 37	VCCST_PWRGD de-assertion to either VDDQ, VCCST, VCCSTG below specification for normal S0 to Sx transitions. Recommend VCCST_PWRGD goes low with SLP_S3#
tCPU26	All			10	65		us		CPU_C10_GATE# de-assertion to VCCSTG, VCC1P8A stable, the rail must meet this max ramp time.
tCPU29	All	CPU	PLT		100		mV/us	13	Processor power rail instantaneous slew rate.
tPCH01	All	PCH	PLT	9			ms	1, 46, 47	VccRTC = 2.0V to the point in time where voltage on the RTC resets equals 0.65 times the voltage present on the VccRTC rail during ramp. This measurement should be made from VccRTC = 2.0V to the first of RTRCRST# or SRTCRST# reaching 0.65 * VccRTC .
tPCH02a	All	PCH	PLT	10	2000		ms		VccDSW stable (@95% of full value) to DSW_PWROK high. Applies to Systems that do not implement G3 Flash sharing
tPCH02b	All	PCH	PLT	10	See Note		ms	48	VccDSW stable (@95% of full value) to DSW_PWROK high. Only applies to Systems that implement G3 Flash sharing
tPCH03a	All	PCH	PLT	10	2000		ms		VccPrimary stable (@95% of full value) to RSMRST# high Applies to Systems that do not implement G3 Flash sharing
tPCH03b	All	PCH	PLT	10	See Note		ms	49	VccPrimary stable (@95% of full value) to RSMRST# high Only applies to Systems that implement G3 Flash sharing
tPCH04a	All		PCH	9			ms	29, 42	VccRTC stable (@90% of full value) to start of VccDSW voltage ramp for systems with coined RTC battery

continued...

Label	Applicable SKU	Required By	Controlled By	Min.	Max.	Typical	Units	Note #	Description
tPCH04b	All		PCH	30			ms	29, 42	VccRTC stable (@90% of full value) to start of VccDSW voltage ramp for systems with coinless RTC . Please refer to IB#549657 for Design considerations technical advisory document without RTC battery. Earlier this timing was referred as tPCH48.
tPCH05	All	PCH	PLT	1			us	42	RTCRST# high (voltage above ViH_min) to DSW_PWROK high (when voltage crosses ViL_max such that internally it might be resolved as a logic '1')
tPCH06	All	PCH	PLT	200			us		VccDSW 3.3 stable (@95% of full value) to VccPrimary 1.8V starting to ramp (for DSx or nonDSx configurations)
tPCH07	All	PCH	PLT	0			ms		DSW_PWROK high to RSMRST# high
tPCH08	All	PCH	PLT	1			ms		SLP_S3# de-assertion to PCH_PWROK assertion
tPCH09	All	PCH	PLT	2, 4, 8, 16			ms		SLP_A# high to PCH assuming ASW rails are stable (95% of full value)
tPCH11	All	PCH	PLT	100			ns		SLP_SUS# asserting to VccPRIM dropping 5% of nominal value
tPCH12	All	PCH	PLT	400			ns	5, 24	RSMRST# asserting to VccPRIM dropping 5% of nominal value
tPCH13	All	PCH		0			ms	14	RTCRST# asserting to VccRTC dropping 5% of nominal value (this applies only when RTC battery is removed)
tPCH14	All	PCH	PLT	400			ns	4, 5	DSW_PWROK falling to any of VccDSW, VccPrimary dropping 5% of nominal value
tPCH15	All	PCH	PLT		100		ms		SLP_LAN# (or LANPHYC) rising to VccLANPHY high and stable
tPCH18	All		PCH	90			us		De-assertion of RSMRST# to De-assertion of eSPI_RESET# on exit from G3 (DSx platforms only), or exit from DSx
	All		PCH	90			us		De-assertion of RSMRST# to De-assertion of eSPI_RESET# on exit from G3 (Non-DSx platforms only)
tPCH20	All	PLT	PCH	-100			ns		DDR_RESET# assertion to SLP_S4# assertion (The negative minimum timing implies that DDR_RESET# must either fall before SLP_S4# or within 100 nS after it).
tPCH21	All		PCH	Refer note 38			ms	38	Warm Reset PLTRST# assertion duration time
tPCH22	All		PCH	210			us		SUS_STAT# active to PLTRST# active. <i>Note:</i> Not applicable for eSPI systems.
tPCH23	All		PCH	60			us		SUS_STAT# de-assertion to PLTRST# de-assertion. <i>Note:</i> Not applicable for eSPI systems.
tPCH24	All		PCH	30			us		PLTRST# assertion to PROCPWRGD de-assertion

continued...



Label	Applicable SKU	Required By	Controlled By	Min.	Max.	Typical	Units	Note #	Description
tPCH25	All		PCH	10			us		PROCPWRGD de-assertion to CLKOUT_BCLK turning OFF. This timing is programmable (Range = 10us, 100us, 1ms, 10ms). Default is set to 10 us.
tPCH26	All		PCH	1			us		CLKOUT_BCLK turning OFF to SLP_S3# assertion
tPCH27	All		PCH	30			us	33	SLP_S4# assertion to SLP_S5# assertion
tPCH28	All		PCH	30			us	33	SLP_S3# assertion to SLP_S4# assertion
tPCH29	All		PCH	0			ms		SLP_S3# assertion to PCH_PWROK deassertion
tPCH31	All		PCH		tPCH02 + tPCH32		ms	15, 44	VccDSW3p3 ramped and stable until SUSCLK is stable. SUSCLK will start before the max timing, but will not be guaranteed valid until max timing is met
tPCH32	All		PCH	95			ms		DSW_PWROK assertion to SLP_SUS# de-assertion
tPCH33	All		PCH	0, 99			ms	16	PCH_PWROK high to PLTRST# de-assertion. This timing is set by the PCH via Soft strap settings
tPCH34	All	PCH	PLT		50		ms		Time from start of ramp of the first prim rail after SLP_SUS# de-assertion to completion of primary and bypass rail ramp.
tPCH35	All	PCH	PLT		See Note		ms	20, 49	SLP_SUS# low to PCH PRIMARY rails reaching 200mV or less.
tPCH36	All	PCH	PLT		100		mV/us		PCH Power rails instantaneous slew rate
tPCH41	All	PCH	PCH	1			ms		PCH_PWROK high to PCH clock outputs stable
tPCH43	All	PCH	PLT	95			ms	28	DSW_PWROK assertion to PWRBTN# monitored
tPCH44	All	PCH	PLT	500			us		tPCH09 expiring to CL_RST# high
tPCH45	All		PCH	1, 5, 50, 100			ms	39	Clock outputs stable to PROCPWRGD assertion to processor. Timing set by PCH via Soft Strap settings
tPCH46	All		PCH	1, 2, 5, 10			ms	39	PROCPWRGD and SYS_PWROK High to SUS_STAT# de-assertion. This timing is not applicable for eSPI systems. Note: Timing can be adjusted through the FIT tool
tPCH47	UP4		PCH	10.5	200		us	21, 51	De-assertion of EXT_PWR_GATE# until gated VCCMPHYGT_1P05 supply stable (@ 95% of full value)
tPCH48	UP4		PCH	10.5	200		us	50, 51	De-assertion of EXT_PWR_GATE2# until gated VCCPRIM_GATED_1P05 supply stable (@ 95% of full value)
tPCH49	UP4		PCH	6			mV/us	21, 50	MPHYGT_1P05/PRIM_GATED_1P05 Supply instantaneous slew rate
tPLT01	All		PCH	200			ms	2	RSMRST# de-assertion to SUSPWRDNACK valid. Timing set by PCH.
tPLT02	All			0	90		ms		RSMRST# de-assertion to ACPRESENT valid (not floating). Note: This is only for platforms not supporting Deep Sx state

continued...

20. No Ramp down requirement between rails. All VRs should stop regulation based on SLP_SUS# signal, or another common shutdown signal. Applies to all power down cases except PCH induced FIVR emergency shut down chase where SLP_SUS# goes low with to shut down PCH VRs.

49. Total exposure to any of the PRIMARY rails > 200mV, with RSMRST# low must not exceed 6 days of the life of the PCH. Failure to meet this requirement over the life of the system could result in reduce component reliability and lifetime.

