[1][2][3][4]

Texas Instruments

Application Report

SLDA039B–June 2015–Revised January 2020

How to Design Flexible Processor Power Systems Using PMICs

ABSTRACT

As systems continue to shrink in both size and power usage, while simultaneously growing in functionality, designers continuously face the challenge of how to effectively power embedded processor systems. Whether battery-powered or connected to a main power supply, embedded processor systems require an elegant power solution that can be implemented quickly and optimize board space. One option for creating the power tree is to use an individual power regulator integrated circuit (IC) for each rail of the processor, FPGA, or SoC. This is commonly referred to as a *discrete* solution. The other option is to use a multi-channel power management IC, or *PMIC*.

Commonly, the various voltage rails and current levels required by the processor and its peripherals are supplied by a handful of discrete power regulator ICs. Less complex systems can operate without power-up and power-down sequencing regulators, while different power states within the system are not always required. However, advanced embedded processors require controlled sequencing of the various power domains and need to achieve low power states to meet stricter industry standards for power consumption.

This application report provides examples of power solutions that can be applied to many processors and FPGAs currently available in the market and outlines the benefits of using a highly integrated PMIC versus the common discrete regulator approach.

Contents

List of Figures

(1) ARM is a registered trademark of Arm Limited.
(2) Bluetooth is a registered trademark of Bluetooth SIG, Inc.
(3) NXP is a trademark of NXP B.V.
(4) Wi-Fi is a registered trademark of Wi-Fi Alliance.

1 Area

Consumers expect electronics to continue reducing in size and cost. In order to save cost, PCBs are typically built without high density processes and components are frequently populated on one side of the board, thus reducing the useable area. The processor, memory, and peripheral connectors have a large fixed area without room for improvement. As a result, power management is a system block where designers frequently try and reduce board space. Using a single PMIC, in place of many external passive components, can help reduce board space as well as make schematic design and layout simpler.

In the discrete solution, each power rail requires its own dedicated IC. For example, Figure 1 shows a processor that needs two DC-DC converters plus two LDO regulators and requires additional space around these components. Each regulator requires a minimum of two external resistors for setting the output voltage, soft-start requires an additional capacitor, one or more pull-up or pull-down resistors for enable and power-good signals, and compensation could require another resistor and two additional capacitors. The majority of PMICs have all these passive components integrated.

In the discrete solution, if rail sequencing is required, additional passive components are needed in order to provide system sequencing and control, which is commonly referred to as *glue logic*. Many PMICs have all this analog and digital glue logic needed for supply sequencing already integrated into the chip, thus saving more board space. Figure 2 shows the PMIC that is suitable for replacing this discrete solution.

