

i.MX233 Power Management Unit and Battery Charger

by *Multimedia Applications Division*
Freescale Semiconductor, Inc.
Austin, TX

This application note describes the low-level blocks of the Power Management Unit (PMU). This document covers the basic configurations for the power supplies and power rails, and how to use these building blocks to perform useful functions in the i.MX233 such as handling power source transitions and raising and lowering power rails. This application note describes the battery charger block and provides instructions to charge a battery. The power savings features of the i.MX233 are also described.

The i.MX233 applications processor integrates a highly-efficient and comprehensive power supply. It is powered by a Li-Ion battery or 5 V power source and generates five internal power rails through two separate supplies. All the hardware in the PMU must be configured through firmware. The i.MX233 requires direct modifications to registers in the power block. This adds complexity, but also provides firmware the ability to optimize the device power beyond a typical power management unit.

The information provided in this document is intended to supplement the information provided in the i.MX233 datasheet. Register names and bitfields are frequently

Contents

1. System Overview	2
1.1. Power Sources	2
1.2. Power Supplies	3
1.3. Power Rails	3
1.4. Battery Charger	3
1.5. Hardware Requirements	4
2. Getting Started with the Power Subsystem	4
2.1. Power Sources	4
2.2. Power Supplies	9
2.3. Power Rails	17
3. Using the Power Subsystem	23
3.1. Powering the Supplies	23
3.2. Transition Between Power Sources	24
3.3. Using the Software Brownout Interrupt	26
4. Getting Started with the Battery Charging Subsystem	27
4.1. Configuring the Charger	28
4.2. Using the Battery Charger	29
5. Optimizing the Power Subsystem	31
5.1. Enabling Interrupt Wait	31
5.2. Enabling HCLK Auto-Slow	31
5.3. Setting Power Transistor Size	32
5.4. Using Pulse-Frequency Modulation (PFM) Mode ..	33
6. Conclusion	33
7. References	33

referenced. [Table 1](#) contains a glossary of terms used in this application note.

Table 1. Glossary

Term	Definition
DC-DC	The i.MX233 internal DC-DC switching converter. There is only one DC-DC switching converter on the device. Register names and bitfields sometimes refer to this as DCDC (without hyphen).
LinReg	Common abbreviation used for linear regulator.
4P2	Alternate name for the 4.2V rail. "P" is the abbreviation used for "point" when referencing the rail in the datasheet or software.

1 System Overview

The PMU contains power sources, power supplies, and power rails. Power sources provide power to the internal power supplies. The power supplies convert the input sources to voltage levels the i.MX233 can use. The supplies output to the power rails. The battery charger can be activated when 5 V is present and charges the attached Li-Ion battery. The architecture of the i.MX233 allows the battery to be fully-charged while still powering the device through a separate power supply. [Figure 1](#) provides the logical diagram of the PMU.

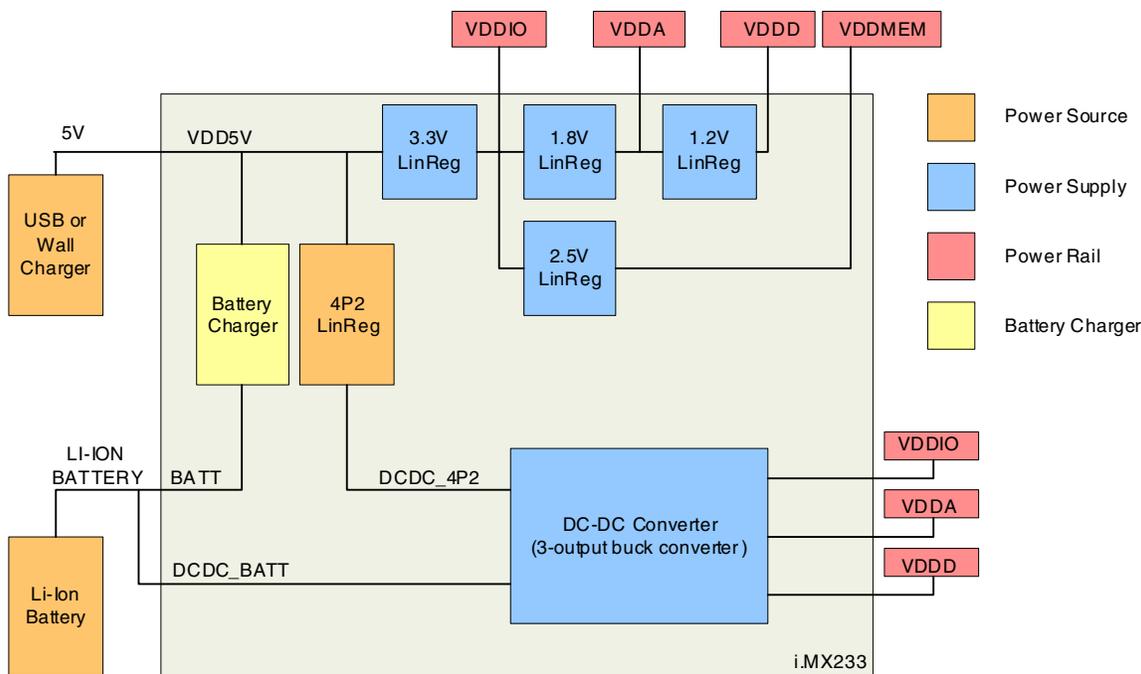


Figure 1. Logical Diagram of i.MX233 PMU

1.1 Power Sources

The i.MX233 can be powered from a Li-Ion battery or 5 V power source. The Li-Ion battery should be in the range 3.0 –4.2 V when powering the device. The 5 V power source comes from a USB connection or wall-charger that converts wall-power to 5 V. The actual 5 V voltage should not drop below 4.4 V as this

may trigger a 5 V loss event in the device. When 5 V is present, the i.MX233 can turn on an internal 4.2 V LinReg to use as an input to the DC-DC. The battery charger is also a power source though it only provides power to the battery and not directly to the system.

1.2 Power Supplies

An internal DC-DC switching converter and multiple LinRegs are available as power supplies in the i.MX233. The DC-DC is sourced from the Li-Ion battery or the internally generated 4P2 rail. The DC-DC uses one of the inputs to generate power to three output rails.

The LinRegs are a group of LinRegs that each take a single input and regulate the voltage to their target output level. The LinRegs for the power rails are daisy-chained from higher output voltage to lower output voltage. In cases where the input is an internal power rail, the LinReg can alternatively be powered from the DC-DC output of that rail.

The 4.2 V LinReg uses the 5 V source to provide 4.2 V to the DC-DC. This output is called the 4P2 rail and it only goes to the DC-DC. This allows the i.MX233 to use the more efficient DC-DC converter instead of the internal LinRegs when 5 V is present. The most power efficient system uses the DC-DC converter as much as possible regardless of the power source. The LinRegs should only be used to power the rails during boot-up while the 4.2 V LinReg is being initialized.

NOTE

The 4P2 LinReg is referred to as an internal power source for i.MX233. It also referred to as the 4P2 power rail since it creates an output rail. For this application note, the 4P2 power source refers to the actual LinReg and the 4P2 rail refers to the output of the LinReg.

1.3 Power Rails

The i.MX233 has four output power rails:

- VDDD—provides power to the digital portions of the device. The system clocks use the VDDD power rail.
- VDDA—provides power primarily for audio. The headphone amplifier uses the VDDA power rail.
- VDDIO—provides power to the I/O peripherals. The NAND Flash and external SD/MMC cards use VDDIO power.
- VDDMEM—provides power to external memory such as SDRAM and mDDR.

The i.MX233 technically has a fifth power rail with the 4P2 LinReg output, but its purpose is to power the DC-DC and it is not used elsewhere in the device.

1.4 Battery Charger

The integrated battery charger can fully charge a Li-Ion battery when 5 V is present. The PMU should be configured to use the 4P2 source for the DC-DC while charging the battery to remove the battery's load. This allows the battery to reach its fully charged state while still operating the device normally.

Current sensors provide firmware a way to monitor the current into the battery. When used with the Low-Resolution ADC (LRADC) reading of battery voltage, the firmware can determine how much charge is in the battery.

1.5 Hardware Requirements

The i.MX233 is a highly-integrated device, but it still requires a few external components. A reference design is available that shows the external hardware components described in the following sections. The reference design, IMX23_EVK_SCH, is available for download from the Freescale web site listed on the back page of this document.

1.5.1 Inductor

A single inductor is needed for the DC-DC switching converter. The size of the inductor depends on the application and product specifications. Refer to the reference design for more details.

1.5.2 Capacitor

A few capacitors are required with the PMU. Each of the power rails requires a capacitor of at least 33 μF for decoupling in addition to capacitors of 0.01 μF , 0.1 μF , and 1 μF . Additionally, the 4P2 rail, battery, and VDD5V rail require capacitors ranging from 0.01 μF to 33 μF . Refer to the reference design for exact capacitor locations and sizes.

1.5.3 Li-Ion Battery

The i.MX233 can operate without a battery if 5 V is present, but only for permanent power situations. Otherwise, the device needs a Li-Ion battery that provides a voltage of 3.0–4.2 V.

1.5.4 External Power Supply

For applications that require more than 1.5 W, an external DC-DC is needed to offload some of the power from the internal DC-DC. Refer to the reference design for more details.

2 Getting Started with the Power Subsystem

This section provides lower level details about the power sources and power supplies.

2.1 Power Sources

There are two external power sources used by the i.MX233. The device can be powered with either the 5 V or the battery power source and does not require both simultaneously to operate. The PMU uses the 5 V source to power the internal 4P2 power source to generate the 4P2 rail which is used as an input to the DC-DC.

2.1.1 Battery Power Source

A Li-Ion battery provides power to the i.MX233. The battery voltage should be between 3.0 and 4.2 V to provide reliable power to the device. The DC-DC converter starts when battery power is detected and the PSWITCH button is pressed. The rails are raised to their default values and ROM begins executing. A battery voltage monitor managed by firmware should alert the system when the voltage reaches a critically low level to prevent brownouts and unexpected device shutdown.

2.1.2 5 V Power Source

The 5 V power source can come from a USB connection or wall-powered source. A wall-powered source must convert the AC wall-power to a 5 V DC power source before powering the i.MX233. The i.MX233 can be programmed to distinguish between 5 V from USB or wall-power by sensing the status of the D+/D- USB signals.

When 5 V is first detected, the internal LinRegs power on the i.MX233. The power rails are raised to their default values and ROM begins processing. After the i.MX233 has booted, the device should enable the 4.2 V LinReg to provide power to the DC-DC converter instead of using the internal LinRegs to power the rails.

