T23x BCT

Deployment Guide

DU-10990-001 | June 2022

Document History

DU-10990-001

Version	Date	Authors	Description of Change
01	June 2022		Initial release

Table of Contents

Chapter 1. Introduction

Boot Configuration Table (BCT)is a set of platform-specific configuration data that is consumed by a boot component. BootROM and MB1 consume BCT in binary form, which is generated by parsing device tree source configuration files (with dts file extension) by `tegrabct_v2`.

Starting from T23x, the config file format has changed from legacy `<parameter> = <value>` to the Device Tree Source (DTS) format for following reasons:

- DTS format supports recursive inclusion and the overriding of properties though inclusion.
- DTC can covert DTS/DTSI to DTB, which is a essentially tree-like format and is easier to parse compared to the current config file format.

1.1 BR-BCT

BR-BCT is loaded by BootROM from storage in coldboot and by MB1 in recovery mode. The primary consumer of this BCT is BootROM, but some fields are also consumed by MB1 and CPU-BL.

1.2 MB1-BCT

MB1 BCT is loaded by MB1 from storage (in coldboot mode) or over USB (in RCM mode). It is primarily consumed by MB1 and is constructed out of the following multiple configuration files:

- "Pinmux and GPIO Configuration" on page 3
- "Common Prod Configuration" on page 7
- "Controller Prod Configuration" on page 9
- "Pad Voltage Binding" on page 12
- "PMIC Configuration" on page 14
- "Storage Device Configuration" on page 27
- "UPHY Lane Configuration" on page 33

- "OEM FW Ratchet Configuration" on page 37
- "BootROM Reset PMIC Configuration" on page 37
- "Miscellaneous Configuration" on page 43

1.3 Mem-BCT

MemBCT is like MB1-BCT in terms of loading and usage. However, it primarily contains SDRAM parameters that are used to initialize MC and EMC. Refer to SDRAM Configuration for more information.

1.4 MB2-BCT

MB2 BCT is loaded by MB1 from storage (in coldboot mode) or over USB (in RCM mode). It is primarily consumed by MB2

It is constructed out of multiple configuration files as listed below:

- GPIO Interrupt Mapping Configuration
- Security Configuration
- MB2 BCT Misc Configuration

Chapter 2. Pinmux and GPIO Configuration

The pinmux configuration file provides pinmux and GPIO configuration. Pinmux and GPIO configuration is generated using pinmux spreadsheet.

The stark contrast in the New DTS format with respect to old legacy format is becasue of the pinmux sheet output.

The pinmux DTS file are kept in the `hardware/nvidia/platform/t23x/<platform>/bct/` directory.

NEW DTS format example of pinmux configuration file:

```
/*This dtsi file was generated by

e3360_1099_slt_a01.xlsm Revision: 126 */ #include

<dt-bindings/pinctrl/pinctrl-tegra.h>

/ {
   pinmux@2430000 {
      pinctrl-names =
      "default", "drive",
      "unused"; pinctrl-0 =
      <&pinmux_default>;
      pinctrl-1 = <&drive_default>;
      pinctrl-2 = <&pinmux_unused_lowpower>;

      pinmux_default: common {
         /* SFIO Pin
         Configuration */
         dap1_sclk_ps0 {
            nvidia,pins =
            "dap1_sclk_ps0";
            nvidia,function
            = "i2s1";
            nvidia,pull =
            <TEGRA_PIN_PULL_
            NONE>;
            nvidia,tristate =
            <TEGRA_PIN_DISABLE>;
            nvidia,enable-input =
            <TEGRA_PIN_DISABLE>;
            nvidia,lpdr =
            <TEGRA_PIN_DISABLE>;
         };

         dap1_dout_ps1 {
            nvidia,pins =
            "dap1_dout_ps1";
            nvidia,function
            = "i2s1";
            nvidia,pull =
            <TEGRA_PIN_PULL_
            NONE>;
            nvidia,tristate =
            <TEGRA_PIN_DISABLE>;
            nvidia,enable-input =
            <TEGRA_PIN_DISABLE>;
            nvidia,lpdr =
            <TEGRA_PIN_DISABLE>;
         };

         dap1_din_ps2 {
            nvidia,pins =
            "dap1_din_ps2";
            nvidia,function
            = "i2s1";
            nvidia,pull =
            <TEGRA_PIN_PULL_
            DOWN>;
            nvidia,tristate =
            <TEGRA_PIN_ENABLE>;
            nvidia,enable-input
            =
            <TEGRA_PIN_ENABLE>;
            nvidia,lpdr =
            <TEGRA_PIN_DISABLE>;
         };
         dap1_fs_ps3 {
            nvidia,pins =
            "dap1_fs_ps3";
            nvidia,function
            = "i2s1";
            nvidia,pull =
            <TEGRA_PIN_PULL_
            NONE>;
            nvidia,tristate =
            <TEGRA_PIN_DISABLE>;
            nvidia,enable-input =
            <TEGRA_PIN_DISABLE>;
            nvidia,lpdr =
            <TEGRA_PIN_DISABLE>;
         };
```

```
aud_mclk_ps4 {
   nvidia,pins =
   "aud_mclk_ps4";
   nvidia,function =
   "aud";
   nvidia,pull =
   <TEGRA_PIN_PULL_NONE>
   ; nvidia,tristate =
   <TEGRA_PIN_DISABLE>;
   nvidia,enable-input =
   <TEGRA_PIN_DISABLE>;
   nvidia,lpdr =
   <TEGRA_PIN_DISABLE>;
};

soc_gpio31_ps6 {
   nvidia,pins =
   "soc_gpio31_ps6";
   nvidia,function =
   "sdmmc1";
   nvidia,pull =
   <TEGRA_PIN_PULL_U
   P>;
   nvidia,tristate =
   <TEGRA_PIN_ENABLE
   >;
   nvidia,enable-input =
   <TEGRA_PIN_ENABLE>;
   nvidia,lpdr =
   <TEGRA_PIN_DISABLE>;
};

soc_gpio32_ps7 {
   nvidia,pins =
   "soc_gpio32_ps7";
   nvidia,function =
   "spdif";
   nvidia,pull =
   <TEGRA_PIN_PULL_D
   OWN>;
   nvidia,tristate =
   <TEGRA_PIN_ENABLE
   >;
   nvidia,enable-input =
   <TEGRA_PIN_ENABLE>;
   nvidia,lpdr =
   <TEGRA_PIN_DISABLE>;
};

soc_gpio33_pt0 {
   nvidia,pins =
   "soc_gpio33_pt0"
   ;
```

```
                nvidia,function
                = "spdif";
                nvidia,pull =
                <TEGRA_PIN_PULL_
                NONE>;
                nvidia,tristate =
                <TEGRA_PIN_DISABLE>;
                nvidia,enable-input =
                <TEGRA_PIN_DISABLE>;
                nvidia,lpdr =
                <TEGRA_PIN_DISABLE>;
            };
        };

        drive_default: drive {
        };
    };
};
```

OLD CFG format

```
//////// Pinmux for used pins ////////
pinmux.0x02434060 = <value1>; //
gen1_i2c_scl_pc5.PADCTL_CONN_GEN1_I2C_SCL_0 pinmux.0x02434064 =
<value2>; // gen1_i2c_scl_pc5.PADCTL_CONN_CFG2TMC_GEN1_I2C_SCL_0
pinmux.0x02434068 = <value1>; //
gen1_i2c_sda_pc6.PADCTL_CONN_GEN1_I2C_SDA_0 pinmux.0x0243406C =
<value2>; // gen1_i2c_sda_pc6.PADCTL_CONN_CFG2TMC_GEN1_I2C_DA_0

//////// Pinmux for unused pins for low-power
configuration //////// pinmux.0x02434040 = <value1>;
// gpio_wan4_ph0.PADCTL_CONN_GPIO_WAN4_0
pinmux.0x02434044 = <value2>; //
gpio_wan4_ph0.PADCTL_CONN_CFG2TMC_GPIO_WAN4_0
pinmux.0x02434048 = <value1>; //
gpio_wan3_ph1.PADCTL_CONN_GPIO_WAN3_0
pinmux.0x0243404C = <value2>; //
gpio_wan3_ph1.PADCTL_CONN_CFG2TMC_GPIO_WAN3_0
```

Chapter 3. Common Prod Configuration

The prod configurations are the system characterized values of interface and controller settings, which are required for the given interface to work reliably for a platform. The prod configuration are set separately at controller and pinmux/pad levels. This file contains the common pinmux/pad level prod settings.

Required properties:

- `addr-value-data`: List of `<Absolute PADCTL register address, mask, data>`

For each such entry in the prod configuration file, MB1 reads the data from the specified address, modifies the data based on mask and value, and writes the data back to the address.

```
val = read(address)

val = (val & ~mask) | (value & mask); write(val, address);
```

The common prod DTS file are kept in the `hardware/nvidia/platform/t23x/<platform>/bct/` directory.

NEW DTS format example of the prod config file:

```
/dts-v1/;


/ {
    prod {
         addr-mask-data = <0x0c302030
                 0x0000100 0x00000000>,
  <0x0c302040 0x0000100 0x00000000>,
  <0x0244100c 0xff1ff000 0x0a00a000>,
  <0x02441004 0xff1ff000 0x0a00a000>;


    };
};
```

OLD CFG format

```
prod.major = 1;
prod.minor = 0;

prod.0x0c302030.0x0000100 =
0x00000000;
prod.0x0c302040.0x0000100 =
0x00000000;
prod.0x0244100c.0xff1ff000 =
0x0a00a000;
prod.0x02441004.0xff1ff000 =
0x0a00a000;
```

Chapter 4. Controller Prod Configuration

The prod configurations are the system characterized values of interface and controller settings, which are required for the given interface to work reliably for a platform. The prod configuration are set separately at controller and pinmux/pad levels. This file contains the controller level prod settings.

The DTS configuration file is of the following form:

```
/ {
    deviceprod {
        <controller-name>-
        <Instance> = <&Label>;
        #prod-cells = <N>;
        <Label>: <controller-name>@<base-address> {
            <mode> {
                prod = <<address offset> <mask> <value>>;
            };
        };
    };
};
```

where:

- `Instance` is controller instance id
- `Label` is the label assigned to the node which can be referenced for mapping instance to a node
- `<controller-name>` is predefined module name (sdmmc, qspi, se, i2c)
- `<base-address>` is base address of the controller
- `<mode>` is controller mode for which the prod setting needs to be applied (e.g. default, hs400, etc)
- `<address offset>` is the register address offset from base address of the controller/instance
- `<mask>` is the mask value (4 bytes, unsigned)

- `<value>` is the data value (4 bytes, unsigned)
- `N` specifies how many entries are there on prod setting tuples per configurations.

  if #prod-cells == 3, there are three entries per prod configuration (`address _offset`, `mask`, and `value`).