Label	Applicable SKU	Required By	Controlled By	Min.	Max.	Typical	Units	Note #	Description
	All				0		ms		RSMRST# de-assertion to ACPRESENT valid (not floating) [For platforms supporting Deep Sx state]. Note: ACPRESENT is a powered input to the PCH while in the Deep Sx state (that is, when DSW_PWROK=1 and RSMRST#=0). The PCH implements a weak pull-down on ACPRESENT while RSMRST#=0 such that the input is observed as 0 if not driven actively by the board during this time. If driven actively by the board to 1 while in Deep Sx and if the PCH is enabled to treat the ACPRESENT indication as a wake event from Deep Sx, then the PCH will power up the Primary well.
tPLT04	All	CPU/PCH	PLT	1			ms	3, 19	ALL_SYS_PWRGD assertion to PCH_PWROK. This timing must be controlled on the platform.
tPLT05	All		PLT	Platform dependent	No limit			18	ALL_SYS_PWRGD assertion to SYS_PWROK. This timing must be controlled on the platform. SYS_PWROK provides a platform/EC mechanism to stall the PCH de-assertion of PLTRST# to the platform. Different platform components may have difference timing requirements from when their power rails are stable until their respective reset signal can de-assert. Platform designer should adjust this timing based on their specific platform requirements.
tPLT14	All		PCH	4			s		Power cycle duration time; programmable via PM_CFG.PWR_CYC_DUR bit.
tPLT15	All	PLT (MEM)	PLT		200		us	40	SLP_S4# assertion to VDDQ VR Enable Low [VDDQ VR disabled]. Memory dependent, refer JEDEC requirements
tPLT16	All	PLT (MEM)	PLT	30			ms	30	VDDQ ramp down to start of VPP ramp down when entering S4 and lower. This is not a Processor requirement but a recommended timing to help meet the JEDEC sequencing requirements. Platform designers are responsible for making sure their designs meet the power sequencing requirements for the memory technology used in their designs.
tPLT20	All	PLT (MEM)	PLT	2.5			ms	30	VPP ramped to VDDQ start of ramp when entering S0 at power up. This is not a Processor requirement but a recommended timing to help meet the JEDEC sequencing requirements. Platform designers are responsible for making sure their designs meet the power sequencing requirements for the memory technology used in their designs.

continued...



Label	Applicable SKU	Required By	Controlled By	Min.	Max.	Typical	Units	Note #	Description
tPLT17	All	CPU	PLT	0			us	35	IMVP VR_ON low to VCCST below 95% of its value
tPLT19	All	PLT	PLT		10		us	22	RSMRST_PWRGD# or SLP_SUS# assertion to RSMRST# assertion

Notes:

1. PCH Primary Rails must never be active while VCCRTC is OFF
2. RSMRST# should be de-asserted for at least 200ms before PCH will consider SUSPWRDNACK valid.
3. PCH can delay de-assertion of PLTRST# with a soft strap setting. Refer tPCH45 (t573) and tPCH46 (t1001) and the SPI Programming Guide for more details.
4. For catastrophic/surprise power failures only.
5. For surprise power down cases, if DSW_PWROK is de-asserted (tPCH16) before DSW3.3 **and** any other Prim rails droop out of spec, there is no risk of RTC corruption (assuming VCCRTC was previously powered and VCCRTC# is de-asserted) and this spec can be ignored
6. VCCST_PWRGD has no edge rate requirement, but edges must be monotonic.
7. VCCST_PWRGD must accurately reflect the state of VCCST and must not glitch when VCCST, VCCSTG or VDDQ power is applied. Additionally, VCCST_PWRGD must track to the state of PCH_PWROK on the platform. When PCH_PWROK de-asserts during S0 --> Sx transitions, then VCCST_PWRGD must also de-assert.
8. Processor's PROCPWRGD is not expected to be used externally on the platform, but is available for monitoring.
9. **DDR_VTT_CTL** will start to go high on VDD2 ramp with VCCST_PWRGD low for Sx to S0 power state transitions.
10. It is strongly recommended that the SLP_S3# be a qualifying input signal to ALL_SYS_PWRGD logic, which drives IMVP VR_ON inputs. Additionally, it is recommended that SLP_S3# also qualify the EN control to the VCCIO power supply
11. Max timing is only applicable during S0i3 exit if the voltage rail is actively power gated. Not applicable during Sx-S0 transition
12. NA
13. Applies to all CPU power supply rails
14. There are no perceived functional failures if this minimum is violated since all data is lost anyway when VccRTC drops. The only real requirement is to avoid driving a high signal level into the powered down device for an extended period of time, assuming an external device(s) on a different power supply then VCCRTC is driving RTCRST# or any other RTC well input signal.
15. SUSCLK is now powered in DSW well.
16. PCH can delay de-assertion of PLTRST# with a soft strap setting. If tPCH33 is not enabled, the platform is responsible for controlling the assertion timing of PCH_PWROK and SYS_PWROK in such a way that it satisfies platform component timing requirements of power stable to reset de-assertion. Refer SPI Programming Guide for more details. On a platform level this timing doesn't necessarily need to meet 99ms, PCH soft straps guarantees a min of 99ms.
17. NA
18. Example, if the platform only has mini-PCIe* devices requiring a 1 ms delay from power rails stable to PCIe* reset de-assertion, then the minimum value for ALL_SYS_PWRGD assertion to SYS_PWROK can be reduced to 1 ms. If SYS_PWROK asserts before PCH_PWROK, no additional delay will be added by the PCH to delay PLTRST# de-assertion and the final timing value will be subject to internal PCH timing parameters.
19. PCH_PWROK assertion assumes CPU and PCH voltage rails are ramped and stable.
20. No Ramp down requirement between rails. All VRs should stop regulation based on SLP_SUS# signal, or another common shutdown signal. Applies to all power down cases except PCH induced FIVR emergency shut down chase where SLP_SUS# goes low with to shut down PCH VRs.
21. Only applicable to platforms that implement external VCCMPHYGT_1P05 power gating. Does not apply to G3/DSx to Sx ramp up.
22. 10us max limit is an estimation and will vary based on platform VR and EC implementation details. Platform designers are responsible for ensuring that tPCH12 is not violated during normal DSx entries based on EC and VR response times to assertion of SLP_SUS# and/or RSMRST_PWRGD#
23. PCH will have a minimum of a 1ms delay from PCH_PWROK to assertion of PROCPWRGD. Therefore, the platform is only responsible to have these rails ready a minimum of 0ms before PCH_PWROK assertion.
24. Applicable to all power down sequences where Prime well rails are turned off and to emergency power loss scenario. Refer RSMRST#/DSW_PWROK Special Requirements section.
25. NA
26. NA
27. If VCCSTG and VCCIO supplies are merged together as a single supply, then the timing requirement is between VCCST/VCCSTG/VCCIO and VCCSA
28. Applicable to all G3 exits where GEN_PMCON_B.AG3E=1. The PWRBTN# must assert for at least 16ms after the minimum tPCH43 timing before PCH will detect PWRBTN# as a wake event
29. For scenarios where the coin-cell is dead or missing, the PCH can electrically tolerate the VCCRTC and VCCDSW/VCCPRIM 3.3V supplies ramping simultaneously with VCCDSW/VCCPRIM 3.3V leading VCCRTC by ~0.2 – 0.3V (ie. voltage drop associated with the coin cell Schottky diode circuit). These scenarios are not allowed as typical power sequencing scenarios, and are expected to be very infrequent. In these scenarios, glitch-free signal operation on DSW signals can not be guaranteed when the platform is configured for DSx mode. Likewise, glitch-free signal operation on DSW and PRIM well signals can not be guaranteed when the platform is configured for non-DSx mode
30. Generally, JEDEC specifications require VPP >= VDDQ for both power up and power down sequences. This timing parameter is a suggested value as ensure this requirement is met on power down, but system designer must account for other platform level considerations such as output decoupling, discharge circuits, etc that can alter this relationship.
31. VCCST supply is typically controlled by SLP_S3# only, and VCCSTG supply is typically controlled by SLP_S0# AND SLP_S3#. Since the timing delay between SLP_S3# and SLP_S4# deassertion during a S4/S5 à S0 transition can be small (defined by tPCH28), OEMs may need to take extra steps to ensure this timing is met. Example, power gates of equivalent slew rate may be required on both VCCST and VCCSTG supplies. Alternatively, a small timing delay path might be required on the SLP_S3# path of the VCCSTG power gate enable
32. **VCCST_PWRGD** should start to assert no later than when **PCH_PWROK** asserts; however, VCCST_PWRGD may lag completing its ramp with respect to PCH_PWROK by up to 20us
33. Not applicable to eSPI mode, refer to the eSPI Compatibility Specification (#508740) for eSPI implementations
34. Only applies to configurations that use DDR_VTT_VCTRL signal to enable/disable VTT VR and only applies after PLTRST# has de-asserted (ie, S0). VTT must be completely ramped and ready for full load at or before the de-assertion of PLTRST# on Sx -> S0 transitions.
35. Timing to VR being disabled, not until the VR is fully ramped down
36. S0 to Sx transition with VCCST powered in Sx state. In TGL platforms ST control changes to SLP_S3# OR VCCST_OVERRIDE .
37. S0 to Sx transition with VCCST unpowered in Sx
38. Recommend not to exceed 10ms delay with respect to SLP_S3#
39. During a warm reset sequence, the PCH sequencing flows will internally re-execute several timing parameters and tPCH45 and tPCH46 are two of the parameters that are re-executed. tPCH45 and tPCH46 are primarily for G3/DSx/Sx à S0 flows, but they are executed for cold boot, cold reset, and warm reset. The nominal setting for these timings is partially controlled by the setting of SPI soft straps associated with tPCH45 and tPCH46. The