It is important to setup the 5 V removal detection to alert the system to switch to battery power. Hardware can be configured to automatically switch to battery power when 5 V is lost.

2.1.2.1 5 V Detection Methods

There are two 5 V detection methods available on the i.MX233. Each method detects 5 V differently. Additionally, the 5 V detection method must be configured once for hardware (DC-DC control logic) and once for software. The hardware configuration for DC-DC control logic is set by software through register settings.

VBUSVALID Detection Method

The VBUSVALID method is the most accurate 5 V detection method on the i.MX233. Freescale recommends using this method during normal operation. The VBUSVALID method compares the VDD5V voltage level with an internal bias voltage. The threshold for detection is configurable in software. Additionally, the VBUSVALID method can be configured to generate an interrupt when a 5 V removal or insertion is detected.

VDD5V_GT_VDDIO Detection Method

The VDD5V_GT_VDDIO detection method is the initial detection method on the i.MX233. It is used at device start-up to allow the DC-DC to determine the 5 V status without enabling extra comparators. After software gains control, the VBUSVALID method should be initialized and used as the 5 V detection method.

The VDD5V_GT_VDDIO method compares VDD5V with VDDIO plus an offset. The offset is approximately 600 mV. This detection method is not robust for detecting 5 V removal because the VDDIO LinReg is powered from the VDD5V rail. When there is a heavy load on VDDIO, the VDDIO voltage may drop at the same rate as VDD5V delaying the detection of 5 V removal. This method works well for device

start-up, but is not recommended during normal operation. Freescale recommends using the VBUSVALID 5 V detection method for applications requiring accurate 5 V removal detection.

2.1.2.2 Selecting the DC-DC Control Logic 5 V Detection Method

The DC-DC control logic requires 5 V status to operate properly. By default, the DC-DC control logic uses the VDD5V_GT_VDDIO detection method. This setting needs to be changed to VBUSVALID during device initialization for proper operation as described in the following steps. Typically, the 5 V detection threshold is set to 4.0 V, but the threshold is configurable and can be changed based on application needs.

1. Turn on the VBUS comparators.
HW_POWER_5VCTRL[PWRUP_VBUS_CMPS] = 1
2. Set the VBUS threshold to 4.0 V (register setting 0x1).
HW_POWER_5VCTRL[VBUSVALID_TRSH] = 0x1
3. Change the DC-DC control logic 5 V detection method to VBUSVALID.
HW_POWER_5VCTRL[VBUSVALID_5VDETECT] = 1

2.1.2.3 Enabling 5 V Detection Interrupts

The i.MX233 can generate interrupts when a 5 V insertion or removal is detected. The interrupt occurs when the threshold is crossed with the correct polarity. The 5 V interrupt has the standard IRQ status and enable bits, and it also has a polarity bit to trigger on a rising or falling 5 V voltage. The 5 V event triggers an interrupt in the ICOLL block which is then handled by software.

These steps describe how to configure the VBUSVALID detection method to generate an interrupt when 5 V removal is detected.

1. Set threshold voltage for VBUSVALID detection method to 4.0 V (0x1).
HW_POWER_5VCTRL[VBUSVALID_TRSH] = 0x1
2. Clear the 5 V interrupt status bit by writing 1 to the SCT clear address space for IRQ status bit.
HW_POWER_CTRL_CLR[VBUSVALID_IRQ] = 1
3. Set the polarity field to detect 5 V removal.
HW_POWER_CTRL[POLARITY_VBUSVALID] = 0
4. Initialize the interrupt handler in the ICOLL block. For more information, refer to the Interrupt Collector (ICOLL) chapter of the *i.MX23 Reference Manual (IMX23RM)*.
5. Enable the VBUSVALID interrupt.
HW_POWER_CTRL[ENIRQ_VBUS_VALID]

2.1.3 4P2 Power Source

The 4.2 V power source internally generates the 4P2 rail from a 5 V power source. The 4.2 V output is used as an input to the DC-DC converter. It should be used when 5 V is present to achieve the best possible power efficiency. [Figure 2](#) shows the logical diagram of how 5 V power is converted to the three power rails.

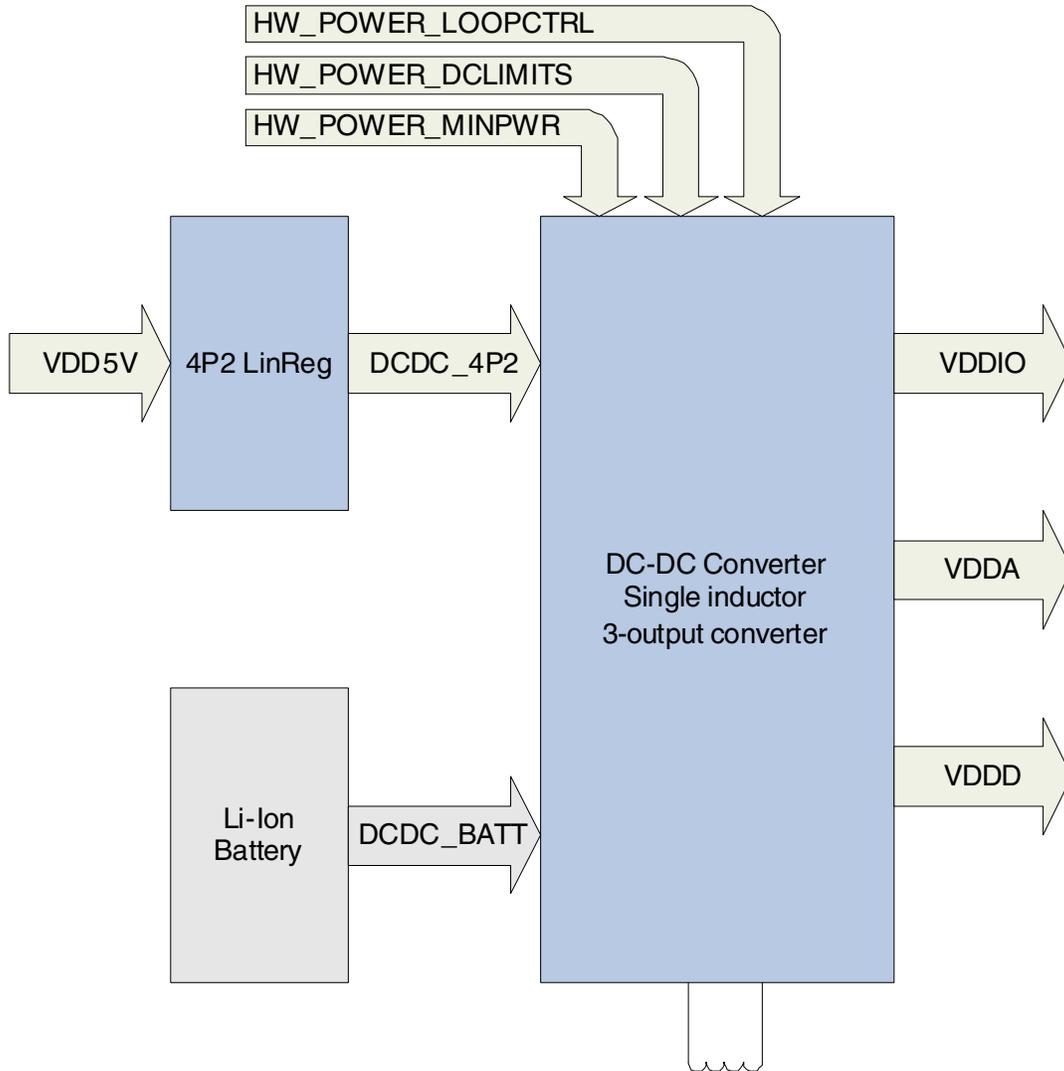


Figure 2. Converting 5 V Power to VDDD, VDDA, and VDDIO Power Rails

The following steps are required to enable the 4P2 power source and use it as an input to the DC-DC:

1. Enable the 4P2 LinReg.
2. Charge capacitance to prevent large current spikes on the 5 V source.
3. Configure the DC-DC to accept the 4P2 LinReg output as an input.

During these steps, the power rails are receiving power from their LinRegs and the DC-DC is not active.

2.1.3.1 Enabling the 4P2 LinReg

The 4P2 LinReg is initialized in preparation for activation using the following steps:

1. Set the 4P2 target to 4.2 V.
2. Enable the 4P2 circuitry to control the LinReg.
 $HW_POWER_DCDC4P2[ENABLE_4P2] = 1$

3. The 4P2 LinReg needs a static load to operate correctly. Since the DC-DC is not yet loading the LinReg, another load must be used.

`HW_POWER_CHARGE[ENABLE_LOAD] = 1`

4. Provide an initial current limit for the 4P2 LinReg. Use the smallest value possible.

`HW_POWER_5VCTRL[CHARGE_4P2_ILIMIT] = 1`

5. Power on the 4P2 LinReg.

`HW_POWER_5VCTRL[PWD_CHARGE_4P2] = 1`

6. Ungate the path from 4P2 LinReg to DC-DC.

`HW_POWER_DCDC4P2[ENABLE_DCDC] = 1`

This `ENABLE_DCDC` bitfield is different from the `ENABLE_DCDC` bitfield located in the `HW_POWER_5VCTRL` register.

2.1.3.2 Charging the 4P2 Capacitance

There is a capacitor on the 4P2 output and parasitic capacitance that must be charged before attempting to fully power on the 4P2 LinReg. Failure to slowly ramp up the current results in brownouts on the 5 V source. To charge the 4P2 capacitance, slowly increment the current limit for the 4P2 rail.

To increment the current limit by the smallest resolution:

1. `HW_POWER_5VCTRL[CHARGE_4P2_ILIMIT] += 1`
2. Wait at least 10 ms, then repeat step 1 until the current limit has reached its maximum of 780 mA.

When the current limit is at 780 mA, the voltage on the VDD4P2 pin should read 4.2 V. The LinReg is fully powered on at this point.

2.1.3.3 Enabling 4P2 Input to DC-DC

The last step is to configure the 4P2 LinReg as an input to the DC-DC so it can supply power to the rails. The DC-DC can be configured to choose which power source to use based on voltage comparisons, and to determine how to arbitrate the 5 V current with the battery charger. These steps are outlined below. After configuring the input, the automatic DC-DC activation must be disabled since it should not be turned on when already on. Then the DC-DC is enabled and begins to power the rails.

Configure the comparator trip point for the DC-DC control block to select the 4P2 LinReg input. This is accomplished by setting the trip point to compare the 4P2 voltage with 85% of the battery. The larger of the two determine whether the 4P2 LinReg or the battery is used as the source. Even though 85% of the battery voltage is compared, 100% of the battery voltage is used if it is selected as the source.