The legacy config format used device instance `<controller-name>.<instance-index>` instead of using the base address of the device. The legacy format keeps one byte to store the instance, but new DTS format has a base address, which requires four bytes in the BCT. As a result, some of the fields in the BCT structure have to be shifted accordingly.

For each entry in the prod configuration file, MB1 reads data from the specified address, modifies the data based on the mask and value, and the data back to the address.

```
    val = read(address)
    val = (val & ~mask) | (value
    & mask); write(val,
    address);
```

The common prod configuration file is in the `hardware/nvidia/platform/t23x/<platform>/bct/` directory.

NEW DTS example of prod configuration file:

```
 /dts-v1/;

 / {
    deviceprod {
        qspi-0 = <&qspi0>; qspi-1 = <&qspi1>; sdmmc-3 = <&sdmmc3>;

        #prod-

        cells =

        <0x3>;

        qspi0:

        qspi@327

        0000 {
            default  {
 prod = <0x00000004 0x7C00
                      0x0>,
  <0x00000004 0xFF 0x10>;
            };
        };
        qspi1:
            qspi
            @330
            0000
            {
            defa
            ult
            {
```

```
 prod = <0x00000004 0x7C00
                     0x0>,
  <0x00000004 0xFF 0x10>;
           };
       };
       sdmmc3:
           sdmm
           c@34
           6000
           0 {
           defa
           ult
           {
              prod = <0x000001e4 0x00003FFF 0x0>;
           };
           hs400 {
     prod = <0x00000100 0x1FFF0000
                       0x14080000>,
<0x0000010c 0x00003F00 0x00000028>;
           };
           ddr52 {
              prod = <0x00000100 0x1FFF0000 0x14080000>;
           };
       };
    };
};
```

OLD CFG format

```
//Qspi0
deviceprod.qspi.0.default.0x03270004.0
x7C00 = 0x0  //TX Trimmer
deviceprod.qspi.0.default.0x03270004.0
xFF                                  =
0x10 //RX Trimmer

//Qspi1
deviceprod.qspi.1.default.0x03300004.0
x7C00 = 0x0  //TX Trimmer
deviceprod.qspi.1.default.0x03300004.0
xFF                                  =
0x10 //RX Trimmer

//SDMMC
deviceprod.sdmmc.3.default.0x034601e4.0x00003FFF = 0x0 // auto cal pd and
pu offsets deviceprod.sdmmc.3.hs400.0x03460100.0x1FFF0000 = 0x14080000 //
tap and trim values deviceprod.sdmmc.3.hs400.0x0346010c.0x00003F00 =
0x00000028 // DQS trim val deviceprod.sdmmc.3.ddr52.0x03460100.0x1FFF0000 =
0x14080000 // tap and trim values
```

Chapter 5. Pad Voltage Binding

Tegra pins and pads are designed to support multiple voltage levels at an interface. They can operate at 1.2 volts (V), 1.8 V or 3.3 V. Based on the interface and power tree of a given platform, the software must write to the correct voltage of these pads to enable interface. If pad voltage is higher than the I/O power rail, the pin does not work on that level. If pad voltage is lower than the I/O power rail, it can damage the SoC pads. Consequently, configuring the correct pad voltage is required , and this configuration is based on the power tree.

The Pad voltage DTSI is generated using pinmux spread sheet.

The prod configuration files are kept in the `hardware/nvidia/platform/t23x/<platform>/bct/` directory. The contrast in the New DTS format is because of the pinmux sheet output.

NEW DTS format example of pad-voltage configuration file:

```
/*This dtsi file was generated by e3360_1099_slt_a01.xlsm Revision: 126 */


/ {
    pmc@c360000 {
          io-pad-defaults {
               sdmmc1_hv {
                 nvidia,io-pad-init-voltage = <IO_PAD_VOLTAGE_1_8V>;
             };

             sdmmc3_hv {
                 nvidia,io-pad-init-voltage = <IO_PAD_VOLTAGE_1_8V>;
             };

             eqos {
                 nvidia,io-pad-init-voltage = <IO_PAD_VOLTAGE_1_8V>;
             };

             qspi {
                 nvidia,io-pad-init-voltage = <IO_PAD_VOLTAGE_1_8V>;
             };

             debug {
                 nvidia,io-pad-init-voltage = <IO_PAD_VOLTAGE_1_8V>;
```

```
            };

            ao_hv {
               nvidia,io-pad-init-voltage = <IO_PAD_VOLTAGE_3_3V>;
            };

            audio_hv {
               nvidia,io-pad-init-voltage = <IO_PAD_VOLTAGE_3_3V>;
            };

            ufs {
               nvidia,io-pad-init-voltage = <IO_PAD_VOLTAGE_1_2V>;
            };


         };
      };
   };
```

OLD CFG format

```
pad-voltage.major = 1;
pad-voltage.minor = 0;
pad-voltage.0x0c36003c = 0x0000003e;
// PMC_IMPL_E_18V_PWR_0 pad-
voltage.0x0c360040 = 0x00000079; //
PMC_IMPL_E_33V_PWR_0
```

Chapter 6. PMIC Configuration

During system boot, MB1 enables system power rails for CPU, CORE, DRAM and completes some system PMIC configurations. Here is a list of the typical configurations:

- Enabling rails
- Setting rail voltages
- FPS configurations

Enabling and setting of voltages of rails might require following platform-specific configurations:

- I2C commands to devices
- PWM commands to devices
- MMIO accesses to Tegra registers, either read-modify-write or write-only
- Delay after the commands

The entries in PMIC configuration files are either common or rail-specific.

6.1 Common Configuration

The common configuration parameters are applicable to all rails. Each PMIC common configuration is of the following form:

```
/ {
    pmic {
      <parameter> = <value>;
    };
};
```

where:

- `<parameter>` is one of the following:

<parameter>	Description
rail-count	Number of rails in the configuration file.
command-retries-count	The number of allowed command attempts.
wait-before-start-bus-clear-us	Wait timeout, in microseconds before issuing the bus clear command. The wait time is calculated as 1 <<n microseconds where n is provided by this parameter.

6.2 Rail-Specific Configuration

The rail-specific configuration are divided into blocks.

- Each rail can have one or more blocks.
- Each block can have only one type of commands (I2C, PWM, or MMIO).

Each PMIC rail-specific configuration is of the following form:

```
/ {
    pmic {
      <rail-name> {
        block@<index> {
          <parameter> = <value>;
        };
      };
    };
};
```

where:

- `<rail-name>` identifies the rail and is one of the following:

Name	Description
system	System PMIC configuration
cpu / cpu0	CPU rail configuration
cpu1	CPU rail configuration
core	Core/SOC rail configuration
memio	DRAM related rail configuration
thermal	External thermal sensor configuration

Name	Description
platform	Platform's other I2C configuration

- `block-<index>` The rail specific commands are divided into blocks.

  Each rail can have multiple blocks. Each block of given rails indexed starting from 0.

- `<parameter>` is one of the following:

<parameter>	Description
<type>	Type of commands in the block. Valid properties are i2c-controller, pwm or mmio.

commands {
<group-name> {
command@N {
reg-addr = <reg-address>; mask = <reg-mask>;
value = <reg-value>;
};
};
};
	<group-name> node is the logical command group name and it is OPT⇜,
	IONAL. N is the sequential number for the command starting from 0. MMIO or I2C commands (based on

Type of command block	Description
mmio	MMIO command (valid only if block has mmio property), where, <address> is absolute address of MMIO register and <mask> is the 32bit mask that is applied to the value read from the MMIO address to facilitate read-modify-write operation. <value> value written into the register
i2c-controller	I2c command, where <address> is the I2c slave register address, and <mask> is I2C slave mask that

<parameter>	Description
<i2c-parameter>	I2C parameter (valid only if block has i2c- controller property), which is one of the following:

<i2c-parameter>	Description
block-delay	Delay (in microseconds), after each command in the block. <index> is the block index (starting from 0).
i2c-controller-id	I2C controller instance
slave-addr	7-bit I2C slave address
reg-data-size	Register size in bits. Valid values are 0(1-byte), 8(1-byte) and 16(2-byte)
reg-addr-size	Register address size in bits. Valid values are 0(1-byte)

pwm	PWM parameter (valid only if block has pwm property), which is one of the following:

<pwm-parameter>	Description
controller-id	PWM controller instance (0-7)
source-frq-hz	PWM clock source frequency (in Hz)
period-ns	PWM time period (in nanoseconds)
min-microvolts	Vout from PWM regulator if duty cycle is 0
max-microvolts	Vout from PWM regulator if duty cycle is 100
init-microvolts	Vout from PWM regulator after initialization
enable	0 (just configure PWM, do not enable); 1 (enable PWM after configuring)

6.3 Relative Order of Execution of the PMIC Configuration by MB1

Apart from the order and point in boot in which the different rail configurations are executed, there is no difference in how MB1 treats each of these configurations. Depending on the platform, some of these configurations might be optional. Therefore, MB1 treats all configurations as optional and prints only a warning when a configuration is not provided in the MB1-BCT.

These sequences are executed in the following order by MB1:

1. External thermal sensor configuration.
2. Generic PMIC configuration.
3. SOC rail configuration.
4. DRAM related rail configuration.
5. DRAM initialization.
6. CPU rail configuration.
7. Loading CPU related microcode and enabling CPUs.
8. Platform's other I2C configuration.

The PMIC configuration files are kept in the `hardware/nvidia/platform/t23x/<platform>/bct/` directory.