Label	Applicable SKU	Required By	Controlled By	Min.	Max.	Typical	Units	Note #	Description
minimum possible PLTRST# assertion time will be equal to the soft strap settings for tPCH45 + tPCH46. The maximum PLTRST# assertion time is not guaranteed across all warm reset cycles, but the typical worst case assertion time is approximately equal to (tPCH45 soft strap setting) + (tPCH46 soft strap setting) + 20ms.									
40. This is a platform timing recommendation to help ensure that memory device power down sequencing requirements between VDDQ and VPP are likely to be met (refer tPLT16). This is not a requirement for the TGL Processor itself. Refer to the JEDEC LPDDR4 and DDR4 power down sequencing requirements for more details									
41. This parameter is programmable, refer to the Flash Descriptor Record in the SPI Programming Guide in the ME firmware package									
42. For the dead RTC coin cell and coin cell-less with depleted main battery RTC cases where the 3.3DSW ramps and powers the RTC well, the board designer should set the 3.3DSW rail stable to DSW_PWROK assertion delay (tPCH02) for = 30ms. This is required to ensure that SRTCST# and RTCRST# de-assert after VCCRTC is stable, but before DSW_PWROK assertion. Failure to meet this requirement may result in DSW_PWROK asserting with, or before, SRTCST# and RTCRST# reach VIH, which is a sequencing violation and can result in a non-booting system scenario. Refer Real Time Clock (RTC) Design Guidelines on page 123 and RTC External RTCRST# Circuit on page 126 and RTC External SRTCST# Circuit on page 127 for SRTCST# and RTCRST# RC timing network details									
43. tCPU02 and tCPU03 are mutually exclusive, only tCPU02 or tCPU03 needs to be met									
44. SUSCLK stable means the clock is toggling and is within its defined parameters. This timing spec is applicable to all G3 exits, excluding the G3 exit immediately following the ramp of VCCRTC, which may have a longer delay									
45. Refer tPCH43 for- DSW_PWROK assertion to PWRBTN# monitored timing aspect.									
46. C3 and C4 Capacitors used in the RTC external circuit should be evaluated with regards to aging, voltage and temperature characteristics to ensure reliable operation in the intended operating environment. See section 6.2.3 RTC External RTCRST# Capacitors.									
47. For measurement details, reference RTC Reset Timing Technical Advisory - Document #610459.									
48. Total exposure to DSW above > 200mV, with DSW_PWROK low must not exceed 6 days of the life of the PCH. Failure to meet this requirement over the life of the system could result in reduce component reliability and lifetime.									
49. Total exposure to any of the PRIMARY rails > 200mV, with RSMRST# low must not exceed 6 days of the life of the PCH. Failure to meet this requirement over the life of the system could result in reduce component reliability and lifetime.									
50. Only applicable to platforms that implement external VCCPRIM_GATED_1P05 power gating. Does not apply to G3/DSx to Sx ramp up.									
51. MPHYGT_1P05 and PRIM_GATED_1P05 gates are no longer mandatory to implement. However existing designs with gates can continue with them.									
Additional Notes:									
• Unless otherwise noted, all specifications in this table apply to all processor frequencies.									
• DDR_RESET# behavior does not directly track the state of VDDQ rail. For DDR_RESET# to assert properly on surprise power failure, the VDDQ VR pwrgood output should be an input into the logic generating PCH_PWROK									

10.12.6 DSW and PRIMARY Power up / Power Down Special Requirements

The PCH has lifetime exposure limits for the DSW and PRIMARY rails as follows:

- **DSW rail and DSW_PWROK:** The PCH requires that the sum total time with the VCC_DSW > 200mV **AND** DSW_PWROK low, during rail start up and shutdown, must not exceed 6 days of the life of the PCH.
- **Primary Rails and RSMRST#:** The PCH requires that the sum total time with any of the PRIMARY rails > 200mV **AND** RSMRST low, during rail start up and shutdown, not exceed 6 days of the life of the PCH.

Total exposure time will vary by platform implementation. Platform designs that support G3 flash sharing where the PCH is powered with RSMRST# low for more than 2 seconds must pay particular attention to the above requirements. Failure to meet this requirement over the life of the system could result in reduce component reliability and lifetime.

The following example demonstrates how to calculate DSW and PRIMARY rails total exposure time with their respective power goods (DSW_PWROK and RSMRST#) low over the product lifetime.

Example System Assumptions

DSx enabled design where system goes to DSx on AC and G3 on battery, **with G3 Flash sharing.**

System Characteristics as Determined by OEM for Target System Usage



VR Ramp & Power Good Delays (Power Up):

- VCCDSW Ramp Time: 1 ms
- DSW rail stable to DWPROK L-->H delay for design: 15 ms
- PRIMARY rail total ramp Time (start of ramp to rail ramp compete): 5 ms
- PRIMARY rails stable to RSMRST# L-->H delay for design: 20 seconds

Rail Discharge Characteristics (at DSx and G3 entry):

- DSW discharge time at G3 entry: 30 ms
- PRIMARY discharge time at G3 or DSx entry: 200 ms

DSx and G3 entry Frequency:

- Number of DSx entries per day: 10
- Number of G3 entries per day: 5

Target Product Life: 5 years (1825 days)

DSx/G3 Entry Exposure Time Calculation:

DSW per day power down exposure:

(5 G3 entries per day) * (30ms discharge per entry) = **150ms per day**

PRIMARY rail power down exposure on G3 & DSx Entry:

(10 DSx entries per day + 5 G3 entries per day) * (200ms discharge per entry) = **3s per day**

DSx and G3 Exit Exposure Time Calculation:

G3 Exit:

DSW Rail Power Up Exposure on G3 exit:

5 G3 exits per day * (1ms DSW ramp time + 15ms DSW stable to DPWROK H) = **80ms per day**

Primary Rails Power Up Exposure on G3 exit:

5 G3 exits per day * (5ms PRIMARY ramp time + 20s PRIMARY rails stable to RSMRST# H) = **100.025s per day**

DSx Exit:

Primary Rails Power Up Exposure on DSx exit:

10 DSx exits per day * (5ms PRIMARY ramp time + 20s PRIMARY rails stable time) = **200.05s per day**

Final Calculation for Total Life Exposure Time:**For DSW Rail:**

Total DSW per day exposure time = (G3 Entry Exposure Time) + (G3 Exit Exposure Time)

Total DSW rail per day Exposure Time = 150ms per day + 80ms = 230ms (0.23s) per day

Total DSW rail exposure over life of part: 0.23s per day * (1825 days)/86400(s/day) = **0.005 days**

DSW Final Result: DSW rail meets the 6 day max exposure requirement.

For PRIM Rails:

Total PRIMARY rail per day Exposure time = (DSx/G3 Entry Exposure Time) + (Total DSx and G3 Exit Exposure Time)

Total PRIMARY rail per day Exposure Time = 3s per day + (100.025s per day + 200.05s per day) = 303.075s per day

Total PRIMARY rail exposure over life of part: 303.075s per day * (1825 days)/86400(s/day) = **6.4days**

PRIMARY Final Result: PRIMARY rails fail to meet the 6 day limit.

NOTE

This must be corrected to ensure component reliability over the life of the part.

For this case the DSW rail meets the 6 day max limit, but the PRIMARY rails do not due to the long time spent with PRIMARY power and RSMRST# low for G3 FLASH sharing. To fix this the designer must go back and lower the PRIMARY stable to RSMRST# H time to meet the 6 day requirement. The designer must either:

Find a way to lower the per instance time with PRIMARY rails powered with RSMRST# low

OR

Add isolation to the shared SPI interface so that the PCH does not need to be powered during point in sequence when the EC is mastering the SPI bus to access the shared FLASH device.

Refer to Section <*Flash Sharing Topology in "SPI0 Flash" sheet provided with PDG package*> for more information on PCH SPI bus isolation with G3 FLASH sharing.

10.12.7 Rail-to-Rail Power Sequencing Requirements

PCH Rail-to-Rail Sequencing For Various Supplies

The following diagrams show the recommended rail-to-rail sequencing requirement for Deep Sx configuration(the below figure) and Non-Deep Sx configuration ([Figure 258](#) on page 460).

PCH architecture implements capabilities to help minimize rail-to-rail sequencing related problems. As long as the RTC 3.3 V supply is powered up properly in advance of DSW and Primary rails, isolation logic is active and will help achieve proper isolation between power rails during power up and power down sequences. The I/O interfaces will achieve proper isolation between the 1.8 V and 3.3 V Primary / DSW supplies. This does not guarantee glitch-free start up during transitions out of G3 and Sx, refer Bypass and Aux Configuration Registers in Intel® 500 Series Chipset Family Platform Controller Hub (PCH) EDS Vol 2 (#619207)

During scenarios where the RTC supply is not powered such as a dead coin cell scenario and it is instead ramping along the Primary /DSW supplies through external platform circuitry, proper isolation between the 1.8 V and 3.3 V Primary / DSW supplies cannot be guaranteed. It is thus possible that current inrush/back drive events could exist in these cases, but these scenarios are not a reliability risk to the PCH. It is, however, assumed that for general operation of PCH, the RTC supply should be ramped and valid before the DSW and Primary rails ramp. Ramping the RTC supply simultaneously with the VCCDSW_3p3 rail for Deep Sx systems and VCCPRIM_3p3/ VCCDSW_3p3 rails for Non-Deep Sx systems as a standard power up sequence for every G3 exit is not a valid configuration and is not allowed.