The battery charger and 4P2 LinReg share the 5 V power source. Since the 4P2 rail powers a device, it is given priority over the 5 V current should a conflict arise. The dropout control logic specifies the amount of output voltage drop to tolerate before stealing 5 V current from the charger. Steps to configure the allowable voltage drop are given below.

1. Adjust the comparison between the battery and 4P2 LinReg to 85%.

`HW_POWER_DCDC4P2[CMPTTRIP] = 0x0`

2. Configure the DC-DC control logic to select the greater of the 4P2 LinReg voltage and 85% of the battery and set the dropout voltage threshold to 200 mV before stealing current from the battery charger. Since both of these contained in one bitfield, write both settings to the bitfield simultaneously.

200 mV threshold = 0b11XX

DC-DC selects higher of 4P2 or battery = 0bXX1X

Together the setting should be 0b111X or 0b1110 for simplicity.

HW_POWER_DCDC4P2[DROPOUT_CTRL] = 0b1110

3. Disable the automatic DC-DC startup when 5 V is lost.

HW_POWER_5VCTRL[DCDC_XFER] = 0

4. Enable the DC-DC to begin using it. To do this, follow steps for enabling the DC-DC while 5 V is present.

HW_POWER_5VCTRL[ENABLE_DCDC] = 1

2.2 Power Supplies

The i.MX233 has two types of power supplies. The first is an integrated single-inductor three-output DC-DC switching converter. It is the most power efficient supply on the device. The second is a collection of LinRegs that output to specific power rails. The LinRegs are reliable and stable supplies, but they are very inefficient.

NOTE

The 4P2 LinReg is not discussed in the power supply section since its enable logic and set-up is more complex than the other supplies. It is discussed in [Section 2.1.3.1, “Enabling the 4P2 LinReg.”](#)

2.2.1 LinReg Power Supply

The linear regulators are daisy-chained. The VDDIO LinReg only receives power from the 5LRADCV source, but the VDDA and VDDD receive power from another power rail. The VDDMEM receives power from VDDIO. [Figure 3](#) provides a visual representation of the daisy-chained architecture. Since the DC-DC powers VDDIO and VDDA, and since the VDDIO rail powers VDDMEM and VDDA and since the VDDA rail powers VDDD, the DC-DC can be the power source for LinRegs.

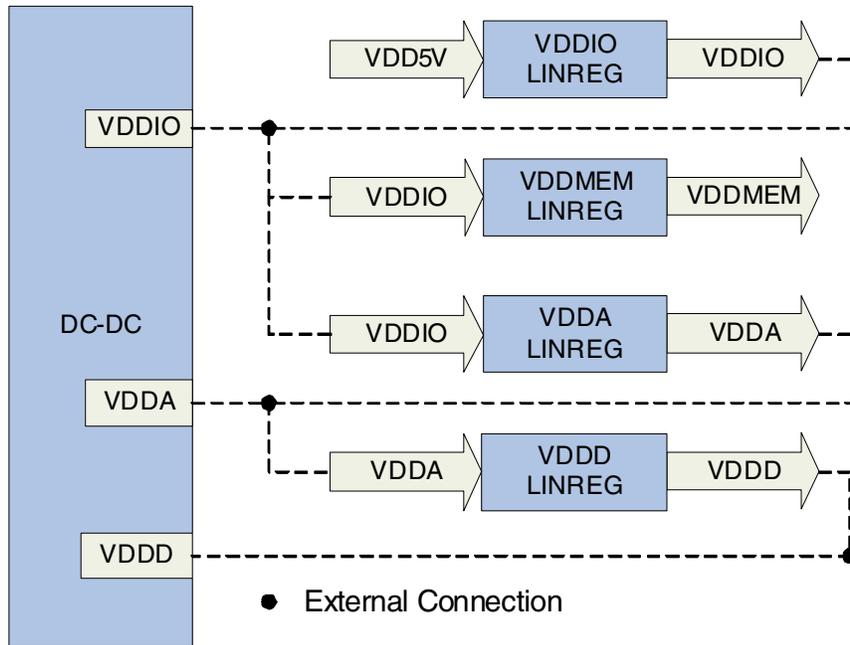


Figure 3. Linear Regulator Daisy-Chain Architecture

2.2.1.1 LinReg Supply Enable Logic

The VDDD, VDDA, VDDIO and VDDMEM rails can be powered from LinRegs. For VDDIO, the LinReg can only be enabled when 5 V is present. Additionally, the 5 V current limiter must be disabled for VDDIO to operate. For VDDD and VDDA, the rail’s LinReg is enabled with particular combinations of 5 V status, ENABLE_DCDC status and the rail’s LinReg enable bit. VDDMEM only requires its enable bit to be set. Figure 4 shows each LinReg power source and enable logic.

The logic to enable the VDDIO and VDDMEM LinRegs is straight forward. With these two rails, there is only one combination of bits that enable the LinReg. For the VDDD and VDDA power rails, there are three different combinations that can enable the LinReg output. Two combinations are applicable when 5 V is present and the other is applicable only when 5 V is not present.

The first combination is the default configuration when the device is powered on by 5 V. In this case, indicated by AND-gate 1 in Figure 4, ENABLE_DCDC is false and 5 V is present. This means the presence of 5 V can activate or deactivate the rail’s LinReg without any software interaction.

In the second case, AND-gate 2 in Figure 4, the LinReg is activated when 5 V is present, ENABLE_DCDC is true, and ENABLE_LINREG is true. The DC-DC is active because ENABLE_DCDC is true. When 5 V is present and ENABLE_DCDC is true, the ENABLE_LINREG bit for the rail must be set to activate the LinReg.

The third case occurs when 5 V is not present and is indicated by AND-gate 3 in Figure 4. In this scenario, the VDDD and VDDA LinReg are enabled by simply setting ENABLE_LINREG.

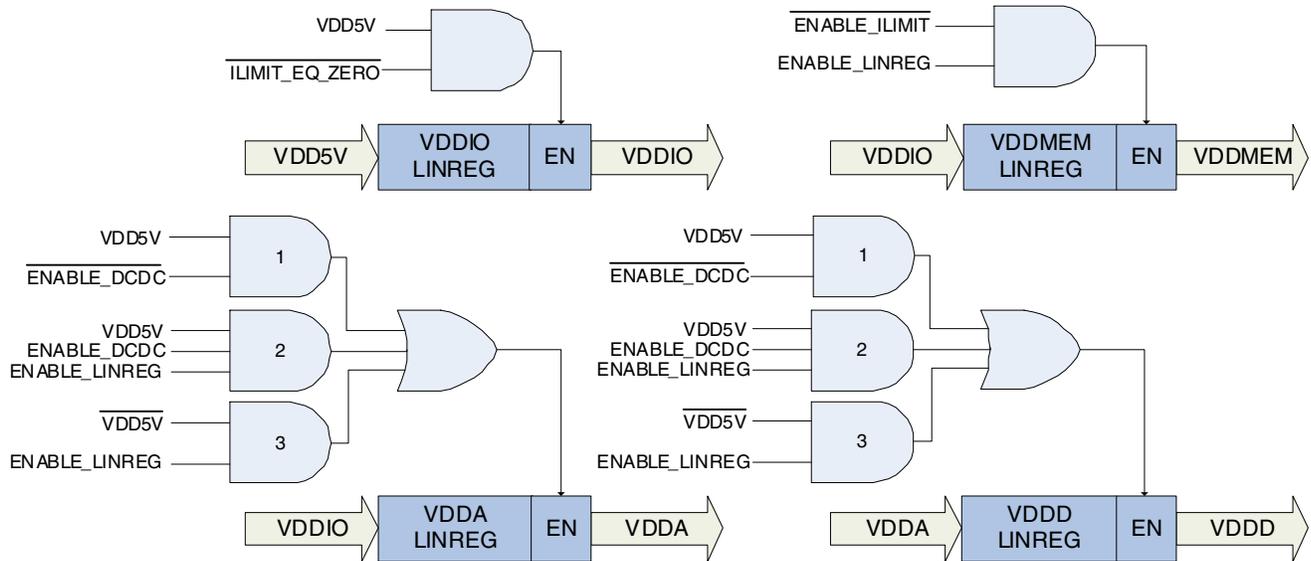


Figure 4. Enable Logic for Linear Regulators

2.2.1.2 Enabling the LinReg Supply

To enable the LinReg supply:

1. Determine the logic that enables the LinReg. See [Figure 4](#) for each LinReg's requirement. 5 V is present in this example.
2. `HW_POWER_VDDIOCTRL[ILIMIT_EQ_ZERO] = 0`
3. `HW_POWER_VDDIOCTRL[ENABLE_LINREG_ILIMIT] = 0`
4. `HW_POWER_VDDACTRL[ENABLE_LINREG] = 1`
5. `HW_POWER_VDDDCTRL[ENABLE_LINREG] = 1`
6. `HW_POWER_VDDDMEM[ENABLE_LINREG] = 1`

2.2.1.3 LinReg Voltage Offset

A power rail's LinReg target voltage is a configurable offset from the DC-DC target voltage. The offset can be 25 mV below, equal to, or 25 mV above the DC-DC target voltage. Freescale recommends always using the 25 mV below setting to prevent LinReg and DC-DC contention. Refer to [Figure 5](#) for a visual description of the LinReg offset logic.