NEW DTS format example of PMIC configuration file

```
/dts-v1/;
/ {
   pmic {
          system {
                 block@0 {
                    controller-id = <4>;
                    slave-addr = <0x78>; // 7BIt:0x3c reg-data-size = <8>;
                    reg-addr-size = <8>;
                    block-delay = <10>;
                    i2c-update-verify = <1>; //update and verify
                    commands {
                               cpu-rail-cmds {
                                       command@0 {
                                                  reg-addr = <0x50>;
                                                  mask = <0xC0>;
                                                  value = <0x00>;
                                      };
                                      command@1 {
                                                 reg-addr = <0x51>;
                                                 mask = <0xC0>;
                                                  value = <0x00>;
                                     };
                                     command@2 {
                                                reg-addr = <0x4A>;
                                                mask = <0xC0>;
                                                 value = <0x00>;
                                     };
                                     command@3 {
                                                reg-addr = <0x4B>;
                                                mask = <0xC0>;
                                                value = <0x00>;
                                    };
                                    command@4 {
                                               reg-addr = <0x4C>;
```

```
                                                    mask = <0xC0>;
                                                    value = <0x00>;
                                           };
};
gpio07-cmds {
         command@0 {
                    reg-addr = <0xAA>;
                    mask = <0xBB>;
                     value = <0xCC>;
       };
       command@1 {
                  reg-addr = <0xDD>;
                 mask = <0xEE>;
                 value = < 0xFF>;
      };
};
misc-cmds {
          command@0 {
                      reg-addr = <0x53>;
                     mask = <0x38>;
                     value = <0x00>;
              };
              command@1 {
                         reg-addr = <0x55>;
                         mask = <0x38>;
                         value = < 0x10>;
              };
              command@2 {
                         reg-addr = <0x41>;
                         mask = <0x1C>;
                         value = <0x1C>;
               };
             };
         };
      };
};
core {
      block@0 {
              pwm;
              controller-id = <6>;
              source-frq-hz = <204000000>;
              period-ns = <1255>;
              min-microvolts = <400000>;
              max-microvolts = <1200000>;
              init-microvolts = <850000>; enable;
};
block@1 {
        mmio;
        block-delay = <3000>; commands {
         command@0 {
                   reg-addr = <0x02434080>;
                   mask = <0x10>;
                    value = <0x0>;
                     };
               };
        };
};
```

```
cpu@0 {
          block@0 {
                  pwm;
                  controller-id = <5>;
                  source-frq-hz = <204000000>;
                  period-ns = <1255>;
                  min-microvolts = <400000>;
                  max-microvolts = <1200000>;
                   init-microvolts = <800000>; enable;
};
block@1 {
         mmio;
         block-delay = <3>; commands {
                 commands@0 {
                            reg-addr = <0x02214e00>;
                            mask = <0x3>;
                            value = <0x00000003>;
                 };
                command@1 {
                           reg-addr = <0x02214e0c>;
                          mask = <0x1>;
                          value = <0x00000000>;
               };
               command@2 {
                        reg-addr = <0x02214e10>;
                        mask = <0x1>;
                        value = <0x00000001>;
                  };
                  command@3 {
                             reg-addr = <0x02446008>;
                             mask = <0x400>;
                             value = <0x00000000>;
                  };
                  command@4 {
                             reg-addr = <0x02446008>;
                             mask = <0x10>;
                              value = <0x00000000>;
                    };
          };
};
block@2 {
mmio;
commands {
           command@0 {
                      reg-addr = <0x02434098>;
                      mask = <0x10>;
                      value = <0x00>;
            };
       };
    };
};
platform {
       block@0 {
                i2c-controller; controller-id = <1>;
                slave-addr = <0x40>;
                reg-data-size = <8>;
                reg-addr-size = <8>;
                block-delay = <10>;
                i2c-update-verify = <1>;
commands {
           command@0 {
                      reg-addr = <0x03>;
                      mask = <0x30>;
                       value = <0x00>;
            };
            command@1 {
                       reg-addr = <0x01>;
                      mask = <0x30>;
                       value = <0x20>;
             };
         };
       };
     };
  };
};


/dts-v1/;

/ {
     pmic {
         system {
            block@0 {
               controller-id = <4>;
               slave-addr = <0x78>; // 7BIt:0x3c reg-data-size = <8>;
               reg-addr-size = <8>;
               block-delay = <10>;
               i2c-update-verify = <1>; //update and verify commands {
                    cpu-rail-cmds {
                        command@0 {
                             reg-addr = <0x50>;
                              mask = <0xC0>;
                              value = <0x00>;
                         };
                         command@1 {
                             reg-addr = <0x51>;
                             mask = <0xC0>;
                             value = <0x00>;
                          };
                        command@2 {
                             reg-addr = <0x4A>;
                             mask = <0xC0>;
                             value = <0x00>;
                        };
                        command@3 {
                             reg-addr = <0x4B>;
                             mask = <0xC0>;
                             value = <0x00>;
                         };
                         command@4 {
                              reg-addr = <0x4C>;
                              mask = <0xC0>;
                              value = <0x00>;
                         };
                         };
                         gpio07-cmds {
                             command@0 {
                                 reg-addr = <0xAA>;
                                 mask = <0xBB>;
                                 value = <0xCC>;
                          };
                          command@1 {
                               reg-addr = <0xDD>;
                               mask = <0xEE>;
                               value = < 0xFF>;
                                      };
                         };
                         misc-cmds {
                                command@0 {
                                                   reg-addr = <0x53>;
                                                   mask = <0x38>;
                                                    value = <0x00>;
                                          };
                                          command@1 {
                                                     reg-addr = <0x55>;
                                                     mask = <0x38>;
                                                      value = < 0x10>;
                                          };
                                          command@2 {
                                                     reg-addr = <0x41>;
                                                     mask = <0x1C>;
                                                      value = <0x1C>;
                                           };
                                   };

                             };
                      };
};
core {
       block@0 {
               pwm;
               controller-id = <6>;
               source-frq-hz = <204000000>;
               period-ns = <1255>;
               min-microvolts = <400000>;
               max-microvolts = <1200000>;
               init-microvolts = <850000>; enable;
 };
Block@1 {
         mmio;
          block-delay = <3000>; commands {
                   command@0 {
                              reg-addr = <0x02434080>;
                              mask = <0x10>;
                               value = <0x0>;
                          };
                  };
           };
};
```

```
cpu@0 {
          block@0 {
      pwm;
      controller-id = <5>;
      source-frq-hz = <204000000>;
      period-ns = <1255>;
      min-microvolts = <400000>;
      max-microvolts = <1200000>;
      init-microvolts = <800000>; enable;
     };
           block@1 {
                   mmio;
                   block-delay = <3>; commands {
                       commands@0 {
                           reg-addr = <0x02214e00>;
                           mask = <0x3>;
                           value = <0x00000003>;
                       };
                       command@1 {
                           reg-addr = <0x02214e0c>;
                           mask = <0x1>;
                           value = <0x00000000>;
                       };
                       command@2 {
                           reg-addr = <0x02214e10>;
                           mask = <0x1>;
                           value = <0x00000001>;
                       };
                       command@3 {
                           reg-addr = <0x02446008>;
                           mask = <0x400>;
                           value = <0x00000000>;
                       };
                       command@4 {
                           reg-addr = <0x02446008>;
                           mask = <0x10>;
                           value = <0x00000000>;
                              };
                          };
};
block@2 {
     mmio;
      commands {
                 command@0 {
                            reg-addr = <0x02434098>;
                            mask = <0x10>;
                            value = <0x00>;
                   };
              };
        };
};
platform {
       block@0 {
           i2c-controller;
           controller-id = <1>;
           slave-addr = <0x40>;
           reg-data-size = <8>;
```

```
            reg-addr-size = <8>;
            block-delay = <10>;
            i2c-update-verify = <1>;
            commands {
                       command@0 {
                                  reg-addr = <0x03>;
                                  mask = <0x30>;
                                   value = <0x00>;
                         };
                          command@1 {
                                     reg-addr = <0x01>;
                                     mask = <0x30>;
                                      value = <0x20>;
};
```

OLD CFG format

```
//////////////////////////////////////////////// System Configurations
////////
// PMIC FPS to turn SD4 (VDD_DDR_1V1) on in time slot 0
// PMIC FPS to set GPIO2 (EN_DDR_VDDQ)  high in time slot 1
// Set SLPEN = 1 and CLRSE on
POR reset
pmic.system.block[0].type =
1; //I2C
pmic.system.block[0].controll
er-id = 4;
pmic.system.block[0].slave-
add = 0x78; // 7BIt:0x3c
pmic.system.block[0].reg-
data-size = 8;
pmic.system.block[0].reg-add-size = 8;
pmic.system.block[0].block-delay = 10;
pmic.system.block[0].i2c-update-verify = 1; //update and verify
pmic.system.block[0].commands[0].0x53.0x38 = 0x00; // SD4 FPS UP
slot 0 pmic.system.block[0].commands[1].0x55.0x38 = 0x10; //
GPIO2 FPS UP slot 2 pmic.system.block[0].commands[2].0x41.0x1C =
0x1C; // SLPEN=1, CLRSE = 11

// PMIC FPS programming to reassign SD1, SD2, LDO4 and LDO5 to
// FPS0 to leave those rails
on in SC7
pmic.system.block[1].type =
1; //I2C
pmic.system.block[1].controll
er-id = 4;
pmic.system.block[1].slave-
add = 0x78; // 7BIt:0x3c
pmic.system.block[1].reg-
data-size = 8;
pmic.system.block[1].reg-add-size = 8;
pmic.system.block[1].block-delay = 10;
pmic.system.block[1].i2c-update-verify = 1; //update and verify
pmic.system.block[1].commands[0].0x50.0xC0 = 0x00; // SD1 FPS to
```

```
FPS0 pmic.system.block[1].commands[1].0x51.0xC0 = 0x00; // SD2
FPS to FPS0 pmic.system.block[1].commands[2].0x4A.0xC0 = 0x00;
// LDO4 FPS to FPS0 pmic.system.block[1].commands[3].0x4B.0xC0 =
0x00; // LDO5 FPS to FPS0
pmic.system.block[1].commands[4].0x4C.0xC0 = 0x00; // LDO6 FPS to
FPS0

// VDDIO_DDR to 1.1V, SD4 to 1.1V
pmic.system.block[2].type =
1; //I2C
pmic.system.block[2].controll
er-id = 4;
pmic.system.block[2].slave-
add = 0x78; // 7BIt:0x3c
pmic.system.block[2].reg-
data-size = 8;
pmic.system.block[2].reg-add-size = 8;
pmic.system.block[2].block-delay = 10;
pmic.system.block[2].i2c-update-verify = 1; //update and verify
pmic.system.block[2].commands[0].0x1A.0xFF =  0x28; // SD4 to
1.1V

///////////////////////////////////////////////// //CORE(SOC) RAIL
Configurations /////////////////////////////
// 1. Set 850mV
voltage.
pmic.core.block[0].typ
e = 2; // PWM Type
pmic.core.block[0].controller-id
= 6; //SOC_GPIO10: PWM7
pmic.core.block[0].source-frq-hz
= 204000000; //204MHz
pmic.core.block[0].period-ns =
1255; // 800KHz.
pmic.core.block[0].min-microvolts
= 400000;
pmic.core.block[0].max-microvolts = 1200000;
pmic.core.block[0].init-microvolts = 850000;
pmic.core.block[0].enable = 1;

// 2. Make soc_gpio10
pin  in  non-tristate
pmic.core.block[1].ty
pe = 0; // MMIO TYPE
pmic.core.block[1].bl
ock-delay = 3000;
pmic.core.block[1].commands[0].0x02434080.0x10 = 0x0; // soc_gpio10: tristate
(b4) = 0

///////////////////////////////////////////////// //CPU0 RAIL configurations
//////////////////////////////
// 1. Set 800mV
voltage.
pmic.cpu0.block[0].typ
```

```
e = 2; // PWM Type
pmic.cpu0.block[0].controller-id
= 5; //soc_gpio13; PWM6
pmic.cpu0.block[0].source-frq-hz
= 204000000; //204MHz
pmic.cpu0.block[0].period-ns =
1255; // 800KHz.
pmic.cpu0.block[0].min-microvolts
= 400000;
pmic.cpu0.block[0].max-microvolts = 1200000;
pmic.cpu0.block[0].init-microvolts = 800000;
pmic.cpu0.block[0].enable = 1;

// 2. CPU PWR_REQ
cpu_pwr_req_0_pb0 to be
1
pmic.cpu0.block[1].type
= 0; // MMIO TYPE
pmic.cpu0.block[1].bloc
k-delay = 3;
pmic.cpu0.block[1].commands[0].0x02214e00.0x3 = 0x00000003; // CONFIG B0
pmic.cpu0.block[1].commands[1].0x02214e0c.0x1 = 0x00000000; // CONTROL B0
pmic.cpu0.block[1].commands[2].0x02214e10.0x1 = 0x00000001; // OUTPUT B0
pmic.cpu0.block[1].commands[3].0x02446008.0x400 = 0x00000000; //
cpu_pwr_req_0_pb0 to GPIO mode
pmic.cpu0.block[1].commands[4].0x02446008.0x10 = 0x00000000; //
cpu_pwr_req_0_pb0 tristate(b4)=0

// 3. Set soc_gpio13 to
untristate
pmic.cpu0.block[2].type = 0; //
MMIO Type
pmic.cpu0.block[2].commands[4].0x02434098.0x10 = 0x00; // soc_gpio13 to be
untristate

////////////////////////////////////////////////// Platform Configurations
//////////////////////////////
// Configure pin4/pin5 as output of gpio expander(0x40)
// Configure pin4 low and pin5 high
of gpio expander(0x40)
pmic.platform.block[0].type = 1;
//I2C
pmic.platform.block[0].controller-
id = 1; //gen2
pmic.platform.block[0].slave-add =
0x40; // 7BIt:0x20
pmic.platform.block[0].reg-data-
size = 8;
pmic.platform.block[0].reg-add-size = 8;
pmic.platform.block[0].block-delay = 10;
pmic.platform.block[0].i2c-update-verify
= 1; //update and verify
pmic.platform.block[0].commands[0].0x03.0x30 = 0x00; //Configure pin4/pin5
as output pmic.platform.block[0].commands[1].0x01.0x30 = 0x20; //Configure
pin4 low and pin5 high
```