Figure 257. Rail-to-Rail Sequencing Requirement for Deep Sx Configured System

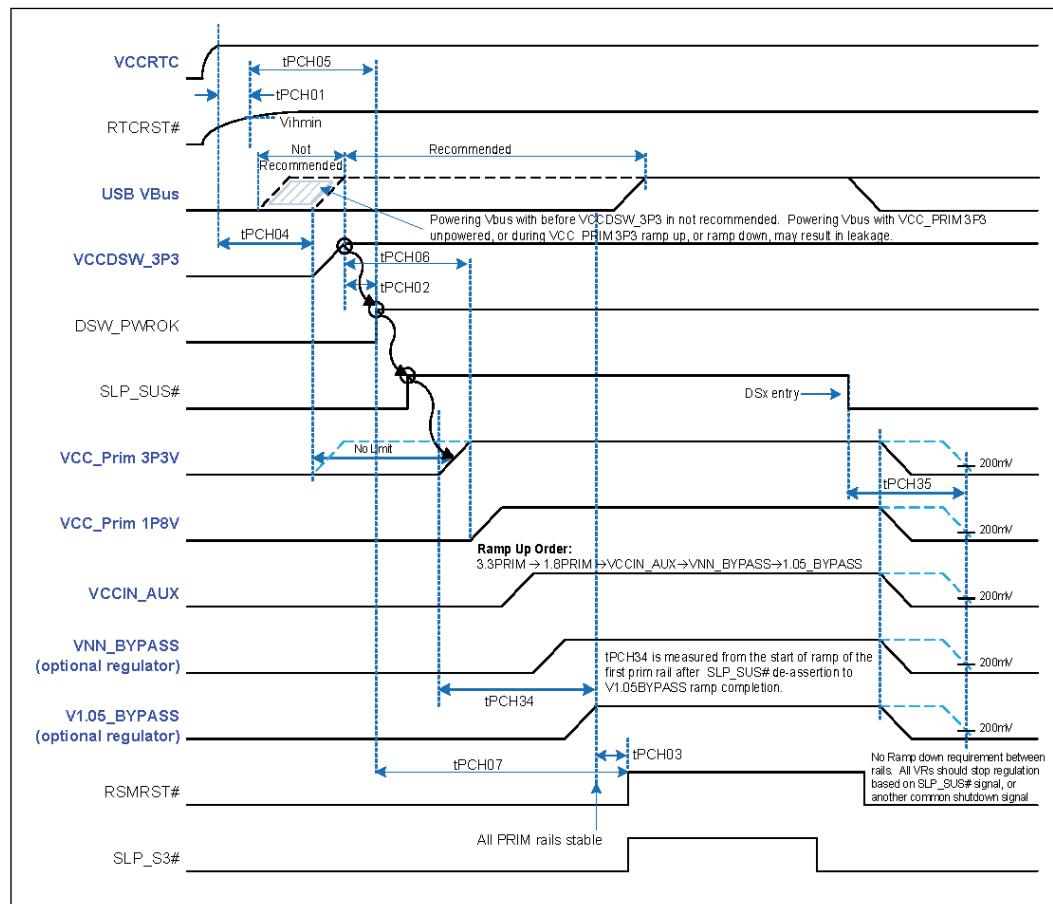
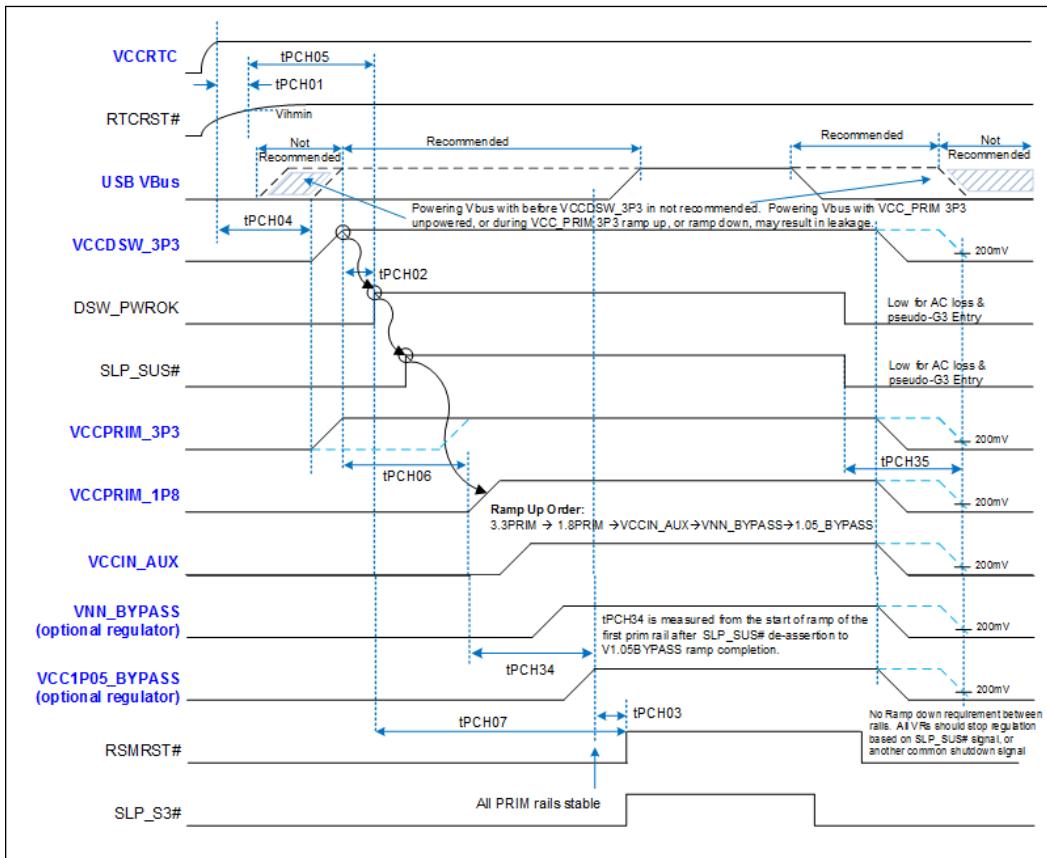


Figure 258. Rail-to-Rail Sequencing Requirement for Non-Deep Sx Configured System**Table 232. Rail-to-Rail Sequencing Requirements**

Rail 1	Rail 2	Ramp Up	Ramp Down
VCC_PRIM 3.3 V	VCC_PRIM 1.8 V	VCC_PRIM_3.3 ramps before VCC_PRIM_1.8	All rails must ramp down within tPCH35. Refer General Rail-to-Rail Sequencing
VCC_PRIM 3.3 V	VCCIN_AUX	3.3 V Primary rail ramp in advance of the VCCIN_AUX	All rails must ramp down within tPCH35. Refer General Rail-to-Rail Sequencing
VCC_PRIM 1.8 V	VCCIN_AUX	1.8 V Primary rail ramp in advance of the VCCIN_AUX. VCCIN_AUX can ramp with V1.8A for fixed 1.8V VCCIN_AUX design	All rails must ramp down within tPCH35. Refer General Rail-to-Rail Sequencing
VCC_PRIM 3p3/ VCC_PRIM 1.8V	Vbus	Vbus ramp after VCC_PRIM 3.3V reached 95% of their final value	Vbus ramp down with or at the same time that, the VCCDSW_3P3 is ramping down
VCC_PRIM 1.8V	VNN_BYPASS/ V1.05_BYPASS	VCC_PRIM 1.8V must ramp before the VNN_BYPASS/ V1.05_BYPASS	All rails must ramp down within tPCH35. Refer General Rail-to-Rail Sequencing

- Primary Rails and External USB Vbus Power Sequencing**

The Vbus provided to external USB ports can be used by USB devices to power their USB speed detect pull-up resistors. Some devices generate a local 3.3V power supply and pull up the D+ / D- lines with a 1.5k resistor. Other devices, per more recent USB ECRs, are allowed to pull up the D+/D- lines with Vbus voltage directly using ~7.5k resistor; implementation varies with device design.

If Vbus is powered while VCCPRIM_3p3 is not powered, and a device pulls either data line to 3.3V via its speed select pull-up resistor, the PCH will be exposed to leakage current through its un-powered USB 2.0 buffers. These leakage paths potentially impact both Non-Deep Sx and Deep Sx board designs alike.

The leakage paths exist only when Vbus is powered while VCCPRIM_3p3 and VCCPRIM_1.8V are unpowered.

Powering Vbus while the PRIM rails are not powered is permitted, but designers should be aware that leakage through the PCH may occur.

- **General Rail-to-Rail Sequencing**

All platform rails should follow a sequential order for system power on which are already called out in the [Table 231](#) on page 449.

On CPU side, for power up, VCCST always should ramp with or before VCCSTG. System should meet VDDQ rail order relative to VCCST/VCCSTG.

On PCH rails, for power up, all Primary rails should ramp within 80ms of each other. For power down, there are no explicit timing requirements/relationships between the various Primary Well voltages on power down, but it is required that all Primary Well voltage regulators stop regulation simultaneously based on SLP_SUS# assertion. The power down sequencing should not be staggered from one VR to the next. Natural variance between VRs disabling due to differences in voltage regulator disable time is acceptable on the order of tens of microseconds. It is required that all rails ramp down within tPCH35.

For surprise power down requirements, refer [Surprise Power Down Sequencing Considerations](#) on page 462 for assumptions.