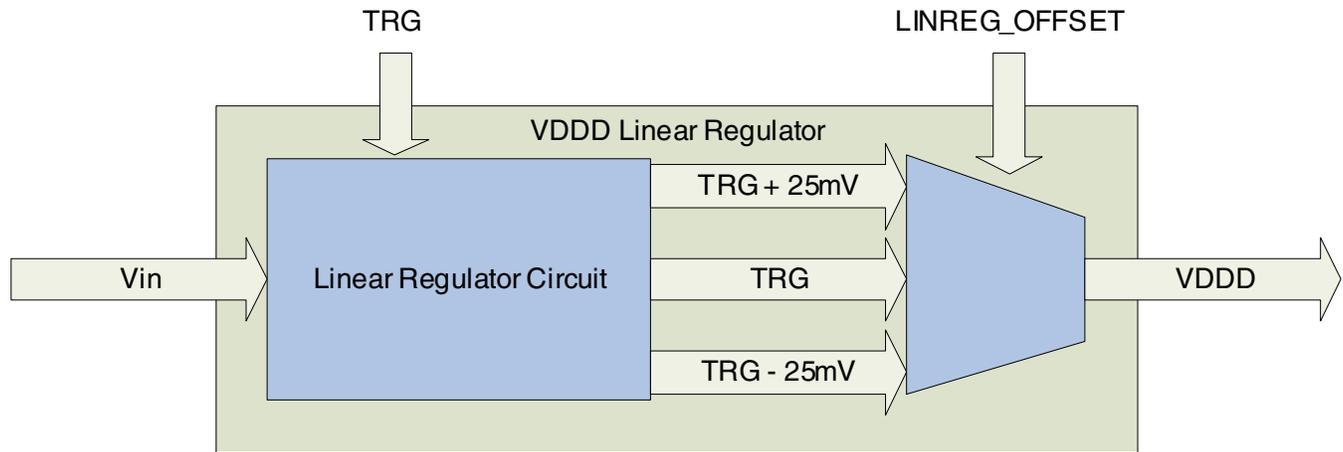


Figure 5. Logical Diagram of VDDD LinReg Offset

2.2.1.4 Setting the LinReg Voltage Offset

To set the LinReg voltage offset to 25 mV below the DC-DC target voltage:

1. `HW_POWER_VDDIOCTRL[LINREG_OFFSET] = 0x2`
2. `HW_POWER_VDDACTRL[LINREG_OFFSET] = 0x2`
3. `HW_POWER_VDDDCTRL[LINREG_OFFSET] = 0x2`

2.2.1.5 5 V Current and VDDIO LinReg

Software can control the amount of current that passes through the VDDIO LinReg. At device boot-up, a large amount of current may be drawn from 5 V to charge the capacitance on the device. This may violate USB specifications for inrush current. As previously shown in [Figure 3](#), the VDDIO receives power from 5 V, then provides power to the other LinRegs. Limiting the current through VDDIO limits the current through all the LinRegs.

[Figure 6](#) shows the current limiting logic in the VDDIO linear regulator. The USB specification for inrush current limits current to 100 mA + 50 μ C. The `HW_POWER_5VCTRL[ENABLE_LINREG_ILIMIT]` bit limits VDDIO current to 100 mA. It is enabled at device boot up, but clears itself before ROM begin execution.

Additionally, the `HW_POWER_5VCTRL[ILIMIT_EQ_ZERO]` bit minimizes the 5 V current drawn by the device through the LinRegs.

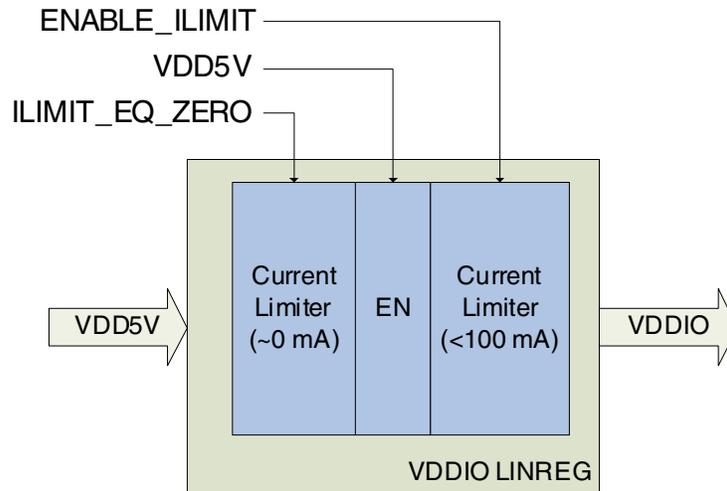


Figure 6. 5 V Current Limiting Logic in VDDIO Linear Regulator

2.2.1.6 Setting VDDIO LinReg 5 V Current Limits

The following steps show the options for setting the VDDIO LinReg 5 V current limits:

1. Limit current to 0 mA.

```
HW_POWER_5VCTRL[ILIMIT_EQ_ZERO] = 1
```

```
HW_POWER_5VCTRL[ENABLE_LINREG_ILIMIT] = X
```

2. Limit current to 100 mA.

```
HW_POWER_5VCTRL[ILIMIT_EQ_ZERO] = 0
```

```
HW_POWER_5VCTRL[ENABLE_LINREG_ILIMIT] = 1
```

3. Remove current limit for VDDIO LinReg.

```
HW_POWER_5VCTRL[ILIMIT_EQ_ZERO] = 0
```

```
HW_POWER_5VCTRL[ENABLE_LINREG_ILIMIT] = 0
```

2.2.2 DC-DC Power Supply

The single-inductor three-output DC-DC switching converter uses a 3.0–4.2 V input to output three rails at 3.3 V, 1.8 V, and 1.2 V. This is the most efficient power supply available on the i.MX233. An application can achieve maximum power efficiency by using the DC-DC in 5 V and battery power source configurations. The DC-DC has additional features that allow it to reduce power even further when the overall system power requirement is low. [Figure 7](#) is a simple block diagram of the DC-DC input source and output rails.

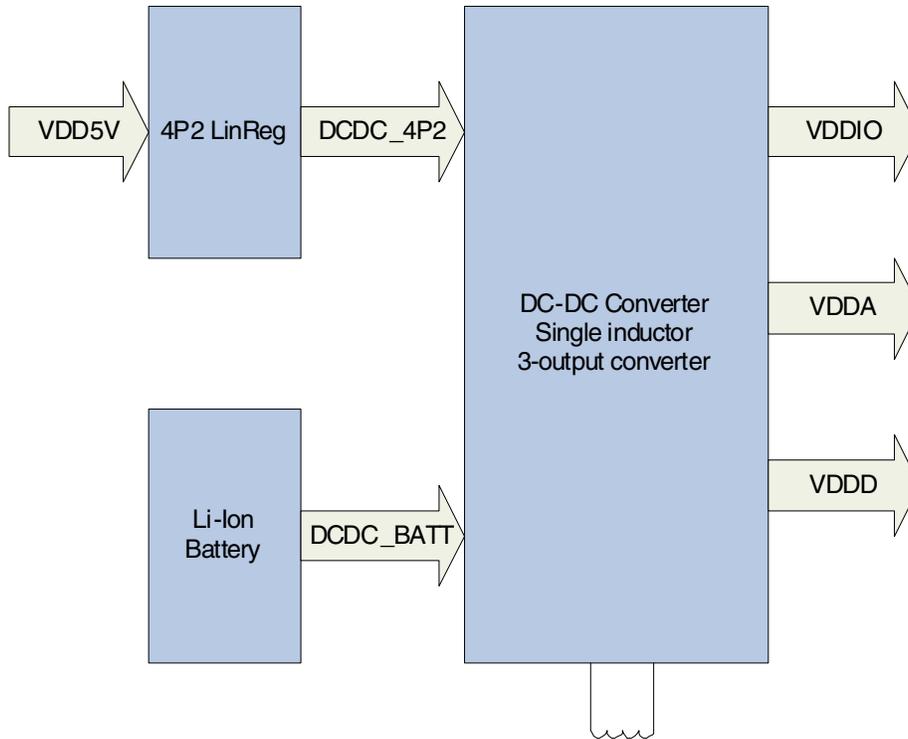


Figure 7. Block Diagram of DC-DC Inputs and Outputs

2.2.2.1 Automatic Battery Voltage Input

The DC-DC control logic requires the current battery voltage as input data. The HW_POWER_BATTMONITOR[BATT_VAL] field must contain an accurate battery voltage so the feedback circuit can operate properly. The field can be updated automatically by using a special channel in the LRADC block.

Channel 7 of the LRADC is dedicated to monitoring the battery voltage. It has been configured to accept a voltage in the range of a Li-Ion battery, and it also has the ability to automatically update the HW_POWER_BATTMONITOR[BATT_VAL] field. Figure 8 shows a simple block diagram of the LRADC channel 7 interacting with the DC-DC. The channel must be initialized and configured by software. The automatic reading and conversion requires an LRADC delay channel. A delay channel can be used to restart the conversion after a period of time. For more information, refer to the LRADC chapter of the *i.MX23 Reference Manual (IMX23RM)*. After set-up is complete, no further modifications are needed to operate normally.

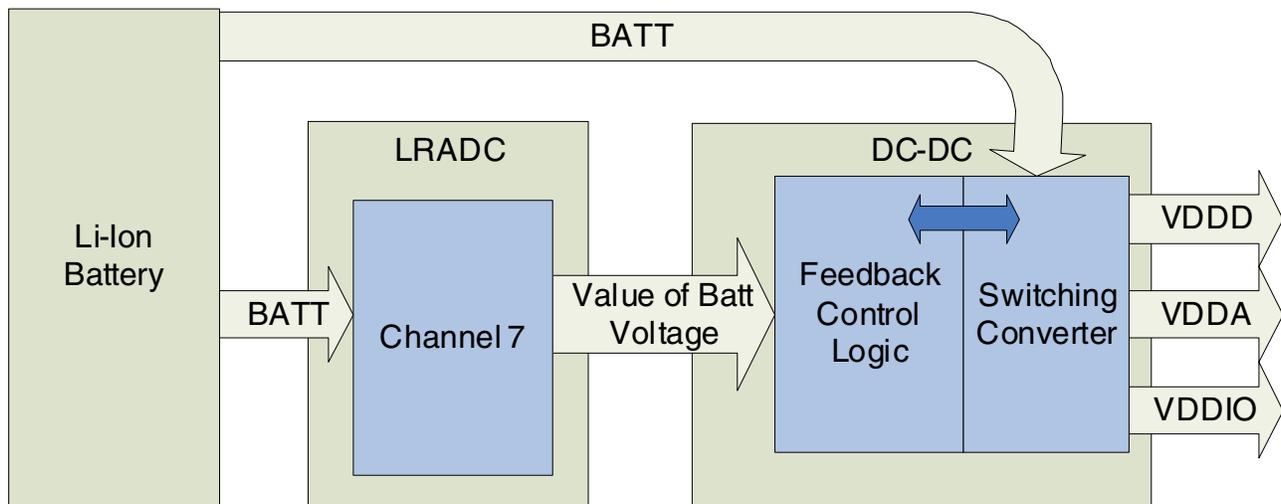


Figure 8. Block Diagram of the Automatic Battery Update

2.2.2.2 Enabling the Automatic Battery Voltage Input

To enable the automatic battery voltage input:

1. Select the Li-Ion conversion factor and configure the automatic update to the power register.
 $\text{HW_LRADC_CONVERSION[SCALE_FACTOR]} = 0x2$
 $\text{HW_LRADC_CONVERSION[AUTOMATIC]} = 0x1$
2. Configure LRADC channel 7 to use a non-pre-divided reading. Bit 7 of the DIVIDE_BY_TWO fields corresponds to channel 7.
 $\text{HW_LRADC_CTRL2[DIVIDE_BY_TWO]} |= 0x80$
3. Configure channel 7 to sample once per conversion. Disable accumulation and set the number of samples to sum to zero.
 $\text{HW_LRADC_CH7[ACCUMULATE]} = 0$
 $\text{HW_LRADC_CH7[NUM_SAMPLES]} = 0$
4. Schedule the channel 7 conversion. Bit 7 of the SCHEDULE field corresponds to channel 7.
 $\text{HW_LRADC_CTRL0[SCHEDULE]} |= 0x80$
5. Configure delay channel 3 to force the LRADC channel 7 to take a reading and perform a conversion every 50 ms.
 $\text{HW_LRADC_DELAY3[TRIGGER_LRADCS]} = 0x80$
 $\text{HW_LRADC_DELAY3[TRIGGER_DELAYS]} = 0x8$
 $\text{HW_LRADC_DELAY3[LOOP_COUNT]} = 0$
 $\text{HW_LRADC_DELAY3[DELAY]} = 0x64$
 $\text{HW_LRADC_DELAY3[KICK]} = 1$
6. Enable the battery voltage as an input to the DC-DC switching logic.
 $\text{HW_POWER_BATMONITOR[EN_BATADJ]} = 1$

7. The battery voltage measured in 8 mV resolution should now appear in the HW_POWER_BATTMONITOR[BATT_VAL] field.

2.2.2.3 5 V Detection for DC-DC Control Logic

The DC-DC control logic requires the status of 5 V to determine when to activate and deactivate the DC-DC output. Two methods of 5 V detection are available to the DC-DC: VDD5V_GT_VDDIO and VBUSVALID. Of the two options, VBUSVALID is the recommended configuration.

NOTE

The 5 V detection for DC-DC control logic is separate from the 5 V detection used by software. The 5 V detection for the DC-DC is used internally by the control block. Software must setup 5 V detection for software and interrupt use.

2.2.2.4 Enabling 5 V Detection for DC-DC Control Logic

To enable 5 V detection for DC-DC control logic:

1. Turn on the VBUS comparators.
HW_POWER_5VCTRL[PWRUP_VBUS_CMPS] = 1
2. Select the voltage threshold for 5 V detection. A voltage reading above the threshold is considered as 5 V present.
HW_POWER_5VCTRL[VBUSVALID_TRSH] = 0x1
3. Configure the DC-DC to use the VBUSVALID 5 V detection method.
HW_POWER_5VCTRL[VBUSVALID_5VDETECT] = 1

2.2.3 DC-DC with 5 V Present

The DC-DC is by default disabled when 5 V is present, but can be enabled using the ENABLE_DCDC bit. The default configuration is intended to prevent LinReg and DC-DC contention. For best power efficiency, the DC-DC must be powered from the 4P2 rail when 5 V is present so special care must be taken to prevent LinReg and DC-DC contention. Contention is possible if the LINREG_OFFSET fields are not configured correctly.

2.2.3.1 DC-DC and LinReg Contention

Contention between the DC-DC and LinReg can occur when the both power sources are activated and simultaneously outputting to a rail. If the target output voltage is not configured correctly, both sources attempt to regulate the target to different voltages. It is possible to configure the DC-DC and LinReg to have mutually exclusive voltage goals so extra attention is needed when enabling both simultaneously.

Cause of Contention

The main cause of contention is the LinRegs are enabled when 5 V is present unless they are explicitly disabled by software. The nature of a LinReg is to pull its output up to reach its target while the nature of

a DC-DC is to pull its output up or down to reach its target. When the LinReg target is higher than the DC-DC target, the LinReg attempts to pull the rail up while the DC-DC pulls the rail down.

Effects of Contention

When contention occurs, the DC-DC can pull a high amount of current which can cause dips on the power source. It can also cause 5 V current to exceed USB specifications. Additionally, the high current can cause the device to heat up which raises the temperature within the device.

Contention Prevention

Preventing contention is the policy for overcoming it. First, the DC-DC and LinReg targets should be configured so both can be satisfied. This means the LinReg target output needs to be less than the DC-DC target output. Since the LinReg is satisfied when the output is at least the target voltage, the DC-DC output should be configured so it is higher than the LinReg output.

Hardware allows this configuration through the LINREG_OFFSET bit. For each of the three main power rails, VDDD, VDDA, and VDDIO, set the LINREG_OFFSET to 25 mV below the target.

2.2.3.2 Enabling DC-DC with 5 V Present

To prevent contention between the DC-DC and LinReg, set the LINREG_OFFSET to 25 mV below the target for each power rail.

```
HW_POWER_VDDDCTRL[LINREG_OFFSET] = 0x2
```

```
HW_POWER_VDDACTRL[LINREG_OFFSET] = 0x2
```

```
HW_POWER_VDDIOCTRL[LINREG_OFFSET] = 0x2
```

The DC-DC is off by default when 5 V is present. Set the ENABLE_DCDC bit to override the default setting.

```
HW_POWER_5VCTRL[ENABLE_DCDC] = 1
```

2.3 Power Rails

The power rails provide power to separate blocks of the device that require different voltages. The power rails are generated by either of the power supplies, DC-DC or LinReg. Each of the power rails voltage is determined by the target field and the LinReg offset in the rail control register. When the rail is powered by the DC-DC, the target field solely determines the output voltage. For LinReg powered rails, the target field plus the LinReg offset determines the voltage. Additionally, each rail has a brownout voltage threshold that is below the target. A system brownout occurs when the output voltage drops to the established threshold and typically indicates the system load exceeds the available power. The i.MX233 can be configured to generate an interrupt when a brownout occurs. The power rails are described in [Table 2](#).

Table 2. i.MX233 Power Rails

Name	Recommended Operating Range (V)	Power Supply	Description
VDDD	1.0 - 1.55	1) Direct from DC-DC 2) LinReg from VDDA	Provides power to digital core such as CPU and clocks
VDDA	1.62 - 2.1	1) Direct from DC-DC 2) LinReg from VDDIO	Provides power to the analog core and audio components
VDDIO	2.9 - 3.63	1) Direct from DC-DC 2) LinReg from VDD5V	Provides power to non-external memory I/O peripherals such as NAND Flash
VDDMEM	1.6 - 3.63	1) LinReg from VDDIO	Provides power to external memory

2.3.1 Target Voltages

The target voltage setting determines the output voltage. The setting specifies the number of voltage steps above a base voltage. The VDDD, VDDA, and VDDIO rails use 25 mV steps and the VDDMEM rail uses 50 mV steps. Each power rail has a different base voltage. The equation to calculate the output voltage is:

$$\text{Voltage} = \text{Base Voltage} + (\text{Step Size} \times \text{Number of Steps})$$

NOTE

For easier math calculations, voltages are written in millivolts.

2.3.1.1 Setting VDDD, VDDA, or VDDIO Target Voltages

Setting the target voltage is similar for the VDDD, VDDA, and VDDIO rails as shown in the following steps. All three use 25 mV steps, but have different base voltages. For this example, VDDD is set to 1.45 V.

1. Calculate the number of steps needed to reach 1450 mV.

$$\text{Voltage} = \text{Base Voltage} + (\text{Step Size} \times \text{Number of Steps})$$

$$1450 = 800 + (25 \times \text{Number of Steps})$$

$$\text{Number of Steps} = 26$$

2. Write the target voltage setting to the control register.

$$\text{HW_POWER_VDDDCTRL}[\text{TRG}] = 26$$

2.3.1.2 Setting VDDMEM Target Voltage

The VDDMEM rail is used to power external memories so it uses 50 mV steps to give it a large range of output voltages. The rail is not enabled by default so it must be enabled prior to use. The following steps provide instructions for setting the VDDMEM rail to 1.8 V and enabling the output. The VDDMEM rail requires an additional step to activate pull-downs to ensure the rail powers up from ground.

1. Calculate the number of steps needed to reach 1800 mV.

$$\text{Voltage} = \text{Base Voltage} + (\text{Step Size} \times \text{Number of Steps})$$

$$1800 = 1700 + (50 \times \text{Number of Steps})$$

$$\text{Number of Steps} = 2$$

2. Write the target voltage setting to the control register.
`HW_POWER_VDDMEMCTRL[TRG] = 2`
3. Activate pull-downs on the external memory power pins to allow the rail to power up from ground.
`HW_POWER_VDDMEMCTRL[PULLDOWN_ACTIVE] = 1`
4. Turn on the VDDMEM LinReg.
`HW_POWER_VDDMEMCTRL[ENABLE_LINREG] = 1`
5. Wait at least 500 μ s to allow the rail to ramp up before proceeding.
6. Disable the current limiter on the LinReg. The limiter is present to meet USB inrush specifications and is active by default.
`HW_POWER_VDDMEMCTRL[ENABLE_ILIMIT] = 0`

2.3.1.3 Setting VDD4P2 Target Voltage

The 4P2 power source converts 5 V power to a voltage level that can be input to the DC-DC. Typically, the voltage level is set once to 4.2 V, but it may be necessary to change this level. There are four pre-defined voltage levels (4.2 V, 4.1 V, 4.0 V, and 3.9 V) and one battery-matching voltage level. The battery-matching voltage level outputs the same voltage currently measured on the battery. The battery-matching voltage level must only be set when there is a battery present with a valid voltage (3.0–4.2 V). To set the VDD4P2 target voltage:

1. Determine which voltage level to output.
4.2 V output translates to 0x0 bitfield setting
2. Write the target voltage setting to the control register.
`HW_POWER_DCDC4P2 [TRG] = 0`

2.3.2 Brownouts

Brownouts occur when the voltage on a rail drops below the target voltage setting. This is typically due to a heavy load on the output or insufficient input power. The i.MX233 PMU uses brownout detectors to allow software or hardware to handle brownouts. Software can be notified with an interrupt of a brownout condition and act accordingly. Hardware can automatically shut down the device when a brownout condition is detected.

2.3.2.1 Master Brownout Shutdown

All hardware power down events are controlled by a master switch. This includes brownouts and device resets, but not the watchdog reset. The master shutdown must be managed by software to allow the device to shutdown as expected. The master shutdown is a single bit, but since it can turn off the device, the register it resides in requires an unlock key to write to the register. This bit, named power-down off, is set to disable the hardware's ability to power down the device.

1. Generate one register mask to enable the master shutdown switch and one to disable the master shutdown switch. The unlock key must be placed in the upper 16-bits of the mask.