Chapter 7. Storage Device Configuration

The Storage Device configuration file contains the platform-specific settings for storage devices in the MB1/MB2 stages.

The DTS Configuration file is of the following form:

```
/ {
  device {
    <storage_device>@instance-#  {
        <parameter> = <value>;
    };
  };
};
```

where:

- `<storage-device>` is the storage device controller (`qspiflash/ufs/sdmmc/sata`).
- `<instance-#>` is the instance of the storage controlle
- `<parameter>` is controller-specific parameter as shown below.

7.1 QSPI Flash Parameters

Parameter	Description
clock-source-id	QSPI controller Clock Source 1: CLK_M 3: PLLP_OUT0 4: PLLM_OUT0 5: PLLC_OUT0 6: PLLC4_MUXED
clock-source-frequency	Frequency of clock source (in Hz)
interface-frequency	QSPI controller frequency (in Hz)
enable-ddr-mode	0: QSPI SDR mode 1: QSPI DDR mode

Parameter	Description
maximum-bus-width	Maximum QSPI bus width 0: QSPI x1 lane 2: QSPI x4 lane
fifo-access-mode	0: PIO mode 1: DMA mode
ready-dummy-cycle	No. of dummy cycles as per QSPI flash
trimmer1-val	TX trimmer value
trimmer2-val	RX trimmer value

7.2 SDMMC Parameters

Parameter	Description
clock-source-id	SDMMC controller Clock Source 0: PLLP_OUT0 1: PLLC4_OUT2_LJ 2: PLLC4_OUT0_LJ 3: PLLC4_OUT2 4: PLLC4_OUT1 5: PLLC4_OUT1_LJ 6: CLK_M 7: PLLC4_VCO
clock-source-frequency	Frequency of clock source (in Hz)
best-mode	Highest supported mode of operation 0: SDR26 1: DDR52 2: HS200 3: HS400
pd-offset	Pull-down offset
pu-offset	Pull-up offset
enable-strobe-hs400	Enable HS400 strobe
dqs-trim-hs400	HS400 DQS trim value

7.3 UFS Parameters

Parameter	Description
max-hs-mode	Highest HS mode supproted by UFS device 1: HS GEAR1 2: HS GEAR2 3: HS GEAR3
max-pwm-mode	Highest PWM mode supproted by UFS device 1: PWM GEAR1 2: PWM GEAR2 3: PWM GEAR3 4: PWM GEAR4
max-active-lanes	Maximum number of UFS lanes (1-2)
page-align-size	Alignment of pages used for UFS data structures (in bytes)
enable-hs-mode	Whether to enable UFS HS modes 0: disable 1: enable
enable-fast-auto-mode	Enable fast auto mode 0: disable 1: enable
enable-hs-rate-a	Enable HS rate A 0: disable 1: enable

Parameter	Description
enable-hs-rate-b	Enable HS rate B 0: disable 1: enable
init-state	Initial state of UFS device at MB1 entry 0: UFS not initialized by BootROM 1: UFS is initialized by BootROM (MB1 can skip certain steps)

7.4 SATA Parameters

Parameter	Description
transfer-speed	0: GEN1 1: GEN2

The storage device configuration file are kept in the `hardware/nvidia/platform/t23x/<platform>/bct/` directory.

NEW DTS format example of storage device configuration file:

```
/dts-v1/;

#include <defines.h>

/ {
    device {
        qspiflash@0 {
            clock-source-id
            = <PLLC_MUXED>;
            clock-source-
            frequency =
            <13000000>;
            interface-
            frequency =
            <13000000>;
            enable-ddr-
            mode;
            maximum-bus-
            width =
            <QSPI_4_LANE>;
            fifo-access-
            mode =
            <DMA_MODE>;
            read-dummy-
            cycle = <8>;
            trimmer1-val = <0>;
            trimmer2-val = <0>;
        };
        sdmmc@3 {
            clock-source-id
            = <PLLC4_OUT2>;
            clock-source-
            frequency =
            <52000000>;
            best-mode =
            <HS400>;
            pd-offset = <0>;
            pu-offset = <0>;
            //enable-strobe-hs400; This property is not
            there means it is disabled dqs-trim-hs400 =
            <0>;
        };
        ufs@0 {
            max-hs-
            mode =
            <HS_GEAR_3
            >; max-
            pwm-mode =
            <PWM_GEAR_
            4>; max-
            active-
            lanes =
            <2>;
            page-
            align-
            size =
            <4096>;
            enable-
            hs-
            mode;
            //enabl
            e-fast-
            auto-
            mode;
            enable-
            hs-
            rate-b;
            //enable-hs-rate-a = <0>;
            init-state = <0>;
        };
    };
};
```

OLD CFG format

```
// QSPI flash 0
device.qspiflash.0
.clock-source-id =
6;
device.qspiflash.0.clock-source-frequency = 13000000;
device.qspiflash.0.interface-frequency = 13000000;
device.qspiflash.0.enable-ddr-mode = 0;
device.qspiflash.0.maximum-bus-width = 2;
device.qspiflash.0.fifo-access-mode = 1;
device.qspiflash.0.read-dummy-cycle = 8;
device.qspiflash.0.trimmer1-val = 0;
device.qspiflash.0.trimmer2-val = 0;

// Sdmmc 3
device.sdmmc.3.clock-
source-id = 3; //PLLP_OUT0
device.sdmmc.3.clock-
source-frequency =
52000000;
device.sdmmc.3.best-mode =
3; //1=DDR52, 3=HS400
device.sdmmc.3.pd-offset =
0;
device.sdmmc.3.pu-offset = 0;
device.sdmmc.3.enable-strobe-hs400 = 0;
device.sdmmc.3.dqs-trim-hs400 = 0;

// Ufs 0
device.ufs.0.max-hs-mode = 3;
device.ufs.0.max-pwm-mode = 4;
device.ufs.0.max-active-lanes = 2;
device.ufs.0.page-align-size = 4096;
device.ufs.0.enable-hs-mode = 1;
device.ufs.0.enable-fast-auto-mode = 0;
device.ufs.0.enable-hs-rate-b = 1;
device.ufs.0.enable-hs-rate-a = 0;
device.ufs.0.init-state = 0;
```

Chapter 8. UPHY Lane Configuration

UPHY lanes can be configured to be owned by various IPs such as XUSB, NVME, MPHY, PCIE, NVLINK, and so on. MB1 supports NVME, UFS as boot devices for the UPHY lanes that need to be configured to access the storage in MB1 and MB2. This file defines the UPHY lane configuration that is necessary for MB1.

In T23x, BPMP-FW is loaded by MB1 and MB2 relies on BPMP-FW for UPHY configuration. Each entry in the configuration file is of the form:

Each entry in the configuration file is of the following form:

```
/ {
    uphy-lane {
        <instance-type> {
 lane-owner-map = < <id> <owner-
                            id> >,
             < <id> <owner-id> >;
        };
    };
};
```

Where:

- `<instance-type>` is the type of UPHY which needs to be configured, it can be hsio or nvhs
- `<uphy-component>` is either lane or pll which needs to be configured
- `<id>` is the lane/pll number which needs to be configured
- `<owner-id>` is the unique id of the owner to which lane/pll will be assigned

The UPHY lane configurations are kept in the `hardware/nvidia/platform/t23x/<platform>/bct/` directory.

NEW DTS example uphy lane DTS configuration file and old CFG file format:

```
/dts-v1/;

/ {
uphy-lane {
    hsio {
        lane-owner-map = <10 2>,
                 <11 1>;
    };
  };
};
```

OLD CFG format

```
//UPHY
uphy-lane.major = 1;
uphy-lane.minor = 0;
uphy-lane.hsio.lane.10 = 2;
uphy-lane.hsio.lane.11 = 1;
```

Chapter 9. OEM FW Ratchet Configuration

Roll-back prevention for `oem-fw` is controlled through the OEM-FW Ratchet configuration. Ratchetting is when older version of software is precluded from loading. The ratchet version of a software is incremented after fixing the security bugs, and this version is compared to the version stored in the Boot Component Header(BCH) of the software before loading. This file defines the minimum ratchet level for OEM-FW components. If the version in BCH is lower than the minimum ratchet level in BCT, the binary/firmware will not be loaded.

Each entry in the config file is of the following form:

```
/dts-v1/;

/ {
    ratchet {
        <loader_name1> {
            <fw_name1> = < <fw_index1>  <ratchet_value> >;
            <fw_name2> = < <fw_index2>  <ratchet_value> >;
        };
        <loader_name2> {
            <fw_name3> = < <fw_index3>  <ratchet_value> >;
        };
    };
};
```

Where:

- `<fw_index#>` is the unique index for each `oem-fw`.
- `<loader_name#>` is the name of the Boot Stage binary, which loads the firmware that corresponds to `fw_index`.
- `<fw_name#>` is the name of the firmware.
- `<ratchet_value>` is the `ratchet_value` for the firmware.

The ratchet configuration file is in the `hardware/nvidia/platform/t23x/<platform>/bct/ratchet` directory.