RSMRST#/DSW_PWROK Special Requirements

1. When the system is powered off (G3), DSW_PWROK and RSMRST# must not glitch from their Low states while the corresponding PCH rails are not powered or are below normal operating voltage specifications to ensure RTC corruption does not happen.
2. When a system is in S0-S5 state and not entering a Deep Sx, the RSMRST# may only be driven low if the DSW_PWROK is also driven low at the same time (by an external controller such as EC). Failure to meet this requirement may result in unexpected PCH behavior, including failure to boot (which may only be recovered through a G3 cycling) and possible RTC corruption.
3. RSMRST# must always be driven low before any of the PRIMARY rails fall below the lower end of their tolerance band. This is true for all power states transitions including emergency power loss.

DSW_PWROK Min LOW Time during Cycling

DSW_PWROK de-assertion min low time is 100ms. The platform must then follow the normal power sequence (normal G3 exit sequence). Platform must ensure that all the PCH PRIM rails reach 200mV or less, within tPCH35 timing of the falling edge of SLP_SUS#. Refer to [Table 231](#) on page 449.

RTEST# (RTCRST#) Min LOW Time during Cycling

For RTEST# (RTCRST#) de-assertion min low time is 100ms, after the min time platform must follow the normal power sequence (normal G3 exit sequence).

Surprise Power Down Sequencing Considerations

Surprise power down events will be treated slightly differently on Tiger Lake PCH compared to past generation platforms. The main goal of the various power down timing specification such as tPCH10, tPCH12, tPCH14, etc., is to ensure proper isolation between the associated power well and the RTC well to guarantee that RTC contents are not accidentally corrupted. There are many events that could cause a surprise power down. The following is a short list of some events, but is not exhaustive:

- VR failure (over current, over voltage, IC failure, etc.)
- AC removal with no DC Battery present
- Removal of the primary battery

Properly designed platforms generally should not be experiencing VR failures of any kind; therefore the focus of this section is on the unexpected power removal caused by user interaction, which could be an end user, factory technicians and system level induced power down that removes all power from PCH.

To ensure RTC is not corrupted, the platform must de-assert the appropriate power good signals BEFORE the rails go out of their defined tolerance range. This implies that the platform should monitor the highest voltage available which is usually the main power supply like the battery voltage to determine when it has dropped too low and VR failure/shutdown is eminent. At that point, the PCH power good signals (PCH_PWROK, RSMRST#, DSW_PWROK) should be driven Low before their associated rails turn off and droop below the defined tolerance.

Finally, regardless of the reason, any time that SLP_SUS# is taken low, the platform is responsible for ensuring that tCPH10, 12, and 14 are met. Additionally, once RSMRST# is taken low, the platform is responsible for taking the system to G3 before attempting a restart. See RSMRST# / DSW_PWROK Special Requirements for details regarding RSMRST# and DSW_PWROK requirements.

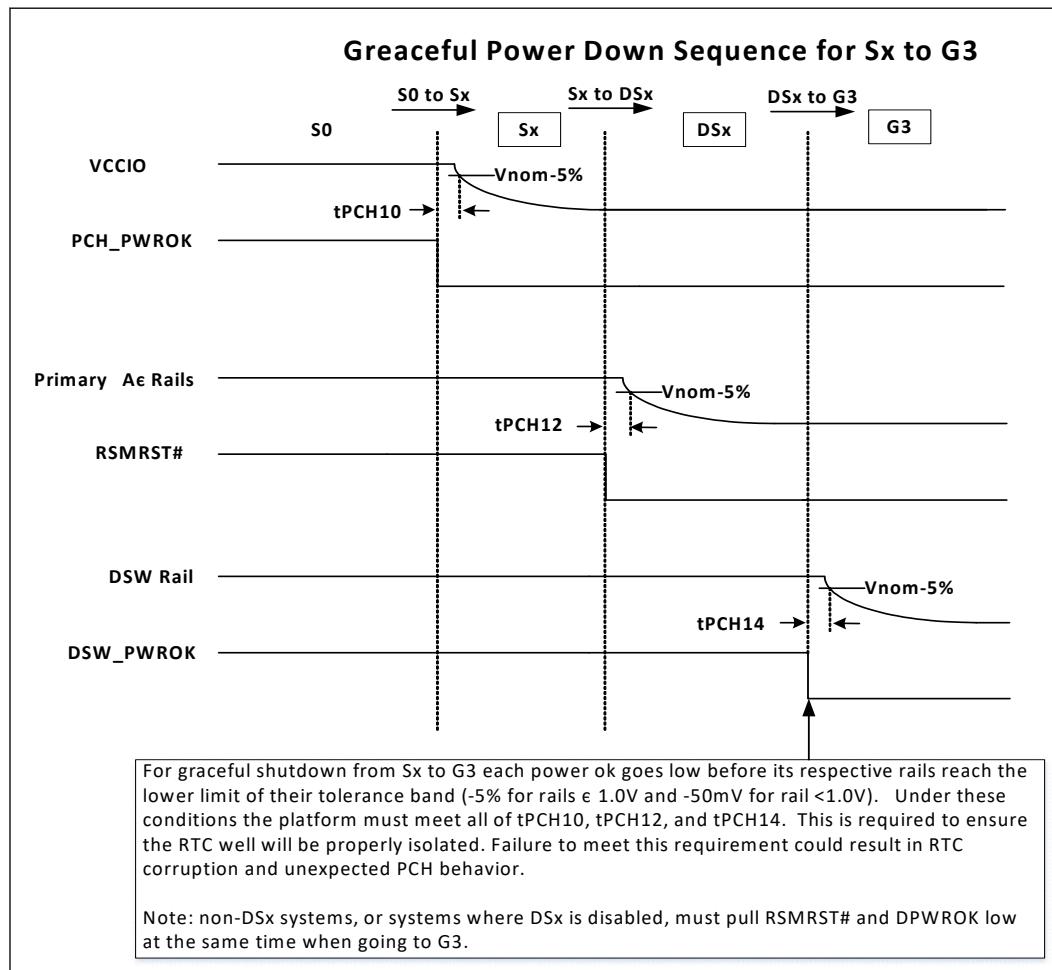
NOTE

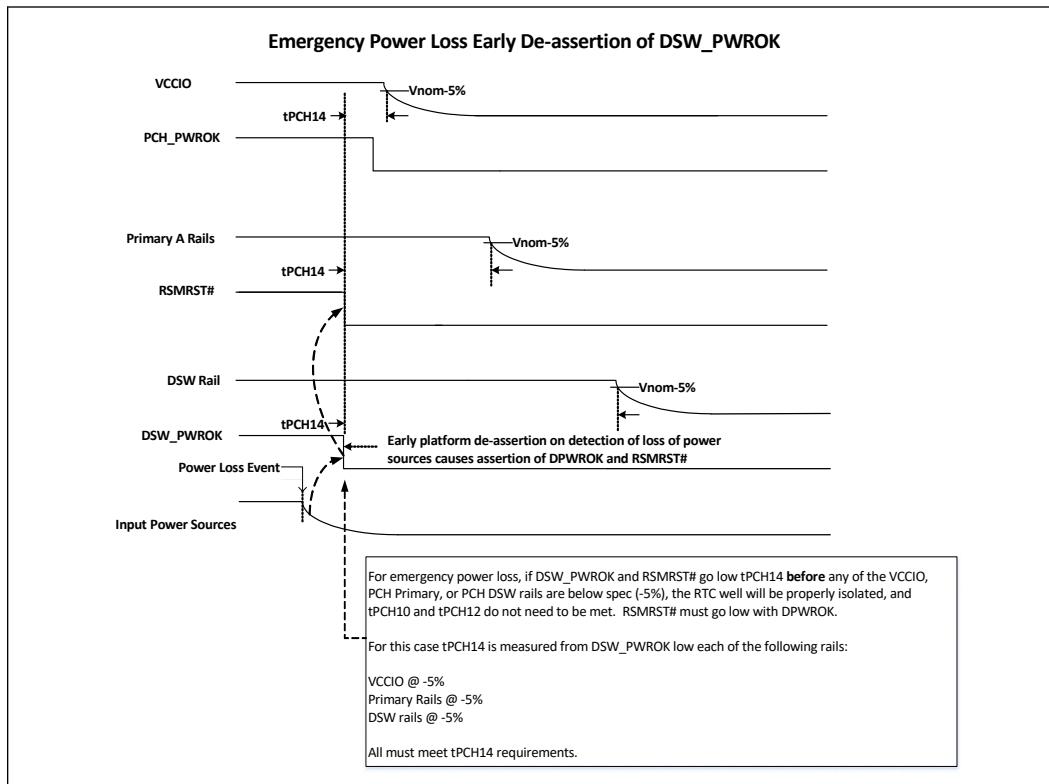
For these cases, de-assertion of DSW_PWROK and RSMRST# signal is sufficient to activate isolation logic for ALL power wells, thus guaranteeing RTC corruption cannot happen, as shown in the below figure.

NOTE

As stated previously in [RSMRST#](#) and [DSW_PWROK Power Down Design Requirement](#) on page 431, it is not permissible to take DSW_PWROK low without taking RSMRST# low when NOT entering a DSx state. If the platform is designed to take DSW_PWROK low on emergency power loss, it must also take RSMRST# low at the same time, refer the figure below.