$ENABLE_MASK = (0x3E77 \ll 16) | 0x0 = 0x3E770000$

$DISABLE_MASK = (0x3E77 \ll 16) | 0x2 = 0x3E770002$

2. To enable the master shutdown switch, write the `ENABLE_MASK` to the register.

$HW_POWER_RESET = ENABLE_MASK$

3. To disable the master shutdown switch, write the `DISABLE_MASK` to the register.

$HW_POWER_RESET = DISABLE_MASK$

2.3.2.2 Brownout Thresholds

Thresholds determine at what level the comparators detect a brownout. The threshold is set by writing to one of the control registers. Calculating the actual value to be written varies by the type of brownout.

Power Rail Brownout

Brownout thresholds are calculated as an offset, or margin, from the target voltage register field. The actual voltage level of the brownout threshold can be calculated by determining the target output voltage and subtracting the number of brownout voltage margin. The equation is shown below.

$MARGIN = (STEP_SIZE \times NUM_STEPS)$

Follow these steps to set the `VDDD` brownout to 100 mV:

1. Calculate the number of 25 mV steps needed for brownout margin.

$MARGIN = (STEP_SIZE \times NUM_STEPS)$

$100 = (25 \times NUM_STEPS)$

$NUM_STEPS = 4$

2. Write the margin to the brownout offset field in the rail control register.

$HW_POWER_VDDDCTRL[BO_OFFSET] = 4$

Battery Brownout

The battery brownout threshold is not an offset like the power rails, but an absolute voltage level. The available range is 2.4–3.64 V with 40 mV resolution. The battery brownout is triggered when the battery voltage reaches the threshold. Typically, the battery brownout is configured to generate an interrupt that notifies the application that the battery is low. The equation to calculate the battery brownout threshold is similar to the equation to set a power rail target voltage.

$BO_VOLT = BASE_VOLT + (STEP_SIZE \times NUM_STEPS)$

Follows these steps to set the battery brownout level to 3.2 V:

1. Calculate the number of steps needed to adjust the brownout level to 3.2 V (3200 mV).

$BO_VOLT = BASE_VOLT + (STEP_SIZE \times NUM_STEPS)$

$3200 = 2400 + (40 \times NUM_STEPS)$

$NUM_STEPS = 20$

2. Write the brownout setting to the register bitfield.

$HW_POWER_BATMONITOR[BRWNOUT_LVL] = 20$

5 V Source Brownout

The 5 V source brownout is called the VBUS droop brownout in the i.MX233 PMU. The brownout level is a voltage threshold. It uses an absolute voltage similar to the battery brownout. It is not an offset from the target. There are four available settings: 4.3 V, 4.4 V, 4.5 V, and 4.7 V. To prevent chattering, each setting has 50 mV of hysteresis. The VBUS droop brownout can generate an interrupt when 5 V has dropped below the threshold. It is useful when determining if the load on 5 V is excessive.

Follow these steps to set the VBUS droop brownout to 4.3 V:

1. Select threshold and determine register setting.
4.3 V corresponds to 0x0 register setting
2. Write the threshold setting to the register bitfield.
`HW_POWER_5VCTRL[VBUSDROOP_TRSH] = 0x0`

4P2 Brownout

Brownout detection is available on the 4P2 rail. The brownout threshold range is 3.6–4.2 V with 25 mV resolution. A brownout occurs when the 4P2 voltage reaches the brownout level. The 4P2 brownout can be configured to generate an interrupt to notify the application when the 4P2 source is drooping. The 4P2 brownout voltage can be calculated using the same equation as the battery brownout voltage.

Follow these steps to set the 4P2 brownout voltage to 3.8 V:

1. Calculate the number of steps needed to adjust the brownout level to target threshold.
2. $BO_VOLT = BASE_VOLT + (STEP_SIZE \times NUM_STEPS)$

2.3.2.3 Automatic Hardware Shutdown on Brownout

In addition to generating interrupts, brownouts can be configured to automatically power down the i.MX233 through hardware depending on the 5 V status. Three power rails (VDDD, VDDA, and VDDIO), the battery and the 5 V source have this feature. Each of these five brownout detectors has an enable bit to control it locally, and all are controlled by the master enable switch.

When 5 V is present, only the 5 V brownout automatic shutdown is available. The VDDD, VDDA, VDDIO and battery automatic shutdown are disabled. The brownout interrupt is enabled regardless of the 5 V connection status.

Automatic 5 V Brownout Shutdown

The automatic 5 V brownout shutdown uses the VBUSVALID threshold instead of the VBUSDROOP threshold. It should only be enabled when the PMU is not configured to handle a 5 V removal event. Typically, this feature is only enabled during initialization until handoff-to-battery is ready. Also, the automatic hardware shutdown is not active until the VBUS voltage level has exceeded the VBUSVALID threshold. The steps to enable automatic 5 V brownout shutdown are as follows:

1. Configure the VBUSVALID threshold to the desired level.
`HW_POWER_5VCTRL[VBUSVALID_TRSH] = Threshold Level`
2. Enable the local brownout enable for 5 V.

HW_POWER_5VCTRL[PWDN_5VBRNOUT] = 1

3. Enable the master brownout switch. See [Section 2.3.2.1, “Master Brownout Shutdown,”](#) for instructions

Automatic Battery Brownout Shutdown

The automatic battery brownout shutdown uses the battery brownout threshold. It should be enabled when the battery-to-5 V handoff is not ready or when the battery is exceptionally low and power loss is eminent. This automatic shutdown is deactivated when 5 V is present. The battery brownout detection to generate an interrupt is still enabled. The steps to enable automatic battery brownout shutdown are as follows:

1. Set the battery brownout voltage following steps indicated in the [Battery Brownout](#) section.
2. Enable the local brownout shutdown enable for battery.

HW_POWER_BATTMONITOR[BRWNOUT_PWD] = 1

3. Enable the master brownout switch. See [Section 2.3.2.1, “Master Brownout Shutdown,”](#) for instructions.

Automatic Power Rail Brownout Shutdown

The three power rails, VDDD, VDDA, and VDDIO have automatic shutdown functionality. The VDDMEM rail is generated from VDDIO and any brownout on the memory rail shows up on the VDDIO rail. The automatic shutdown should be used during device boot-up before software interrupt handling is enabled or when software handling is too slow. It can also be used in final production builds to shut down a device on a rail brownout immediately. The steps to enable automatic shutdown on the VDDIO rail are as follows:

1. Set the brownout voltage following steps indicated in the [Battery Brownout](#) section.
2. Enable the local brownout shutdown enable for VDDIO.

HW_POWER_VDDIOCTRL[PWDN_BRNOUT] = 1

3. Enable the master brownout switch. See [Section 2.3.2.1, “Master Brownout Shutdown,”](#) for instructions.

2.3.2.4 Software Brownout Interrupts

As stated, the brownout detectors can generate interrupts. This allows software to handle brownout conditions and possibly remedy them. The interrupts are sent through the ICOLL block and should be configured as Fast Interrupt Requests (FIQ). Follow these steps to set the VDDD software brownout interrupt:

1. Set the brownout threshold for VDDD to 100 mV below target. Follow the instructions indicated in the [Power Rail Brownout](#) section for setting power rail brownout thresholds.
2. Clear the VDDD brownout interrupt status bit. Write to the VDDD brownout IRQ status field in the CLR register.

HW_POWER_CTRL_CLR[VDDD_BO_IRQ] = 1

3. Initialize the interrupt handler in the ICOLL block. For more information, refer to the Interrupt Collector (ICOLL) chapter of the *i.MX23 Reference Manual (IMX23RM)*.

4. Enable the VDDD brownout interrupt.
`HW_POWER_CTRL[ENIRQ_VDDD_BO] = 1`

3 Using the Power Subsystem

3.1 Powering the Supplies

The DC-DC and LinReg power supplies receive their power from the 5 V connection or from a Li-Ion battery. The following section describes the steps to properly configure a supply for the current power source. As a general rule for optimizing power, the DC-DC should be used as often as possible to achieve the best power efficiency.

3.1.1 LinRegs Powered by 5 V

Powering the LinRegs from 5 V is the default configuration when a device is powered on from a 5 V connection. This configuration provides a stable power source, but is less efficient than using the 4PV source to power the DC-DC. At initialization, the LinRegs automatically turn on when 5 V is present and the DC-DC is turned off.

When configuring the LinRegs for a 5 V source, follow these steps:

1. Configure the LinReg voltage offset to 25 mV below the target.
2. Set the power rail brownout voltage.
3. Set the power rail target voltage. Be sure to compensate for the LinReg offset to ensure the output is at the voltage level expected.

3.1.2 DC-DC Powered by Battery

DC-DC power from a battery is the default configuration when a device is powered on by pressing the PSWITCH and a battery is present. In this state, the VDDA and VDDIO rails are powered on from DC-DC, but the VDDD rail is powered from its LinReg. During power block initialization in software, switch the VDDD rail to DC-DC power source by disabling the LinReg and enabling the DC-DC output.

Follow these steps to configure the DC-DC supply for a battery source:

1. Enable automatic battery voltage input to the DC-DC by following the instructions in [Section 2.2.2.2, “Enabling the Automatic Battery Voltage Input.”](#)
2. Configure the LinRegs voltage offset to 25 mV below the target.
3. Enable the DC-DC with 5 V present. This prevents the DC-DC from turning off when 5 V is detected. It is important to have the LinReg voltage offset correct before performing this step.
4. Set the power rail brownout voltage.
5. Set the power rail target voltage.

3.1.3 DC-DC Powered by 5 V

When 5 V is present, Freescale recommends powering the DC-DC from the 4P2 rail which is in turn powered by the 5 V source. The DC-DC are the most efficient power supplies in the PMU and allow faster battery charging since more current can go to the battery instead of the device.

To configure the DC-DC for 5 V power, follow these steps:

1. Enable automatic battery voltage input to the DC-DC control logic.
2. Configure the DC-DC control logic 5 V detection to use VBUSVALID.
3. Configure the LinReg voltage offset to 25 mV for all rails.
4. Enable the 4.2 V power source.
5. Enable DC-DC while 5 V is present.
6. Set the supply rail brownout voltage.
7. Set the supply rail output voltage.

3.2 Transition Between Power Sources

Throughout the course of an application's life, the power source may change dynamically. [Figure 9](#) shows simplified power states and transitions between the states.