NEW DTS example

```
/dts-v1/;

/ {

ratchet {

    mb1 {
        mb1bct = <1  3>;
        spefw = <2 0>;
     };
     mb2 {
         cpubl = <11 5>;
       };

   };

};
```

OLD CFG format

```
//ratchet
ratchet.1.
mb1.mb1bct
= 3;
ratchet.2.mb1.spefw = 0;
ratchet.11.mb2.cpubl = 5;
```

Chapter 10. BootROM Reset PMIC Configuration

For some T23x platforms, in L1 and L2 reset boot paths, BootROM might be required to bring PMIC rails to OTP values. This process is completed by issuing I2C commands, which are encoded in AO scratch registers by MB1 and are based on the BootROM reset configuration in MB1 BCT.

- The reset cases where the BootROM issues these commands includes:
  - Watchdog 5 expiry
  - Watchdog 4 expiry
  - SC7 exit
  - SC8 exit
  - SW-Reset
  - AO-Tag/sensor reset
  - VF Sensor reset
  - HSM reset
- Each reset case can have three sets of AO blocks of commands.
- Each AO block has multiple blocks, and each block can have multiple commands.

In the configuration file, AO blocks are specified first, and then the reset conditions are initialized by using the ID of the AO blocks.

10.1 Specifying AO Blocks

Each AO block-related line in the configuration file is of the following format:

```
{
  <ResetType>-<Ao-comamnd-index> = <&AoBlock-Label>
  reset {
      <AoBlock-Label>: aoblock@<aoblock-index> {
          <parameter> = <value>;
          ...
          block@<block-index> {
              <parameter> = <value>;
          };
      };
      <AoBlock-Label>: aoblock@<aoblock-index> {
    ...
      }
  };
};
```

Node	<parameter>	Description
<reset-type>-<Ao-command-index>	<&AoBlock-Label>	• `<reset-type>` specifies the reset type and must have one of the values, `watchdog5, watchdog4, sc7,sc8, soft-reset, sensor-aotag, vfsensor, or hsm.` • `<AO-command-index>` is the index of the AO command and can have values 0, 1 or 2. Each reset path can point to maxi- mum three aoblocks • `<AoBblock-Label>` is the label given to one of the Ao Blocks
aoblock@<aoblock-index>	command-retries-count	Specifies the number of command attempts allowed for AO-block with `<aoblock-index>`
	delay-between-command-us	Specifies the delay (in microseconds), in between different commands. The delay is calculated as 1 << n microseconds where n is provided by this parameter.
	wait-before-start-bus-clear-us	Specifies the wait timeout (in microseconds), before issuing the bus clear command for given AO block. The wait time is calculated as 1 << n microseconds where n is provided by this parameter.
block@<block-index>	<command-type>/td>	`<command-type>` can only be one value - i2c-controller. That is the only one supported
	count	Specifies the number of commands in the block `<block-index>`.
	i2c-controller-id	I2C controller instance
	slave-addr	7-bit I2C slave address
	reg-data-size	Register size in bits. Valid values are 0(1-byte), 8(1-byte) and 16(2-byte),
	reg-addr-size	Register address size in bits. Valid values are 0(1-byte), 8(1-byte) and 16(2-byte)

Node	<parameter>	Description
	commands	List of `<Address Value>` pairs where value to be written to the I2c slave register address `<reg-addr>` for the command indexed by `<command- index>`.

NEW DTS example of BootROM reset configuration file:

```
/dts-v1/;

/ {
/dts-v1/;
/ {

   reset {
      // Each reset path can point to upto three aoblocks
      // This is a map of reset paths to aoblocks
      // <reset-path>-<index-pointer> = <aoblock-id>
      // index-number should be 0, 1 or 2
      // aoblock-id is the id of the one of the blocks
      mentioned above sensor-aotag-1 = <&aoblock0>;
      sc7-1 = <&aoblock2>;

      aoblock0: aoblock@0 {
         command-retries-count = <1>;
         delay-between-commands-us = <1>;
         wait-before-start-bus-clear-us = <1>;

         block@0 {
            i2c-controller;
            slave-add = <0x3c>; // 7BIt:0x3c reg-data-size = <8>;
            reg-add-size = <8>;
            commands {
               command@0 {
                  reg-addr = <0x42>; value =  <0xda>;
               };
               command@1 {
                  reg-addr =
                  <0x41>;
                  value =
                  <0xf8>;
               };
            };
         };
      };
      // Shutdown: Set MAX77620
      // Register ONOFFCNFG2, bit SFT_RST_WK = <0>
      // Register ONOFFCNFG1, bit SFT_RST
```

```
        = <1> aoblock1: aoblock@1 {
            // Shutdown: Set MAX77620
            // Register ONOFFCNFG2, bit SFT_RST_WK = <0>
            // Register ONOFFCNFG1, bit
            SFT_RST = <1> command-retries-
            count = <1>;
            delay-between-commands-us = <1>;
            wait-before-start-bus-clear-us =
            <1>; block@0 {
                i2c-controller;
                slave-add = <0x3c>; //
                7BIt:0x3c reg-data-size
                = <8>;
                reg-add-size = <8>;
                commands {
                    command@0 {
                        reg-addr = <0x42>; value =  <0x5a>;
                    };
                    command@1 {
                        reg-addr = <0x41>; value = <0xf8>;
                    };
                };
            };
        };
        // SC7 exit
        // Clear PMC_IMPL_DPD_ENABLE_0[ON]=0
        during SC7 exit aoblock2: aoblock@2 {
            command-retries-count = <1>;
            delay-between-commands-us = <256>;
            wait-before-start-bus-clear-us = <1>; block@0 {
                mmio;
                command
                s {
                    command@0 {
                        reg-addr =
                        <0x0c360010>;
                        value = <0x0>;
                    };
                };
            };
        };
    };
};
```

OLD CFG format

```
/ CFG Version 1.0
// This contains the BOOTROM commands in MB1 for
multiple reset paths. reset.major = 1;
reset.minor = 0;

// Automatic power cycling: Set MAX77620
// Register ONOFFCNFG2, bit SFT_RST_WK = 1 (default is "0" after cold boot),
// Register ONOFFCNFG1, bit
SFT_RST = 1
reset.aoblock[0].command-
retries-count = 1;
reset.aoblock[0].delay-between-commands-us = 1;
reset.aoblock[0].wait-before-start-bus-clear-us  =  1;

reset.aoblock[0].block[0].type = 1; // I2C
Type reset.aoblock[0].block[0].slave-add =
0x3c; // 7BIt:0x3c
reset.aoblock[0].block[0].reg-data-size =
8;
reset.aoblock[0].block[0].reg-add-size =
8;
reset.aoblock[0].block[0].commands[0].0x4
2 = 0xda;
reset.aoblock[0].block[0].commands[1].0x4
1 = 0xf8;

// Shutdown: Set MAX77620
// Register ONOFFCNFG2, bit SFT_RST_WK = 0
// Register ONOFFCNFG1, bit
SFT_RST = 1
reset.aoblock[1].command-
retries-count = 1;
reset.aoblock[1].delay-between-commands-us = 1;
reset.aoblock[1].wait-before-start-bus-clear-us  =  1;

reset.aoblock[1].block[0].type = 1; // I2C
Type reset.aoblock[1].block[0].slave-add =
0x3c; // 7BIt:0x3c
reset.aoblock[1].block[0].reg-data-size =
8;
reset.aoblock[1].block[0].reg-add-size =
8;
reset.aoblock[1].block[0].commands[0].0x4
2 = 0x5a;
reset.aoblock[1].block[0].commands[1].0x4
1 = 0xf8;
```

```
// SC7 exit
// Clear PMC_IMPL_DPD_ENABLE_0[ON]=0
during SC7 exit
reset.aoblock[2].command-retries-count
= 1;
reset.aoblock[2].delay-between-commands-us = 256;
reset.aoblock[2].wait-before-start-bus-clear-us  =  1;

reset.aoblock[2].block[0].type = 0; // MMIO Type
reset.aoblock[2].block[0].commands[0].0x0c360010 = 0x0;

// Shutdown in sensor/ao-tag
// no commands for other
case reset.sensor-
aotag.aocommand[0] = 1;
reset.sc7.aocommand[0] = 2;
```

Chapter 11. Miscellaneous Configuration

The different settings that do not fit into the other categories are documented in the miscellaneous configuration file.

11.1 MB1 Feature Fields

Field	Descriptions
disable_spe	• 0: Enables load of SPE-FW by MB1. • 1: Enables load of SPE-FW by MB1.
enable_dram_page_blacklistin g	• 0: Disables DRAM ECC page blacklisting feature. • 1: Enables DRAM ECC page blacklisting feature.
disable_sc7	• 0: Enables SC7-entry/exit support. • 1: Enables SC7-entry/exit support.
disable_fuse_visibility	Certain fuses cannot be read or written by default because they are not visi ble. • 0: Keeps the default visibility of fuses. • 1: Enables visibility of such fuses.
enable_vpr_resize	• 0: VPR is allocated based on McVideoProtectSizeMb and McVideoProtect, WriteAccess fields of SDRAM config file. • 1: VPR-resize feature is enabled (for example, no VPR carveout is allocated by MB1 and TZ write access to VPR carveout is enabled).
l2_mss_encrypt_regeneration	On L2 RAMDUMP reset, regenerate MSS encryption keys for the carveouts. This is a bit-field with the bit-mapping: • 1:TZDRAM • 2:VPR • 3:GSC
se_ctx_save_tz_lock	Restrict SE context save and SHA_CTX_INTEGRITY operation to TZ.
disable_mb2_glitch_protection	Disable checks on DCLS faults, TCM parity error, TCM and cache ECC

Field	Descriptions
enable_dram_error_injection	• 0: Disable DRAM error injection tests • 1: Enable DRAM error injection tests
enable_dram_staged_scrubbi ng	• 0: If DRAM ECC is enabled, scrub entire DRAM • 1: If DRAM ECC is enabled, scrub DRAM in stages - each BL responsible for the DRAM portions that it uses.
wait_for_debugger_connection	• 0: Don't wait for debugger connection at end of MB1 • 1: Spin in a while(1) loop at end of MB1 for debugger connection
limit_l1_boot_client_freq	0: Keep boot client frequencies (BPMP, SE, CBB, etc) same for L0 and L1 reset

11.1.1 Clock Data

These fields allow certain clock-related customization.