Figure 259. DSW_PWROK Requirement for Power Loss





eSPI Considerations for Sequencing

In general, eSPI support does not have any major impact to power sequencing requirements. However, there are a few behavioral differences worth noting that could have side effects to platform behavior that should be considered.

With eSPI, the EC may or may not take in or drive physical pins that historically would have been supported by the PCH. The values of these signals (ex. SLP_S4#) are tunneled over the eSPI interface between the PCH / EC as virtual wires. If the EC is using a combination of hard signals from the PCH and Virtual Wires over eSPI, there is no guarantee of relative timing between the two signal types, other than the hard signals will change state first.

Only the physical signals directly on the PCH are guaranteed to refer the PCH-defined timing relationships. Example, SLP_S4# de-assertion. SLP_S3# de-assertion relationship is defined as 30us for the physical pins but could be shorter for the virtual wire relationship on the EC.

SUSPWRDNACK at RSMRST# de-assertion for eSPI designs please ignore eSPI SUSPWRDNACK Virtual Wire (VW) until after receiving the first 1->0 transition, then after that it is valid.

10.12.8

Glitch Free Design Options and Recommendations

Platform signals are not guaranteed to be glitch free on power up of the DSW and PRIMARY rails. The platform may mitigate glitches on key signals by adding a pull up or pull down as described [PCH Signal Glitch Free Implementation Requirements](#) on page 195

10.13 S0ix State Definitions

NOTE

Refer Tiger Lake Platform S0ix Technical Advisory (#630692) for more information

S0ix is a system low power state. The PCH is self-optimizing for the lowest power when in S0ix. But, the PCH needs platform information to make the best decisions when self-optimizing.

This section describes how the internal PCH states and platform configuration combine to optimize PCH power when in S0ix.

S0ix is used to indicate when the CPU + PCH has achieved a low power state

- S0i2 denotes a state in which certain IPs may be active
- S0i3 denotes a state in which all IPs are idle
- In certain scenarios, it is possible to dynamically move between S0i2 & S0i3
 - This is also referred to as **Direct Switching**, and allows switching between S0ix states.
 - To enable Direct Switching, external bypass VRs are needed

S0ix Substates are additional power actions taken when certain criteria are met.

- Certain device states, latency requirements , and platform components may be required

Figure 260. S0ix State Definitions

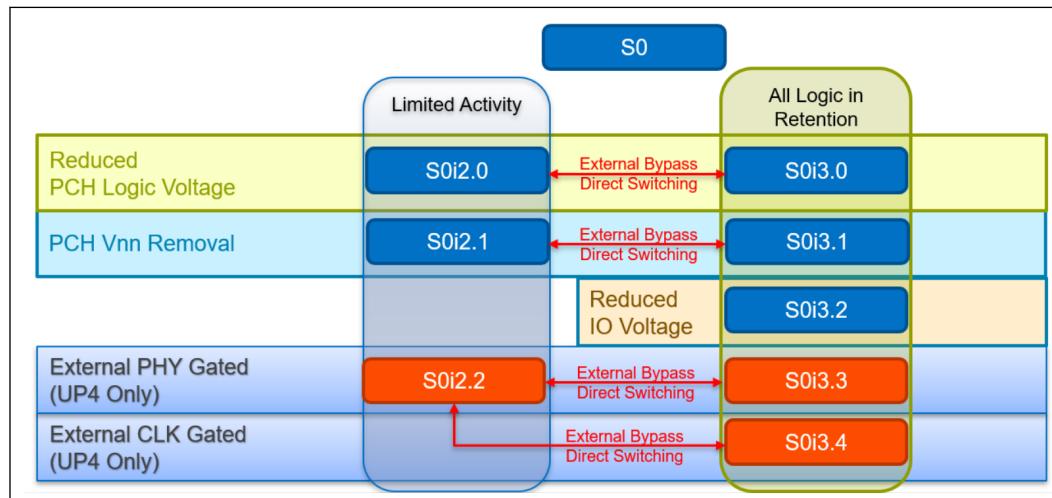


Table 233. S0ix State Actions and Power Impact

State	Actions
S0ix.0	Vnn voltage reduction
S0ix.1	Additional Vnn domains powered off
<i>continued...</i>	



State		Actions											
S0i2.2		Platform phy PG enabled (UP4) (~2 mW Power Benefit)											
S0i3.2		V1p05 voltage reduction											
S0i3.3		Platform phy PG enabled (UP4) (~2 mW Power Benefit)											

Feature		Incremental Power Benefit											
* Vnn Ext Bypass @ 0.78V		See notes											
*V1p05 Ext Bypass @ 1.05V		See notes											

NOTES

- For the bypass rails, the value of the bypass rails is highly dependent on the load on those rails. One could assume an ~60-75% efficiency if internal regulators are used. (Voltage margining for bypass rails can offer better power savings in S0ix)
- UP4 platform external PHY PG refers to VCCMPHYGT_1P05 (+VCC_MODPHY), which is gated by PCH EXT_PWR_GATE# (GPP_F20)
- UP4 External CLK PG refers to VCCPRIM_GATED_1P05 (+VCC_ISCLK), which is gated by PCH EXT_PWR_GATE2# (GPP_F21).

Table 234. Basic Minimum Requirements Table (PCH)

CPU/Devices	CPU	UFS	LPSS	XHCI*	XDCI*	THC	GBE Nahum	CSME	AUDIO	ISH	CNVI	DISPLAY	PCIe	SATA	ModPHY
Condition	C10	D3	D3	D3	D3	D3	Power Gated	Power Gated	Dx	D0lx	Active/Idle	ON/OFF	D3/L23	D3	CORE/SUS PG
S0i2.0	C10	D3	D3	D3	D3	D3	Power Gated	Power Gated	D0 (WoV / LPA)	D0l1	Active	ON	L1.1	D0	Core PG
S0i2.1	C10	D3	D3	D3	D3	D3	Power Gated	Power Gated	D0 (WoV / LPA)	D0l1	Active	ON	D3/L23	D3	Core PG
S0i2.2	C10	D3	D3	D3	D3	D3	Power Gated	Power Gated	D0 (WoV / LPA)	D0l1	Active	ON	D3/L23	D3	SUS PG
S0i3.0	C10	D3	D3	D3	D3	D3	Power Gated	Power Gated	D3	D0l2	Idle	OFF	L1.1	D0	Core PG
S0i3.1	C10	D3	D3	D3	D3	D3	Power Gated	Power Gated	D3	D0l2	Idle	OFF	D3/L23	D3	Core PG
S0i3.2	C10	D3	D3	D3	D3	D3	Power Gated	Power Gated	D3	D0l2	Idle	OFF	D3/L23	D3	Core PG
S0i3.3	C10	D3	D3	D3	D3	D3	Power Gated	Power Gated	D3	D0l2	Idle	OFF	D3/L23	D3	SUS PG

Notes:

- PCIe L23 State can be achieved by enabling D3 Cold support
- PCIe L1.1/L1.2 state can be achieved by enabling D3 Hot support
- Not all constraints (e.g., LTR, timers, internal hysteresis, etc.) are listed

Table 235. Basic Minimum Requirements Table (CPU)

CPU/Devices	CPU		TCSS TBT DMA		TCSS XHCI		TCSS PCIe		PEG (CPU PCIe)	
Condition	C10		D3		D3		D3/L23		D3/L23	
S0i2.0	C10		D3		D3		L1		L1.2	

continued...

CPU/Devices	CPU	TCSS TBT DMA	TCSS XHCI	TCSS PCIe	PEG (CPU PCIe)
S0i2.1	C10	D3	D3	D3/L23	D3/L23
S0i2.2	C10	D3	D3	D3/L23	D3/L23
S0i3.0	C10	D3	D3	L1	L1.2
S0i3.1	C10	D3	D3	D3/L23	D3/L23
S0i3.2	C10	D3	D3	D3/L23	D3/L23
S0i3.3	C10	D3	D3	D3/L23	D3/L23

Notes:

- PCIe L23 State can be achieved by enabling D3 Cold support
- PCIe L1.1/L1.2 state can be achieved by enabling D3 Hot support
- Not all constraints (e.g., LTR, timers, internal hysteresis, etc.) are listed

S0ix State Quick Reference Guide

For UP3

- With External Bypass VRs and Direct Switching supported, the lowest S0ix state that is recommended (for optimal power) is S0i3.1. If you do not use CNVi or ISH, S0i3.2 can be achieved.
- Without the External Bypass VRs and/or Direct Switching, the lowest power state that is recommended is S0i2.1.
- UP3 is not able to achieve S0i2.2 and S0i3.3.

For UP4

- With External Bypass VRs and Direct Switching supported, the lowest S0ix state that is recommended (for optimal power) is S0i3.3. This is assuming no USB3 Type A ports, no C-link used, no SMBus features used and PHY is power gated.
- Without the External Bypass VRs and/or Direct Switching, the lowest power state that is recommended is S0i2.2. This is assuming no USB3 Type A ports and PHY is power gated. Otherwise, can achieve S0i2.1.
-

NOTE

It may be possible to achieve lower power states than recommended. Due to stability, however, the recommended power state may have a lower overall power than would be observed using the lowest possible state

S0ix Optimization Guidelines

Based on the reply to these questions about your platform, follow the optimization guidance instructions below.