The Power Off state is an unpowered state while the other three states are powered-on states. The three powered-on states are discussed in [Section 3.1.1, "LinRegs Powered by 5 V,"](#) [Section 3.1.2, "DC-DC Powered by Battery,"](#) and [Section 3.1.3, "DC-DC Powered by 5 V."](#) The two DC-DC powered states are considered the end states for the application. The only way to leave these states is for 5 V to change state (or for the user to power off the device). The transition from Power Off to LinReg powered by 5 V to DC-DC powered by 4P2 is covered in [Section 2.1.3.1, "Enabling the 4P2 LinReg,"](#) as it is a natural transition to this end state. The transitions between the end states are covered in the following sections.

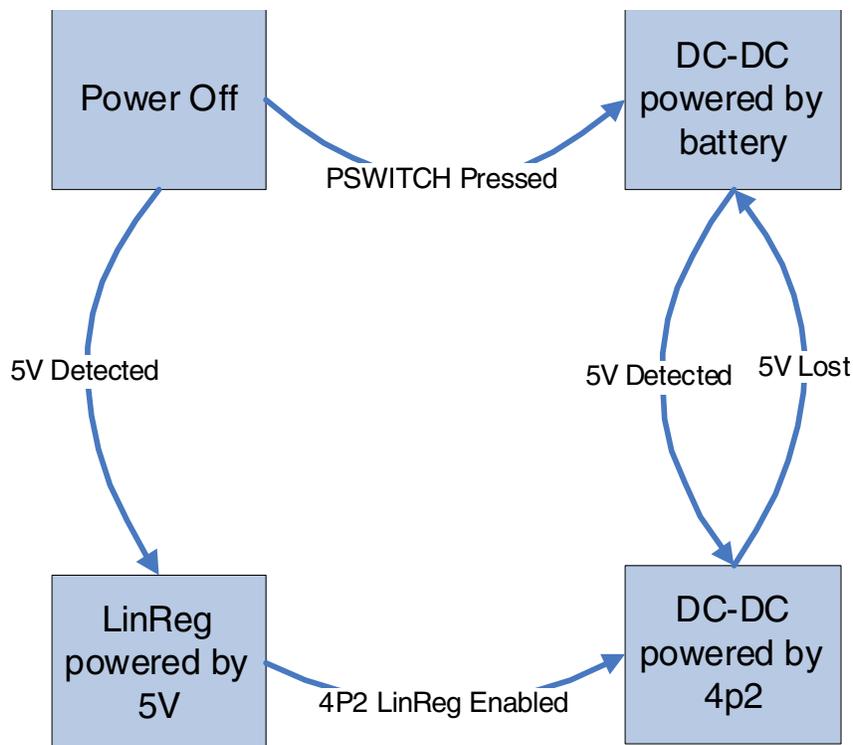


Figure 9. Power Source States and Transitions

3.2.1 Transition from Battery to 5 V Power

The battery to 5 V power transition, or Batt-to-5 V hand-off for short, can take place entirely in software. There is no time limit to performing this handoff because the two power sources are present so software can decide if and when to use the 5 V power.

The steps to transition from battery to 5 V power starting from the DC-DC powered by battery state are as follows:

1. Enable the 5 V detection interrupt to trigger on a 5 V insertion. To trigger on insertion, configure the 5 V interrupt polarity bit for 5 V connected.
2. Wait for the 5 V insertion.
3. When 5 V is detected, enable the 4P2 LinReg and switch the DC-DC to use the 4P2 LinReg as an input.
4. Enable the 5 V detection interrupt to trigger on a 5 V removal to prepare for the next transition when 5 V is disconnected.

3.2.2 Transition from 5 V to Battery Power

The 5 V to battery power transition, or 5 V-to-Batt hand-off, has a hard time limit. When 5 V is lost, there is a short amount of time when the DC-DC must switch to a battery power source before the power rails begin to drop. To help keep power stable, the first half of the transition takes place in hardware. The DC-DC control logic that uses the 4P2 LinReg is configured to use 4P2 if it is greater than 85% of the

battery voltage. When 5 V is lost, the 4P2 rail drops and the battery voltage becomes the input source. At this point, the power rails are stable and the DC-DC has a stable power source, but it is not prepared for the next transition and the application is not aware of the power source transition.

The 5 V detection interrupt triggers when 5 V is disconnected. This notifies the application that 5 V was lost. The application must now configure the 5 V detection interrupt for 5 V connected to prepare for the next transition.

The steps to transition from 5 V to battery power starting in the DC-DC powered by 4P2 state are as follows:

1. Properly configure the DC-DC control logic to select the greater of battery or 4.2 V.
2. Configure the 5 V detection interrupt for 5 V disconnect detection.
3. Wait for removal notification through the disconnect interrupt. When the interrupt is detected, the hardware already has switched from 4P2 power source to the battery.
4. Disable the 4P2 LinReg.
`HW_POWER_DCDC4P2[ENABLE_4P2] = 0`
5. Configure the 5 V detection interrupt for 5 V connect detection.

3.3 Using the Software Brownout Interrupt

The software brownout interrupts are valuable mechanisms that allow the application a way to handle any adverse power events. Each of the brownouts has a different level of severity, so each may be handled in different ways. The following sections describe Freescale's recommended handling of each brownout situation.

3.3.1 Software Battery Brownout Interrupt

The battery brownout interrupt should be used to indicate when the battery has reached a low voltage level. Freescale recommends using at least two thresholds managed by the application. The first threshold should be a low battery voltage that still permits operation, but warns the user of a low battery. Typically, this is set around 3.2–3.4 V. The second threshold should be a voltage level where the device should be powered down immediately. This is typically around 3.0 V.

The common sequence of events for the software battery brownout is as follows:

1. The battery brownout is enabled and the high threshold is used as the brownout voltage level.
2. The battery voltage drops as the device is used.
3. The battery brownout generates an interrupt. The interrupt handler detects the high threshold, re-arms the interrupt and sets the low threshold as the brownout voltage level. The handler also sends a low-battery notification to the application.
4. The battery voltage continues to drop unless a charger is attached.
5. The battery brownout generates an interrupt. The handler detects the low threshold has been reached and shuts down the device.

3.3.2 5 V Detection Interrupt

The 5 V detection interrupt should always be configured to properly detect 5 V connected or 5 V disconnected. In addition to helping to properly configure the power sources, the interrupt can provide information to the battery charger logic to determine when to enable or disable the charger. It is also useful for determining when to begin looking for a USB connection.

3.3.3 Handling Power Rail Brownouts

Power rail brownouts generally indicate a catastrophic event in the PMU. They most often occur when the battery or 5 V rail experiences a brownout, or when there is an excessively heavy load on one of the rails. A power rail brownout is a serious condition in the PMU and should be handled by immediately powering down the device. This can be configured as an automatic hardware power down or handled in the interrupt handler by resetting the device.

4 Getting Started with the Battery Charging Subsystem

The i.MX233 battery charger is a two-stage charger. The first stage is constant current and the second stage is constant voltage. Both stages are shown in [Figure 10](#). In the constant current state, the charge current is set by the application and the battery voltage is below 4.2 V. As the battery receives the current, the battery voltage rises. When it reaches 4.2 V, the battery charger hardware begins lowering the current to the battery while keeping the battery voltage at 4.2 V. This begins the constant voltage stage. The hardware keeps lowering the current until an internal current sensor detects the battery current has reached a pre-determined threshold set by the application. At this time, charging ends.

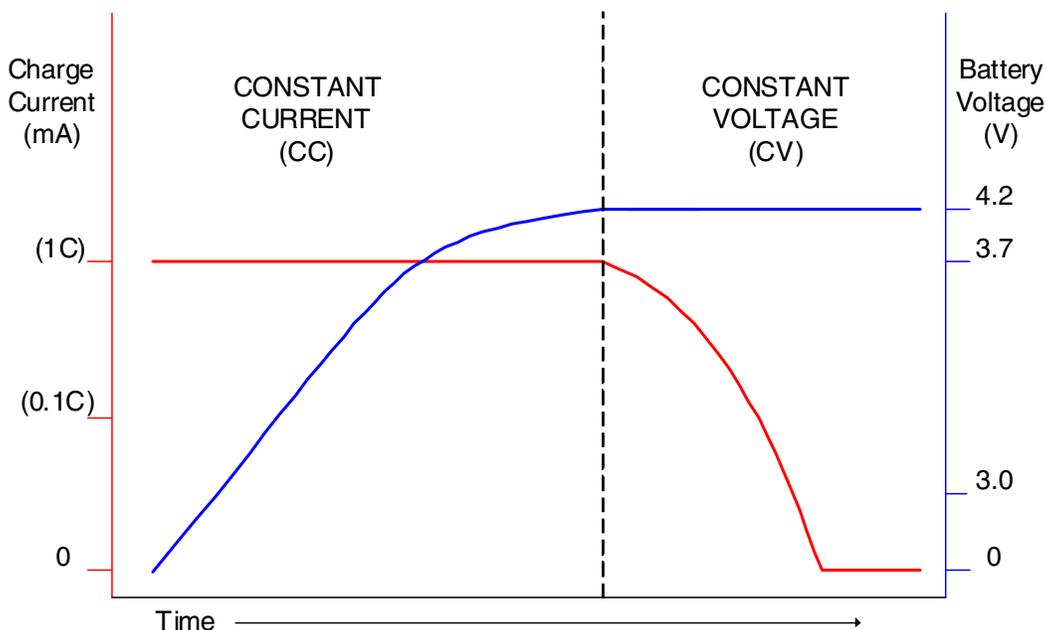


Figure 10. Charger Stages

The battery charger's main purpose is to charge the battery, but it can also be used to detect a battery. Steps for using it in this manner are provided in [Section 4.2, “Using the Battery Charger.”](#)

As previously stated, the 4P2 LinReg and battery charger must share the 5 V current. The 4P2 LinReg is given priority over the 5 V current and is the main controller of the current. [Section 4.2.3, “Managing the Charger and 4P2 Rail,”](#) explains how to manage the 4P2 LinReg and the battery charger. There are master control fields for enabling the charger and for the charger current present in the 4P2 control registers that must be configured before the charger can run as expected.

4.1 Configuring the Charger

The battery charger uses two different current settings. The first is the maximum amount of current the charger should output. This is commonly referred to as the charge current. The second setting is the stop current. This is the threshold set by the application that indicates the battery is full. Both settings should be configured before beginning to charge a battery.

4.1.1 Setting Battery Charge Current

The charge current is represented by six bits where each bit corresponds to some number of milliamps of current. The charge current range is 0–780 mA with 10 or 20 mA resolution depending on the value selected. [Table 3](#) shows the bit to charge current translation.