Field	Description
bpmp_cpu_nic_divide r	Controls BPMP CPU frequency 0: Skip programming CLK_SOURCE_BPMP_CPU_NIC[BPMP_CPU_NIC_CLK_DI, VISOR] non-zero: 1 + Value to be programmed in CLK_SOURCE_BPMP_CPU_NIC[BPMP, _CPU_NIC_CLK_DIVISOR]
bpmp_apb_divider	Controls BPMP APB frequency • 0: Skip programming of CLK_SOURCE_BPMP_APB[BPMP_APB_CLK_DIVISOR] non-zero: • 1 + Value to be programmed in CLK_SOURCE_BPMP_APB[BPMP_APB,_CLK_DIVISOR]
axi_cbb_divider	Controls CBB (control backbone) frequency • 0: Skip programming of CLK_SOURCE_AXI_CBB[AXI_CBB_CLK_DIVISOR] non-zero: • 1 + Value to be programmed in CLK_SOURCE_AXI_CBB[AXI_CBB_CL←, K_DIVISOR]
se_divider	Controls SE (security engine) frequency • 0: Skip programming of CLK_SOURCE_SE[SE_CLK_DIVISOR] • non-zero: 1 + Value to be programmed in CLK_SOURCE_SE[SE_CLK_DIVISOR]
aon_cpu_nic_divider	Controls AON/SPE CPU frequency • 0: Skip programming of CLK_SOURCE_AON_CPU_NIC[AON_CPU_NIC_CLK_DI←, VISOR] non-zero: • 1 + Value to be programmed in CLK_SOURCE_AON_CPU_NIC[AON_C, PU_NIC_CLK_DIVISOR]

Field	Description
aon_apb_divider	Controls AON APB frequency • 0: Skip programming of CLK_SOURCE_AON_CPU_NIC[AON_CPU_NIC_CLK_DI↩, VISOR] • non-zero: 1 + Value to be programmed in CLK_SOURCE_AON_CPU_NIC[AON_C, PU_NIC_CLK_DIVISOR]
aon_can0_divider	Controls AON CAN1 frequency • 0: Skip programming of CLK_SOURCE_CAN1[CAN1_CLK_DIVISOR] • non-zero: 1 + Value to be programmed in CLK_SOURCE_CAN1[CAN1_CLK_DIVI, SOR]
aon_can1_divider	Controls AON CAN2 frequency • 0: Skip programming of CLK_SOURCE_CAN2[CAN2_CLK_DIVISOR] • non-zero: 1 + Value to be programmed in CLK_SOURCE_CAN2[CAN2_CLK_DIVI, SOR]
osc_drive_strength	Oscillator drive strength
pllaon_divn	DIVN value of PLLAON • 0: Use PLLAON_DIVN = 30, PLLAON_DIVM = 1, PLLAON_DIVP = 2 • non-zero: 1 + Value to be programmed in PLLAON_BASE[PLLAON_DIVN]
pllaon_divm	DIVM value of PLLAON (ignored when clock.pllaon_divn = 0) 1 + Value to be programmed in PLLAON_BASE[PLLAON_DIVM]
pllaon_divp	DIVP value of PLLAON (ignored when clock.pllaon_divn = 0) 1 + Value to be programmed in PLLAON_BASE[PLLAON_DIVP]
pllaon_divn_frac	Value to be programmed

11.1.2 AST Data

MB1/MB2 uses the address-translation (AST) module to map the DRAM carveout for different firmware in the 32bit virtual address space of the various auxiliary processor clusters.

11.1.3 MB1 AST Data

Field	Description
mb2_va	Virtual address for MB2 carveout in BPMP-R5 address-space.
spe_fw_va	Virtual address for SPE-FW carveout in AON-R5 address-space.
misc_carveout_va	Virtual address for MISC carveout in SCE-R5 address-space.
rcm_blob_carveout_va	Virtual address for RCM-blob carveout in SCE-R5 address-space.
temp_map_a_carveout_va	Virtual address for temporary mapping A used while loading binaries
temp_map_a_carveout_size	Size for temporary mapping A used while loading binaries

temp_map_a_carveout_va	Virtual address for temporary mapping B used while loading binaries
temp_map_a_carveout_siz e	Size for temporary mapping B used while loading binaries

Note: None of the above VA spaces should overlap with MMIO region or with each other.

- Size fields should be power of 2
- VA fields should be aligned to their mapping/carveouts.

11.1.4 Carveout Configuration

Although "SDRAM Configuration" on page 59 has MC carveout's preferred base, size and permissions, it does not have all information required to allocate the carveouts by MB1. This additional information is specified using miscellaneous configuration file.

For carveouts that are not protected by MC, all information, including size and preferred base address, is specified by using the miscellaneous configuration file.

Each MC carveout configuration parameter is of the following form:

```
/ {
    misc {
        carveout {
            <carveout-type> {
                <parameter> = <value>;
            };
        };
    };
};
```

where:

- `<carveout-type>` identifies the carveout and is one of the following:

carveout-type	Description
gsc@[1-31]	GSC carveout for various purposes
mts	MTS/CPU-uCode carveout
vpr	VPR carveout
tzdram	TZDRAM carveout used for SecureOS
os	OS carveout used for loading OS kernel
rcm	RCM carveout used for loading RCM-blob during RCM mode (temporary boot carveout)

- `<parameter>` is one of the following parameters:

Parameter	Description
pref_base	Preferred base address of carveout
size	Size of carveout (in bytes)
alignment	Alignment of base address of carveout (in bytes)
ecc_protected	When DRAM region-based ECC is enabled and there are non-ECC protected DR, AM regions, whether to allocate the carveout from ECC protected region 0: Allocate from non ECC protected region 1: Allocate from ECC protected region
bad_page_tolerant	When DRAM page blacklisting is enabled, whether it is ok to have bad pages in the carveout (only possible for very large carveouts and which are handled completely by component that can avoid bad pages using SMMU/MMU) 0: No bad pages allowed for the carveout 1: Bad pages allowed for the carveout (allocation can be done without filtering bad pages)

The valid combination of the carveouts and their parameters are specified in following table:

Supported carveout-type	pref_base	size	alignme nt	ecc_protected	bad_page_ tolerant
gsc-[1-31], mts, vpr	N/A	N/A	YES	YES	YES
tzdram, mb2, cpubl, misc, os, rcm	YES	YES	YES	YES	YES

11.1.5 Coresight Data

Field	Description
cfg_system_ctl	Value to be programmed to CORESIGHT_CFG_SYSTEM_CTL
cfg_csite_mc_wr_ctrl	Value to be programmed to CORESIGHT_CFG_CSITE_MC_WR_CTRL
cfg_csite_mc_rd_ctrl	Value to be programmed to CORESIGHT_CFG_CSITE_MC_RD_CTRL
cfg_etr_mc_wr_ctrl	Value to be programmed to CORESIGHT_CFG_ETR_MC_WR_CTRL
cfg_etr_mc_rd_ctrl	Value to be programmed to CORESIGHT_CFG_ETR_MC_RD_CTRL
cfg_csite_cbb_wr_ctrl	Value to be programmed to CORESIGHT_CFG_CSITE_CBB_WR_CTRL
cfg_csite_cbb_rd_ctrl	Value to be programmed to CORESIGHT_CFG_CSITE_CBB_RD_CTRL

11.1.6 Firmware Load and Entry Configuration

Firmware configuration is specified as follows:

```
/{
    misc {
        ...
        ...
        firmware {
            <firmware-type> {
                <parameter> = <value>;
            };
        }
    };
};
```

Where `<firmware-type>` is one of the mb2 or tzram-el3 and <parameter> is specified in below table

Field	Description
load-offset	Offset in <firmware> carveout where <firmware> binary is loaded.
entry-offset	Offset of <firmware> entry point in <firmware> carveout.

11.1.7 CPU Configuration

Field	Description
ccplex_platform_features	CPU platform features (should be 0)
clock_mode.clock_burst_policy	CCPLEX clock burst policy
clock_mode.max_avfs_mode	Highest CCPLEX AVFS mode
nafll_cfg2/fll_init	CCPLEX NAFLL CFG2 [Fll Init]
nafll_cfg2/fll_ldmem	CCPLEX NAFLL CFG2 [Fll Ldmem]
nafll_cfg2/fll_switch_ldmem	CCPLEX NAFLL CFG2 [Fll Switch Ldmem]
nafll_cfg3	CCPLEX NAFLL CFG3
nafll_ctrl1	CCPLEX NAFLL CTRL1
nafll_ctrl2	CCPLEX NAFLL CTRL2
lut_sw_freq_req/sw_override_ndiv	SW override for CCPLEX LUT frequency request
lut_sw_freq_req/ndiv	NDIV for CCPLEX LUT frequency request
lut_sw_freq_req/vfgain	VFGAIN for CCPLEX LUT frequency request
lut_sw_freq_req/sw_override_vfgain	VFGAIN override for CCPLEX LUT frequency request
nafll_coeff/mdiv	MDIV for NAFLL coefficient
nafll_coeff/pdiv	PDIV for NAFLL coefficient
nafll_coeff/fll_frug_main	FLL frug main for NAFLL coefficient

Field	Description
nafll_coeff/fll_frug_fast	FLL frug fast for NAFLL coefficient
adc_vmon.enable	Enable CCPLEX ADC voltage monitor
min_adc_fuse_rev	Minimum ADC fuse revision
pllx_base/divm	PLLX DIVM
pllx_base/divn	PLLX DIVN
pllx_base/divp	PLLX DIVP
pllx_base/enable	Enable PLLX
pllx_base/bypass	PLLX Bypass Enable

11.1.8 Other Configuration

Field	Description
aocluster.evp_reset_addr	AON/SPE reset vector (in SPE BTCM)
carveout_alloc_direction	Carveout allocation direction 0: End of DRAM 1: End of 2GB DRAM (32b address space) 2: Start of DRAM
se_oem_group	Value to be programmed to SE0_OEM_GROUP_0 / SE_RNG1_RNG1_OEM_G ROUP_0 (NVHS / NVLS / Lock bits are forced set)
i2c-freq	List of < <i2c controller instance> <Frequency (in KHz) of I2C controller Instance > where, I2C controller instance has valid values 0 to 8.

11.1.8.1 MB1 Soft Fuse Configurations

There are certain platform-specific configurations or decisions in MB1 that are required before th estorage is initialized and MB1-BCT is read (for example, debug port details, certain boot-mode controls, behavior in case of failure, and so on.). To address this requirement, an array of 64 fields has been added to signed section of BR-BCT that is opaque to BR and will be consumed by MB1. In recovery mode, BR does not read BR-BCT, so this array is kept part of RCM message and is copied by BR to the location that BR would have kept it in coldboot as part of BR-BCT. This array is called MB1 software fuses (soft-fuses).

11.1.8.2 Debug Controls

Field	Description
verbosity	Controls verbosity of debug logs on UART (verbosity increases with increasing value): • 0: UART logs disabled • 1: Critical prints only 2: Error • 3: Warn

	• 4: Info • 5: Debug
uart_instance	UART controller number where debug logs are spewed: • 0: UARTA (only for simulation platforms) • 2: UARTC (open-box debug) • 5: UARTF (closed-box debug, USB-Type C over DP_AUX pins) • 7: UARTH (closed-box debug, USB-Type C over USB-OTG
usb_2_nvjtag	On-chip controller connected to the USB2 pins: • 0: ARMJTAG • 1: NVJTAG
swd_usb_port_sel	USB2 port over which SWD should be configured: • 0: USB2 Port0 • 1: USB2 Port1
uart8_usb_port_sel	USB2 port over which UART should be configured: • 0: USB2 Port0 • 1: USB2 Port1
wdt_enable	Enable BPMP WDT 5th-expiry during execution of MB1/MB2 (boolean).
wdt_period_secs	BPMP WDT period (in secs) per expiry.