Q 1	Does your platform use Integrated Connectivity (CNVi)?	Yes/No
Q 2	Does your platform use the Integrated Sensor Hub (ISH)?	Yes/No
Q 3	Does your platform have USB3 Type A ports?	Yes/No
<i>continued...</i>		

Q 4	Does your platform use C-Link with an External Wi-Fi card?	Yes/No
Q 5	Does your platform use any of the following SMBus features? <ul style="list-style-type: none"> • SMBus Controller as Slave • SMBus Wake • SMBus Device Hot Plug • TCO Slave 	Yes/No
Q 6	Does your platform have the External PHY PG (UP4 Only)?	Yes/No

Instructions

- If the cell is marked "YES": Enable the state in the below row, ONLY if you answered "YES" to the question in the above column
- If a cell is marked "NO": Enable the state in the below row, ONLY if you answered "NO" to the question in the above column
- If a cell is marked "-": Ignore this question when deciding whether to enable or disable the state in this row.

Table 236. External Bypass Supported with Direct Switching

State Enable	Q1 CNVi	Q2 ISH	Q3 Type A	Q4 C-Link	Q5 SMBus	Q6 PHY PG
S0i2.0	Always Enable					
S0i2.1	Always Enable					
S0i2.2	-	-	NO	-	-	YES
S0i3.0	Always Enable					
S0i3.1	Always Enable					
S0i3.2	NO	NO	-	-	-	-
S0i3.3	-	-	NO	-	-	YES

Table 237. No Direct Switching or No External Bypass Supported

State Enable	Q1 CNVi	Q2 ISH	Q3 Type A	Q4 C-Link	Q5 SMBus	Q6 PHY PG
S0i2.0	Always Enable					
S0i2.1	Always Enable					
S0i2.2	-	-	NO	-	-	YES
S0i3.0	NO	NO	-	-	-	-
S0i3.1	NO	NO	-	-	-	-
S0i3.2	NO	NO	-	-	-	-
S0i3.3	NO	NO	NO	-	-	YES

Voltage Margining

Voltage margining allows bypass VRs to reduce their voltages depending on S0i3.x states. To enable this feature, VNN_CTRL and V1P05_CTRL GPIO pins should be configured. This feature can reduce power in S0ix states.

Voltage Margining Feature	Voltage Reduction
Vnn Ext Bypass margining using VNN_CTRL	0.78V -> 0.7V
V1p05 Ext Bypass margining using V1P05_CTRL	1.05v -> 0.96v

- Voltage Rail Sequencing Requirements

The block diagrams and timing diagrams in this document are references to help demonstrate the timing requirements for the platform, and the actual timing specifications that need to be met by the system are defined in [Power Sequencing Timing Requirements](#) on page 449. Depending on the system architecture, all applicable timings need to be met to ensure proper functionality of the CPU/PCH.

10.12.1 Key Changes to Tiger Lake Sequence Architecture

RSMRST# and DSW_PWROK Power Down Design Requirement

DSx and Non-DSx platforms are required to take RSMRST# and DSW_PWROK low at the same time when not entering a DSx state. Taking RSMRST low without taking DSW_PWROK low is not permitted when not entering DSx states. Refer [DSW/PRIM Rail Architecture in DSx and Non-DSx Designs](#) on page 432 and [RSMRST#/DSW_PWROK Special Requirements](#) on page 461

Design Consideration for VCCST Rail Enable Logic

VCCST can be turned OFF even if VCCIN_AUX is ON.

Refer [Additional Power Savings with Respect to VCCST Rail](#) on page 429 for details around VCCST rail state requirements.

Figure 243. VCCST Enable Logic

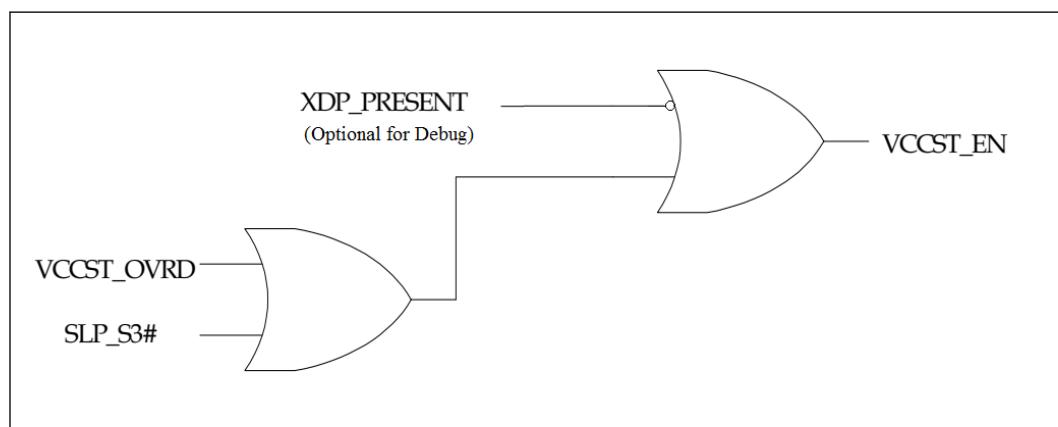
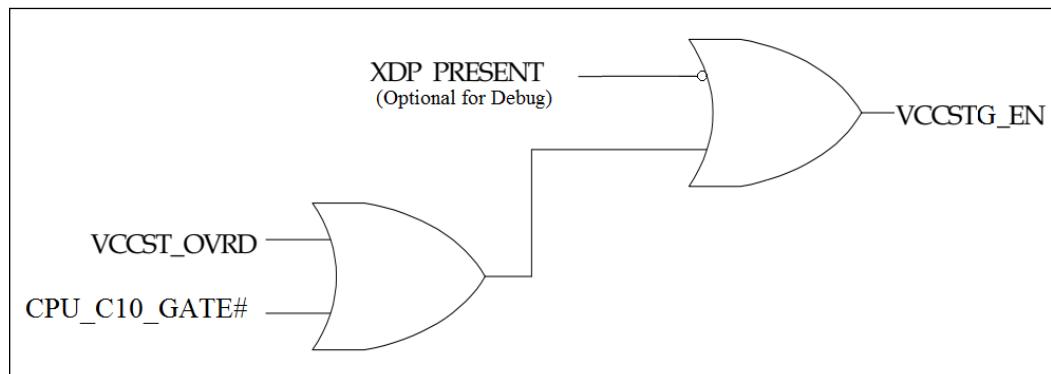


Figure 244. VCCSTG Enable Logic**DSW/PRIM Rail Architecture in DSx and Non-DSx Designs**

DSx and Non-DSx designs are similar in term of power rail architecture, rail, control, and power good signaling.

- SLP_SUS# controls part of all of the PRIM rail enabling in both systems
- DSW_PWROK and RSMRST# are always separate power good signals

Refer the Figure below and [Figure 246](#) on page 434 for details.