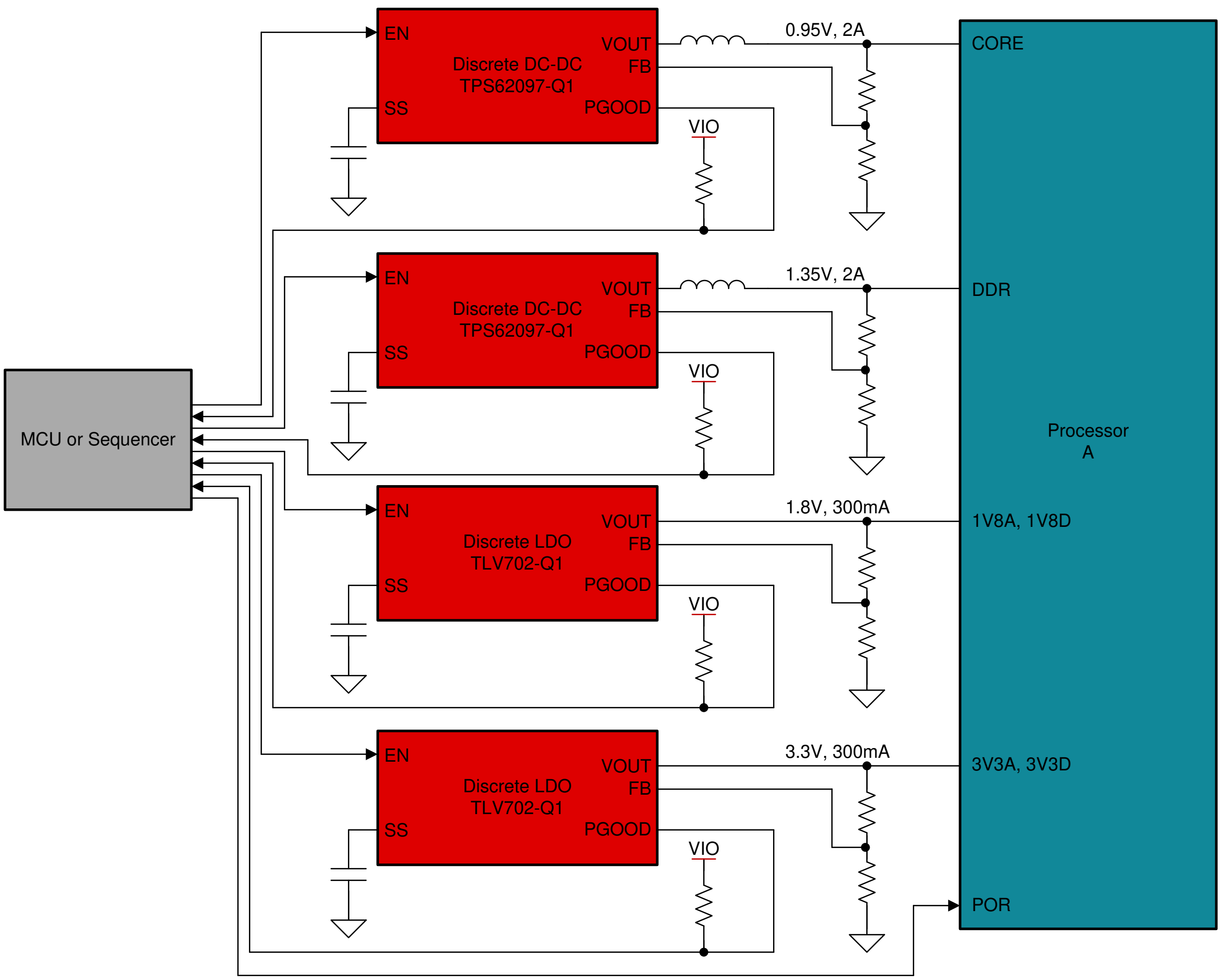

Figure 1. Discrete Processor Power

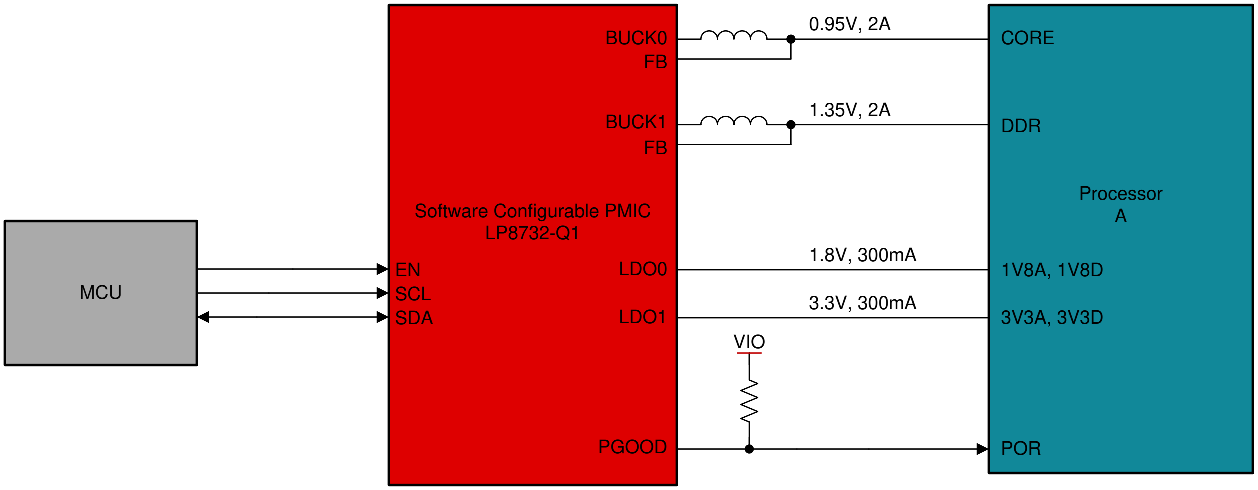

Figure 2. PMIC Processor Power

Figure 3 shows a to-scale example of PCB layouts of a discrete power solution versus a PMIC power solution powering the same embedded processor. The discrete solution requires many additional external components and routing which add to the total board area; whereas the PMIC integrates everything into a smaller overall solution size.