11.1.8.3 Boot Failure Controls

Field	Description
switch_bootchain	Switch boot chain in case of failure (boolean)
reset_to_recovery	Trigger L1 RCM reset on failure (boolean)
bootchain_switch_mechanism	Used if switch_bootchain is set to 1: • 0: Use BR-based boot-chain switching • 1: Use Android A/B based boot-chain switching
bootchain_retry_count	Maximum number of retries for single boot-chain (0-15).

11.1.8.4 On/Off IST Mode Controls

Field	Description
platform_detection_flow	In RCM mode, enter platform detection flow (boolean)
enable_tegrashell	In RCM mode, enter tegrashell mode (Allowed only if `FUSE_SECURITY_MODE_0` is not blown) (boolean).
enable_IST	Enable Key ON/OFF IST boot mode (boolean).
enable_L0_IST	Enable Key ON (L0) IST boot mode (boolean). Used only if EnableIST is set to 1.
enable_dgpu_IST	Enable dGPU IST during IST boot modes (boolean).

11.1.8.5 Frequency Monitor Controls

These should be documented for OEMs after review with security team.

Field	Description
vrefRO_calib_override	Override VrefRO Calibration Override based on SoftFuse instead of fuse (boolean)
vrefRO_min_rev_threshold	Program VrefRO frequency adjustment target based on FUSE_VREF_CALIB_0 if (FUSE_ECO_RESERVE_1[3:0] > vrefRO_min_rev_threshold) (Used if vrefRO_calib_override is set to 0)
vrefRO_calib_val	VrefRO Calibration Value used to program VrefRO frequency adjustment target (Used if vrefRO_calib_override is set to 1)
osc_threshold_low	Lower Threshold of FMON counter for OSC clock
osc_threshold_high	Upper Threshold of FMON counter for OSC clock
pmc_threshold_low	Lower Threshold of FMON counter for 32K clock
pmc_threshold_high	Upper Threshold of FMON counter for 32K clock

The miscellaneous configuration files are in the `bct/t23x/misc/` directory.

NEW DTS example of miscellaneous configuration file:

```
/dts-v1/;

/ {
    misc {
        disable_spe = <0>;
        enable_vpr_resize = <0>;
        disable_sc7 = <1>;
        disable_fuse_visibility = <0>;
        disable_mb2_glitch_protection = <0>;
```

```
carveout_alloc_direction = <2>;

//Specify i2c bus speed in KHz
i2c_freqency = <4 918>;

// Soft fuse configurations
verbosity = <4>; // 0: Disabled: 1: Critical, 2: Error, 3: Warn, 4: Info, 5:
Debug
uart_instance = <2>;
emulate_prod_flow = <0>; // 1: emulate prod flow. For
pre-prod only switch_bootchain = <0>; // 1: Switch to
alternate Boot Chain on failure
bootchain_switch_mechanism = <1>; // 1: use a/b boot
reset_to_recovery = <1>; // 1: Switch to Forced Recovery on failure
platform_detection_flow = <0>; //0: Boot in RCM flow for platform
detection nv3p_checkSum = <1>; //1 Check-sum enabled for Nv3P
command
vrefRO_calib_override = <0>; //1: program VrefRO as per soft fuse VrefRO
calibration value vrefRO_min_rev_tThreshold = <1>;
vrefRO_calib_val = <0>;
osc_threshold_low =
<0x30E>;
osc_threshold_high =
<0x375>;
pmc_threshold_low =
<0x59E>;
pmc_threshold_high =
<0x64E>;
bootchain_retry_count = <0>; //Specifies the max count to retry a particular
boot chain, Max value is

cpu {
   ////////// cpu variables
   ////////// nafll_cfg3 =
   <0x38000000>; nafll_ctrl1 =
   <0x0000000C>; nafll_ctrl2 =
   <0x22250000>;
   adc_vmon.enable = <0x0>;
   min_adc_fuse_rev = <1>;
   ccplex_platform_features =
   <0x00000>; clock_mode {
      clock_burst_policy = <15>;
      max_avfs_mode = <0x2E>;
   };
   lut_sw_freq_req {
      sw_override_ndiv =
      <3>;
      ndiv = <102>;
      vfgain = <2>;
      sw_override_vfgain = <3>;
   };

   nafll_coeff {
      mdiv =
            <3>;
            pdiv =
            <0x1>;
            fll_frug_main = <0x9>;
            fll_frug_fast = <0xb>;
        };
        nafll_cfg2 {
            fll_init = <0xd>;
            fll_ldmem = <0xc>;
            fll_switch_ldmem =
            <0xa>;
        };
        pllx_base {
            divm = <2>;
            divn = <104>;
            divp = <2>;
            enable = <1>;
            bypass = <0>;
        };
    };
    wp {
        waypoint0 {
            rails_shorted = <1>;
            vsense0_cg0 = <1>;
        };
    };
    ////////// sw_carveout variables
    ////////// carveout {
        misc {
            size = <0x800000>; //
            8MB alignment =
            <0x800000>; // 8MB
        };
        os {
            size = <0x08000000>; //128MB
            pref_base =
            <0x80000000>; alignment
            = <0x200000>;
        };
        cpubl {
            alignment = <0x200000>;
            size = <0x04000000>; // 64MB
        };
        rcm {
            size =
            <0x0>;
            alignment =
            <0>;
        };
        mb2 {
            size = <0x01000000>; //16MB
            alignment = <0x01000000>; //16MB
        };
        tzdram {
            size = <0x01000000>; //16MB
            alignment = <0x00100000>; //1MB
        };
        ////////// mc carveout alignment
        ////////// vpr {
            alignment = <0x100000>;
        };
        gsc@6 {
            alignment = <0x400000>;
        };
        gsc@7 {
            alignment = <0x100000>;
        };
        gsc@8 {
            alignment = <0x100000>;
        };
        gsc@9 {
            alignment = <0x800000>;
        };
        gsc@10 {
            alignment = <0x100000>;
        };
        gsc@12 {
            alignment = <0x100000>;
        };
        gsc@17 {
            alignment = <0x200000>;
        };
        gsc@19 {
            alignment = <0x2000000>;
        };
        gsc@24 {
            alignment = <0x200000>;
        };
        gsc@27 {
            alignment = <0x200000>;
        };
        gsc@28 {
            alignment = <0x200000>;
        };
        gsc@29 {
            alignment = <0x200000>;
        };
    };

    ////////// mb1 ast va //////////
```

```
        ast{
           mb2_va = <0x52000000>;
           misc_carveout_va =
           <0x70000000>;
           rcm_blob_carveout_va =
           <0x60000000>;
           temp_map_a_carveout_va =
           <0x80000000>;
           temp_map_a_carveout_size =
           <0x40000000>;
           temp_map_b_carveout_va =
           <0xc0000000>;
           temp_map_b_carveout_size =
           <0x20000000>;
        };
        ////////// clock variables
        ////////// clock {
           pllaon_divp =
           <0x3>;
           pllaon_divn =
           <0x1F>;
           pllaon_divm =
           <0x1>;
           pllaon_divn_frac = <0x03E84000>;
           // For bpmp_cpu_nic, bpmp_apb, axi_cbb and se,
           // specify the divider with PLLP_OUT0 (408MHz) as source.
           // For aon_cpu_nic, aon_can0 and aon_can1,
           // specify the divider with PLLAON_OUT as source.
           // In both cases, BCT specified value = (1 + expected divider
           value). bpmp_cpu_nic_divider = <1>;
           bpmp_apb_divider = <1>;
           axi_cbb_divider = <1>;
           se_divider = <1>;
        };
        ////////// aotag variables
        ////////// aotag {
           boot_temp_threshold = <97000>;
           cooldown_temp_threshold = <87000>;
           cooldown_temp_timeout = <30000>;
           enable_shutdown = <1>;
        };

        //////// aocluster data
        //////// aocluster {
           evp_reset_addr = <0xc480000>;
        };
     };
  };
```

OLD CFG format

```
disable_spe = 0;
enable_vpr_resize = 0;
disable_sc7 = 1;
disable_fuse_visibility = 0;
disable_mb2_glitch_protection = 0;
carveout_alloc_direction = 2;
////////// cpu variables //////////
cpu.ccplex_platform_features = 0x00000;
cpu.clock_mode.clock_burst_policy = 15;
cpu.clock_mode.max_avfs_mode = 0x2E;
cpu.lut_sw_freq_req.sw_override_ndiv = 3;
cpu.lut_sw_freq_req.ndiv = 102;
cpu.lut_sw_freq_req.vfgain = 2;
cpu.lut_sw_freq_req.sw_override_vfgain = 3;
cpu.nafll_coeff.mdiv = 3;
cpu.nafll_coeff.pdiv = 0x1;
cpu.nafll_coeff.fll_frug_main =
0x9;
cpu.nafll_coeff.fll_frug_fast =
0xb; cpu.nafll_cfg2.fll_init =
0xd; cpu.nafll_cfg2.fll_ldmem =
0xc;
cpu.nafll_cfg2.fll_switch_ldmem
= 0xa; cpu.nafll_cfg3 =
0x38000000; cpu.nafll_ctrl1 =
0x0000000C;
cpu.nafll_ctrl2 =
0x22250000;
cpu.adc_vmon.enable =
0x0; cpu.pllx_base.divm
= 2;
cpu.pllx_base.divn = 104;
cpu.pllx_base.divp = 2;
cpu.pllx_base.enable = 1;
cpu.pllx_base.bypass = 0;
cpu.min_adc_fuse_rev = 1;

wp.waypoint0.rails_shorted = 1;
wp.waypoint0.vsense0_cg0 = 1;
////////// sw_carveout variables
////////// carveout.misc.size =
0x800000; // 8MB
carveout.misc.alignment =
0x800000; // 8MB
carveout.os.size = 0x08000000;
//128MB carveout.os.pref_base =
0x80000000;
carveout.os.alignment =
0x200000;
carveout.cpubl.alignment =
0x200000;
firmware.cpubl_load_offset =
0x600000; carveout.cpubl.size =
0x04000000; // 64MB
carveout.rcm.size = 0x0;
carveout.rcm.alignment = 0;
carveout.mb2.size = 0x01000000;
//16MB carveout.mb2.alignment =
0x01000000; //16MB
carveout.tzdram.size =
0x01000000; //16MB
carveout.tzdram.alignment = 0x00100000; //1MB
////////// mc carveout alignment
////////// carveout.vpr.alignment =
0x100000; carveout.gsc[6].alignment =
0x400000; carveout.gsc[7].alignment =
0x800000; carveout.gsc[8].alignment =
0x100000; carveout.gsc[9].alignment =
0x100000; carveout.gsc[10].alignment =
0x800000; carveout.gsc[12].alignment =
0x100000; carveout.gsc[17].alignment =
0x100000; carveout.gsc[19].alignment =
0x200000; carveout.gsc[24].alignment =
0x2000000; carveout.gsc[27].alignment =
0x200000; carveout.gsc[28].alignment =
0x200000; carveout.gsc[29].alignment =
0x200000;

////////// mb1 ast va
////////// ast.mb2_va =
0x52000000;
ast.misc_carveout_va =
0x70000000;
ast.rcm_blob_carveout_va =
0x60000000;
ast.temp_map_a_carveout_va =
0x80000000;
ast.temp_map_a_carveout_size =
0x40000000;
ast.temp_map_b_carveout_va =
0xc0000000;
ast.temp_map_b_carveout_size =
0x20000000;

////////// MB2 AST VA //////////
carveout.bpmp_ast_va = 0x50000000;
carveout.ape_ast_va = 0x80000000;
carveout.apr_ast_va = 0xC0000000;
carveout.sce_ast_va = 0x70000000;
carveout.rce_ast_va = 0x70000000;
carveout.camera_task_ast_va =
0x78000000;

////////// clock variables
////////// clock.pllaon_divp =
0x3; clock.pllaon_divn = 0x1F;
clock.pllaon_divm = 0x1;
clock.pllaon_divn_frac =
0x03E84000;
// For bpmp_cpu_nic, bpmp_apb, axi_cbb and se,
// specify the divider with PLLP_OUT0 (408MHz) as source.
// For aon_cpu_nic, aon_can0 and aon_can1,
// specify the divider with PLLAON_OUT as source.
// In both cases, BCT specified value = (1 + expected
divider value). clock.bpmp_cpu_nic_divider = 1;
clock.bpmp_apb_divider = 1;
clock.axi_cbb_divider = 1;
clock.se_divider = 1;
////////// aotag variables //////////

aotag.boot_temp_threshold = 97000;
aotag.cooldown_temp_threshold = 87000;
aotag.cooldown_temp_timeout = 30000;
aotag.enable_shutdown = 1;
//Specify i2c bus speed
in KHz i2c.4 = 918;

//////// aocluster data ////////
aocluster.evp_reset_addr = 0xc480000;

//////// mb2 feature flags
//////// enable_sce = 1;
enable_rce = 1;
enable_ape = 1;
enable_combined_uart = 1;
spe_uart_instance = 2;
```