Figure 245. Tiger Lake DSx System Architecture Block Diagram

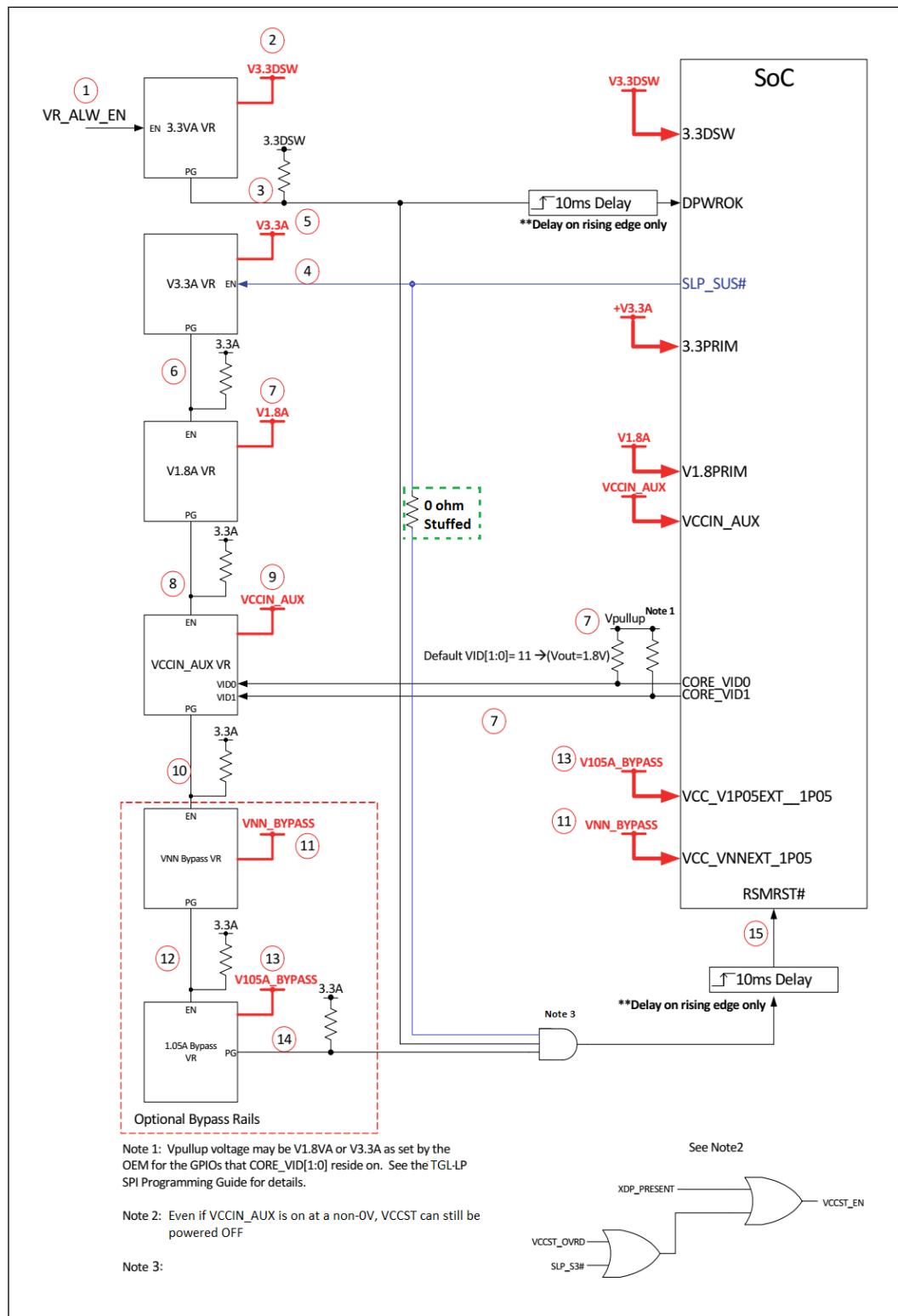
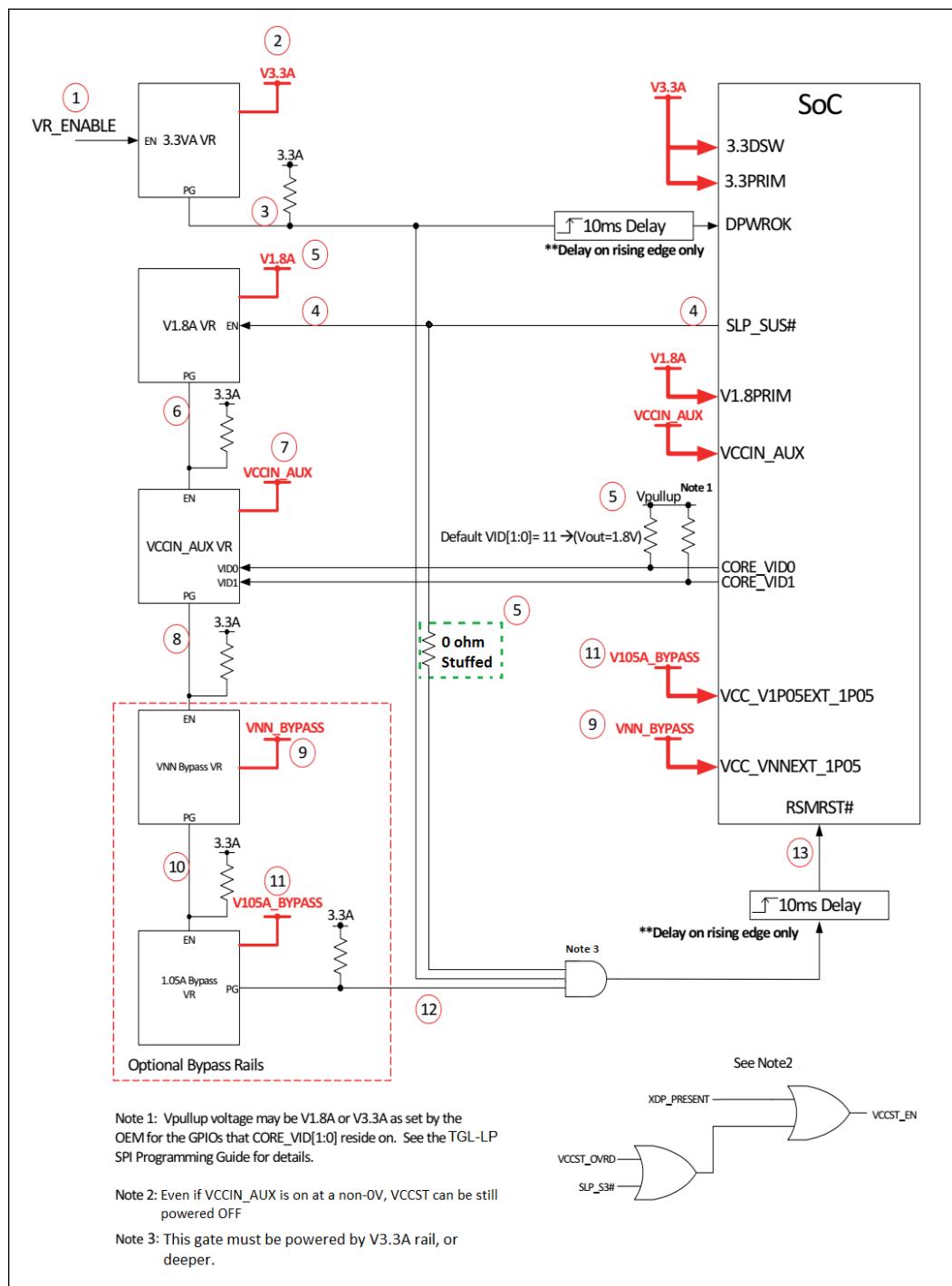


Figure 246. Tiger Lake Non-DSx System Architecture Block Diagram



10.12.2 PCH_PWROK, SYS_PWROK and Other PWRGD Signal Generation

The power sequencing consists of multiple stages of power state transitions. During each stage, different VRs will be turned on through various control signals from the processor such as the SLP_SUS#, SLP_S5#, SLP_S4#, SLP_S3#, SLP_S0# and CPU_C10_GATE#. In response, the platform will ramp the required voltage rails in the required order and then asserted the various powergood signals required by the processor, and other platform components, after the necessarily timing requirements have been met. The below figure shows a high-level representative control signal and the powergood logic diagram for generating the PCH_PWROK and SYS_PWROK signals. This figure shows the recommended power-on sequencing flow steps from SLP_S4# de-assertion until PLTRST# de-assertion for the Tiger Lake platform.

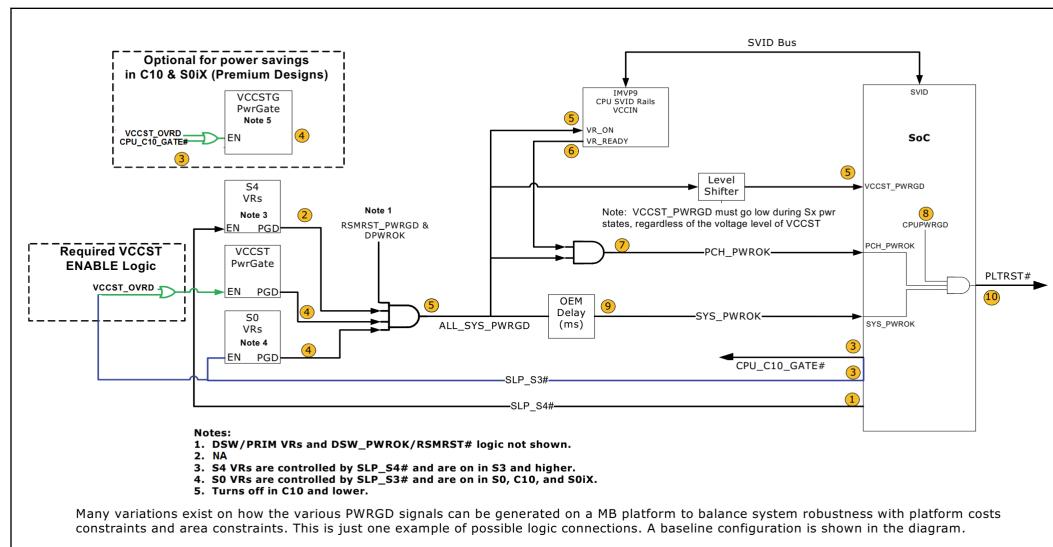
During G3/DSx, S5 to S0 transitions, the platform will need to generate the DSW_PWROK and RSMRST# signals. In this phase of the power up sequence, the DSW and Primary power well voltage rails are ramped to the processor.

During S5 to S0 and DSx to S0 transitions, the platform will need to generate the VCCST_PWRGD, PCH_PWROK and SYS_PWROK signals to the processor. In this phase of the power up sequence, the platform and CPU S0 rails are ramped up.

The PCH_PWROK signal is expected to be asserted by the platform to indicate to the processor that all required CPU voltage rails are up and stable, and that the processor may continue the final boot sequence leading up to PLTRST# de-assertion, such as starting to turn on clocks and executing other internal pre-reset activities.

SYS_PWROK is expected to be asserted by the platform to indicate that the system and all of its non-CPU components are ready for PLTRST# de-assertion. During power state transition to S0, the SYS_PWROK signal is the final platform controlled hardware gate before PLTRST# de-assertion. Platform designers may optimize when the SYS_PWROK signal is asserted with respect to the PCH_PWROK signal to help optimize overall boot latency, depending on system and component timing requirements.

Figure 247. Premium PWROK Generation Flow Diagram





SILEGO

SLG7NT4192

Power Good Generator Logic

Block Diagram