Table 3. Bit to Charge Current Translation

Bit Number	Charge Current
0	10 mA
1	20 mA
2	50 mA
3	100 mA
4	200 mA
5	400 mA

This example shows how to set the charge current to 650 mA:

1. Translate the 650 mA value to a register setting.
Start by subtracting the largest setting and working down until the current left is zero or bit zero is reached.
 $650 \text{ mA} = 0b110100$
2. Write the register setting to the battery charge current field.
`HW_POWER_CHARGE[BATTCHRG_I] = 0b110100`

4.1.2 Setting Battery Charger Stop Current

The stop charge current is a setting that signals a flag to stop charging. The stop current is represented by four bits where each bit corresponds to some number of milliamp of current. The stop range is 0–180 mA

with 10 or 20 mA resolution depending on the value selected. [Table 4](#) shows the bit to stop current translation.

Table 4. Bit to Stop Current Translation

Bit Number	Stop Current
0	10 mA
1	20 mA
2	50 mA
3	100 mA

This example shows how to set the stop current to 120 mA:

1. Translate the 120 mA value to a register setting.
Start by subtracting the largest setting and working down until the current left is zero or bit zero is reached.
 $120 \text{ mA} = 0b1010$
2. Write the register setting to the battery charge current field.
`HW_POWER_CHARGE[STOP_ILIMIT] = 0b1010`

4.1.3 Enabling the Battery Charger

When the 4P2 LinReg is active, the charger can be enabled by toggling one register field. When the 4P2 LinReg is not enabled, the enabling the battery charger is a two-step process. The two-step process is described in [Section 4.2.3, “Managing the Charger and 4P2 Rail.”](#) To enable the charger in the one-step process, clear the charger power down bit:

```
HW_POWER_CHARGE[PWD_BATTCHRG] = 0
```

4.2 Using the Battery Charger

The battery charger can be used anytime the 5 V source is present. The battery charger shares the 5 V current with the 4P2 rail, and under most situations, this does not causes problems. Current arbitration is only necessary when the 4P2 rail is heavily loaded. Detecting a battery is another useful application for a battery charger. This allows the application to definitively know whether or not the battery can be used as a source. But first and foremost, the charger is used to recharge a battery.

4.2.1 Charging the Battery

Charging a battery is a multistep process. The first steps are to determine and configure the charge current and the stop charge current. Next the charger is initialized to begin the charge process. Finally, the current to the battery is monitored and charging ends when the stop charge current is reached. The steps for charging a battery are as follows:

1. Make sure the 4P2 rail is enabled to maximize the current to the battery.
2. Determine the charge current and configure the charger setting.

The maximum current is determined by the battery manufacturer and can be found in the battery datasheet. For this example, use 600 mA charge current.

`HW_POWER_CHARGE[BATTCHRG_I] = 0x30`

3. Determine the stop charge current and configure the charger setting.

This value is determined by the battery manufacturer. It is typically 10% of the maximum charge current and can also be found in the battery datasheet. For this example, use 60 mA as the stop charge current.

`HW_POWER_CHARGE[STOP_ILIMIT] = 0x5`

4. Enable the battery charger following the steps described in [Section 4.1.3, “Enabling the Battery Charger.”](#)
5. Monitor the charger status flag to determine when the charger has reached the stop current threshold.
6. End charging by disabling the battery charger. This is accomplished by setting the power down charger bit.

`HW_POWER_CHARGE[PWD_CHARGE] = 1`

4.2.2 Using the Charger to Detect a Battery

Detecting a battery is useful for production line situations where the storage media may be soldered before the battery. If the battery is not present and detected as such, the application can change its behavior to ensure the battery is not used as a power source.

It is important to note that the test must be performed before the 4P2 rail capacitance has been charged up. The test relies on detecting the voltage spike that occurs when the charger is turned on with an open circuit. If there is a large amount of capacitance on the battery pin, an alternate test method may need to be developed.

The following steps show how to use the battery charger to detect a battery:

1. Enable the automatic battery voltage update for DC-DC control logic. The main purpose of this is to have an accurate battery voltage measurement.
2. Configure the LinReg for 5 V power source. This is the default configuration when 5 V powers on the device.
3. Read the battery voltage.
4. If the battery voltage is in the normal battery voltage range, the battery is present and the test is complete.
5. If the battery voltage is near zero, the battery is either dead or unconnected. Continue to next step.
6. Set the 4P2 master current to 30 mA and enable the master charge switch.

`HW_POWER_5VCTRL[CHARGE_4P2_ILIMIT] = 0x3`

`HW_POWER_5VCTRL[PWD_CHARGE_4P2] = 0`

7. Configure the battery charger to output 30 mA.
8. Wait about 250 ms to allow the battery pin voltage to rise.
9. Read the battery voltage.

If the battery voltage is near 4.2 V, the battery is not present because there is an open circuit.

If the battery voltage is near 3.0 V, the battery is present and has begun to charge.

4.2.3 Managing the Charger and 4P2 Rail

Because the battery charger and 4P2 rail share the 5 V current, some arbitration may be necessary. The 4P2 rail has priority over the 5 V current so it also has the master current limit and enable switch. The battery charger has its own current limit and enable switch, but both are limited to the master settings. Table 5 shows the master switches that must be enabled to allow the local switches to operate normally.

Table 5. 4P2 Rail Master Switch Dependents

Function	4P2 Rail Master Switches	Battery Charger Local Switch
Current	HW_POWER_5VCTRL[CHARGE_4P2_ILIMIT]	HW_POWER_CHARGE[BATTCHRG_I]
Power	HW_POWER_5VCTRL[PWD_CHARGE_4P2]	HW_POWER_CHARGE[PWD_BATTCHRG]

To manage the current limit and enable switch, make sure the master switches are configured properly before using the battery charger local switches.

5 Optimizing the Power Subsystem

The i.MX233 offers many features designed to minimize power usage. The features mainly involve clock frequency manipulation, but also involve dynamic hardware changes to customize the PMU for performance or power savings.

5.1 Enabling Interrupt Wait

The interrupt wait feature is designed to reduce the power used by the CPU clock. This feature automatically gates the CPU clock when the processor is idle. It requires one assembly instruction added to the application scheduler task.

To enable the feature:

1. Add the following instruction just before the scheduler idles itself. At this point in the scheduler, there should be no tasks ready to execute.

```
mcr p15, 0, r0, c7, c0, 4
```

2. Enable the interrupt wait feature in hardware.

```
HW_CLKCTRL_CPU[INTERRUPT_WAIT] = 1
```

5.2 Enabling HCLK Auto-Slow

HCLK auto-slow is a feature that reduces the HBUS frequency when it is not in use for particular activities. The type of activity or traffic can be configured as well as the slower frequency to use.

The following are the activity types that can be monitored by the HCLK auto-slow logic:

- Data Co-processor (DCP)
- Pixel Pipeline (PXP)

- APBH DMA
- APBX DMA
- Any AHB master
- Less than three AHB masters
- CPU data access
- CPU instruction access

To enable the feature:

1. Select the activity types to monitor.

`HW_CLKCTRL_HBUS[DCP_AS_ENABLE]`

`HW_CLKCTRL_HBUS[PXP_AS_ENABLE]`

`HW_CLKCTRL_HBUS[APBHDMA_AS_ENABLE]`

`HW_CLKCTRL_HBUS[APBXDMA_AS_ENABLE]`

`HW_CLKCTRL_HBUS[TRAFFIC_JAM_AS_ENABLE]`

`HW_CLKCTRL_HBUS[TRAFFIC_AS_ENABLE]`

`HW_CLKCTRL_HBUS[CPU_DATA_AS_ENABLE]`

`HW_CLKCTRL_HBUS[CPU_INSTR_AS_ENABLE]`

2. Select the slow divide mode divider. This value additionally divides the current HCLK by the configured amount. For the example, divide by 2.

`HW_CLKCTRL_HBUS[SLOW_DIV] = 2`

3. Enable the auto-slow mode for HBUS.

`HW_CLKCTRL_HBUS[AUTO_SLOW_MODE] = 1`

5.3 Setting Power Transistor Size

The power transistors can be modified during run-time to optimize the DC-DC for performance or for power savings. There are three settings available:

- Half-size FET
- Normal-size FET
- Double-size FET

The half-size FET setting should be used in low-power modes where increased resistance of the power FETs is acceptable. The setting disables half the power FETs. The double-size FETs should be used in high-power conditions to make the DC-DC more robust to handle the heavier load. The half-size and double-size FET options each have their own enable bit. Even though half-size and double-size can be simultaneously enabled, they should not be since they do not cancel out. The normal-size FET setting is not the same as enabling half-size and double-size simultaneously.

- To enable the half-size FETs:

`HW_POWER_MINPWR[DOUBLE_FETS] = 0`

`HW_POWER_MINPWR[HALF_FETS] = 1`

- To enable the double-size FETs:

```
HW_POWER_MINPWR[HALF_FETS] = 0
```

```
HW_POWER_MINPWR[DOUBLE_FETS] = 1
```

- To enable the normal-size FETs:

```
HW_POWER_MINPWR[HALF_FETS] = 0
```

```
HW_POWER_MINPWR[DOUBLE_FETS] = 0
```

5.4 Using Pulse-Frequency Modulation (PFM) Mode

The DC-DC power usage can be reduced during very low power situations by using the PFM mode. This mode should be enabled only when the voltage outputs of the DC-DC are lightly loaded. The PFM mode causes higher transient noise and should only be used in deep sleep or standby modes when the PMU is basically inactive. It is good practice to make sure all clocks are configured in their 24 MHz crystal state before attempting to enter the PFM mode. Use the following statement to enable the PFM mode:

```
HW_POWER_MINPWR[EN_DC_PFM] = 1
```

When the device is exiting its deep sleep or standby mode, the PFM mode should be one of the first power savings features disabled. The PMU is not able to reliably supply power to the rails from the DC-DC in PFM mode in high-power situations.

6 Conclusion

The i.MX233 includes a highly-efficient and comprehensive power supply. The instructions and descriptions contained in the application note help any application using the i.MX233 to maximize its power performance while extending battery life.

7 References

The following references are available for download from the Freescale web site listed on the back page of this document:

- *i.MX23 Reference Manual* (IMX23RM)
- *i.MX23 Reference Design* (IMX23_EVK_SCH)

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

How to Reach Us:

Home Page:

www.freescale.com

Web Support:

<http://www.freescale.com/support>

USA/Europe or Locations Not Listed:

Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or
+1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064
Japan
0120 191014 or
+81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800 441-2447 or
+1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale and the Freescale logo are trademarks or registered trademarks of Freescale Semiconductor, Inc. in the U.S. and other countries. All other product or service names are the property of their respective owners. ARM is the registered trademark of ARM Limited.

© Freescale Semiconductor, Inc., 2009. All rights reserved.