Chapter 12. SDRAM Configuration

DTS format for SDRAM configuration:

```
/
{
 sdram {
  mem_cfg_<N>: mem-cfg@<N> {
    <parameter> = <value>;
    };
  };
};

&mem_cfg_<N> {
  #include " <mem_override_dts >"
};
```

where

- <N> is number starting from 0 representing different memory configurations
- <parameter> is the SDRAM parameter. Usually, this corresponds to MC/EMC register.
- <value> is the 32bit value of the corresponding register.
- <mem_override_dts> is override file for SDRAM parameters which will be applied to all the configs

```
      Format of override dts will be:
          <parameter> = <value>; // example McVideoProtectBom = <0x00000000>;
```

Memory/MSS HW team will use a tool provided by SW to convert legacy configuration format to above DTS format.

12.1 Carveouts

The following hardware carveouts are available:

- GSC carveouts
- Non-GSC carveouts Important carveout fields in BCT cfg

Important carveout fields in BCT cfg.

Field	Description
McGeneralizedCarveout<N>Bom and Mc GeneralizedCarveout<N>BomHi (where N is the GSC#)	Preferred base address of the GSC carveout (recommended value = 0; allows MB1 to allocate on its own)
McGeneralizedCarveout<N>Size128kb (where N is the GSC#)	31:27 - Size of the generalized security carveout region with 4kb granularity 11:0 - Size of the generalized security carveout region with 128kb granularity Size = (MC_SECURITY_CARVEOUT1_SIZE_RAN, GE_128KB <<17) /(MC_SECURITY_CARVEOU, T1_SIZE_RANGE_4KB << 12)
McGeneralizedCarveout<N>Access[0-5] (where N is the GSC#)	Client Access Register for Generalized Carveout. For bit description details refer to TRM
McVideoProtectBom and McVideoProtectBomHi	Preferred base address of VPR carveout (recommended value = 0; allows MB1 to allocate on its own)
McVideoProtectSizeMb	Specifies VPR carveout size (in MB) (see also `enable_vpr_resize` in "Miscellaneous Configuration).

12.2 GSC Carveouts

The following list gives a mapping between Carveout names and its corresponding GSC numbers:

Carveout #	Carveout Name
1	NVDEC
2	WPR1
3	WPR2
4	TSECA
5	TSECB
6	BPMP
7	APE
8	SPE
9	SCE
10	APR
11	TZRAM
12	IPC_SE_TSEC
13	BPMP_RCE

Carveout #	Carveout Name
14	BPMP_DMCE
15	SE_SC7
16	BPMP_SPE
17	RCE
18	CPUTZ_BPMP
19	VM_ENCRYPT1
20	CPU_NS_BPMP
21	OEM_SC7
22	IPC_SE_SPE_SCE_BPMP
23	SC7_RF
24	CAMERA_TASK
25	SCE_BPMP
26	CV
27	VM_ENCRYPT2
28	HYPERVISOR
29	SMMU

12.3 Non-GSC Carveouts

Carveout #	Carveout Name
32	MTS
33	VPR

Chapter 13. GPIO Interrupt Mapping Configuration

To reduce the interrupt hunt time for GPIO pins from single ISR, in T23x, each GPO controller has 8 interrupt lines to LIC. This gives opportunity to map the GPIO pin to any of 8 interrupts. The configuration for the same is specified in the GPIO interrupt configuration file.

Each entry in this configuration file is in the following form:

```
gpio-intmap {
    port@<PORT-ID> {
        pin-<N>-int-line = <INTERRUPT-NUMBER>;
    };
    port@<PORT-ID> {
        pin-<N>-int-line = <INTERRUPT-NUMBER>;
    };
};
```

where:

- `<PORT-ID>` is the port name like A, B, C..Z, AA, BB
- `<N>` is the pin id in the port. Valid values are 0-7.
- `<INTERRUPT-NUMBER>` is interrupt route for that pin. Valid values are 0-7.

The gpio-interrupt mapping configuration file are kept at hardware/nvidia/platform/t23x/<platform>/bct/ NEW DTS example of GPIO interrupt mapping configuration file:

```
 /dts-v1/;

 / {
gpio-intmap {
     port@B {
            pin-1-int-line = <0>; // GPIO B1 to INT0
        };
        port@AA {
            pin-0-int-line  =  <0>;  //  GPIO
            AA0  to  INT0  pin-1-int-line  =
            <0>; // GPIO AA1 to INT0 pin-2-
            int-line = <0>; // GPIO AA2 to
            INT0
        };
    };

};
```

OLD CFG format

```
gpio-intmap.port.B.pin.1 = 0; // GPIO
B1 to INT0 gpio-intmap.port.AA.pin.0 =
0; // GPIO AA0 to INT0 gpio-
intmap.port.AA.pin.1 = 0; // GPIO AA1
to INT0 gpio-intmap.port.AA.pin.2 = 0;
// GPIO AA2 to INT0
```

Chapter 14. MB2 BCT Misc Configuration

14.1 MB2 Feature Fields

These feature bits are boolean flags that enable or disable certain functionality in MB2.

Field	Description
disable-cpu-l2ecc	If this property is present CPU L2 ECC id disabled. Otherwise, it is enabled.
enable-combined-uart	If this property is present combined uart is enabled Otherwise, it is disabled.
spe-uart-instance	UART controller used for combined UART by SPE.

14.2 MB2 Firmware Data

Mb2 firmware configuration is specified as:

```
/ {
    mb2-misc {
        <firmware-type> {
            <parameter> = <value>;
        };
    };
};
```

`Where firmware type is one of the cpubl, ape-fw, bpmp-fw, rce-fw, sce-fw, camera-task-fw, apr-fw.`

`<parameter>` is one of the parameters from below table.

Parameters	Description
enable	Enable loading of firmware from MB2, if this property is present Otherwise disable loading of firmware from MB2.
ast-va	Virtual address for firmware carveout in BPMP-R5 address-space.
load-offset	Offset in firmware carveout where firmware binary is loaded.
entry-offset	Offset of firmware entry point in firmware carveout.

Example of an Mb2 misc DTS configuration file:

```
/dts-v1/;

/ {
    mb2-misc {
        disable-cpu-
        l2ecc; enable-
        combined-uart;
        spe-uart-instance = <0x2>;

    firmware {
         sce {
                ast-va = <0x70000000>;
            };
            ape {
                enable;
                ast-va = <0x80000000>;
            };
            rce {
                enable;
                ast-va = <0x70000000>;
            };
            cpubl {
                load-offset = <0x600000>;
            };
            apr {
                ast-va = <0xC0000000>;
            };

            camera-task {
                ast-va = <0x78000000>;
            };
        };
    };
};
```

Chapter 15. Security Configuration

MB1 and MB2 program most of the SCRs and firewalls in T23x. The list of SCRs/firewalls, their order, and their addresses is predetermined. The values are taken from the SCR configuration file.

Each entry in this configuration file is in the following form:

```
/ {
    scr {
        reg@<index> {
          <parameter> = <value>;
        };
    };
};
```

where:

- `<index>` is the index of the SCR/firewall in the predefined list
- `<parameter>` and its <value> can be as follows:

Parameter	Value	
exclusion-info	Exclusion info is a bit field defined as follows	
	Bit	**Description**
	0	Do not program in coldboot or SC7-exit
	1	Program in coldboot, but skip in SC7-exit
	2	Program at end of MB2, instead of MB1
value	32 bit value for the SCR register	

Note: The values of the SCRs are programmed in increasing order of indexes and not in the order they appear in the configuration file. The scr/firewalls which are not specified in the configuration file are locked without restricting the access to the protected registers.

The scr configuration files are kept in the `hardware/nvidia/platform/t23x/common/bct/scr` directory.

NEW DTS format example of SCR config file:

```
/dts-v1/;
/ {
    scr {
         reg@161 {
             exclusion-info = <7>;
             value = <0x3f008080>;
         };

         reg@162 {
             exclusion-info = <4>;
             value = <0x18000303>;
         };

         reg@163 {
             exclusion-info = <4>;
             value = <0x18000303>;
         };
      };
};;
```

OLD CFG format

```
scr.161.7 = 0x3f008080; //
TKE_AON_SCR_WDTSCR0_0 scr.162.4
= 0x18000303; //
DMAAPB_1_SCR_BR_INTR_SCR_0
scr.163.4 = 0x18000303; //
DMAAPB_1_SCR_BR_SCR_0
```