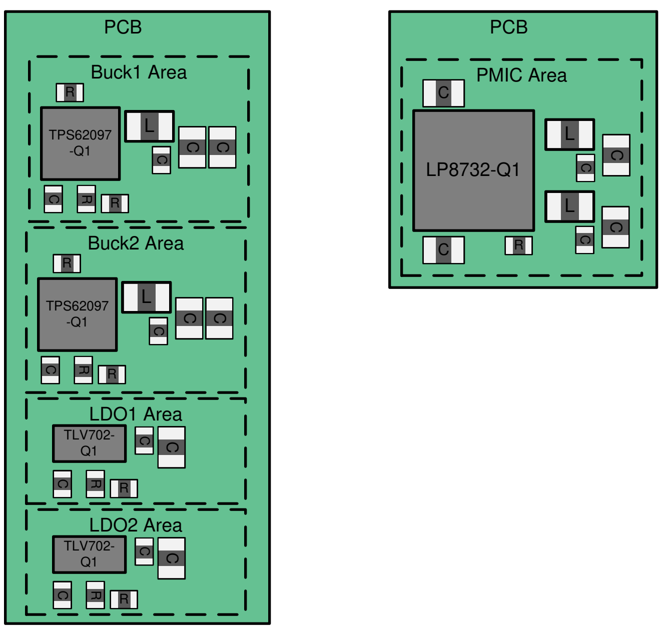

Figure 3. Example PCB Layout of Discrete versus PMIC Power Implementation

2 Flexibility and Scalability

Now let's imagine a scenario where the processor is changed, either to a different vendor with similar specifications or to a more powerful variant in the processor family. Choosing a processor from a different vendor may change the total number of rails in the system. For the more powerful variant, the core and DDR rails will consume more current. Additionally, in the previous example, we deliberately left out the power required for peripheral ICs, such as an ethernet PHY, Wi-Fi® plus Bluetooth® modules, or other I/O. Each of these ICs may add another voltage rail or consume more current from an existing rail. In this section, we will discuss how each of these deviations change the design.

2.1 Discrete

One of the biggest advantages of using a discrete solution is their flexibility when faced with changing power system requirements. Let's say the processor being powered requires four rails, each being powered by a discrete regulator. After the initial round of testing and verification, it is revealed that an additional 3.3 V power rail is needed in the system for powering digital I/O and peripheral ICs. With the discrete solution, it is an easy fix by simply adding one discrete regulator for that rail. Whereas with a quad output PMIC, a fifth rail is needed in the system. More or less rails may also be required when switching from an older processor to a newer processor, or changing from an ARM®-based processor to a SoC that integrates an FPGA and ARM-core processors.

Another advantage of the discrete solution is its ability to support the increasing power requirements of embedded processors. After performing power estimates, it is revealed that the core rail requires 3 A of current instead of 2 A. For the discrete solution you need to procure a 3 A buck regulator. This only changes one block in the layout, and it is typically assumed that a PMIC is not flexible or scalable enough to solve this problem. This scenario also applies to the case of switching to a more powerful variant of a processor: more current on the core rail is required when going from a single core to a quad-core processor or increasing the number of logic gates required in an FPGA.

2.2 Hybrid

Here, we are considering the example where a fifth rail is needed. Figure 4 shows the same PMIC solution from Figure 2 but with the additional 3.3 V rail. We can create a hybrid (PMIC plus discrete) solution by adding a buck regulator to satisfy the new rail. The hybrid solution cuts back on the total external passive components needed versus an all discrete solution. Figure 5 shows an externally configurable PMIC that integrates all 5 rails.

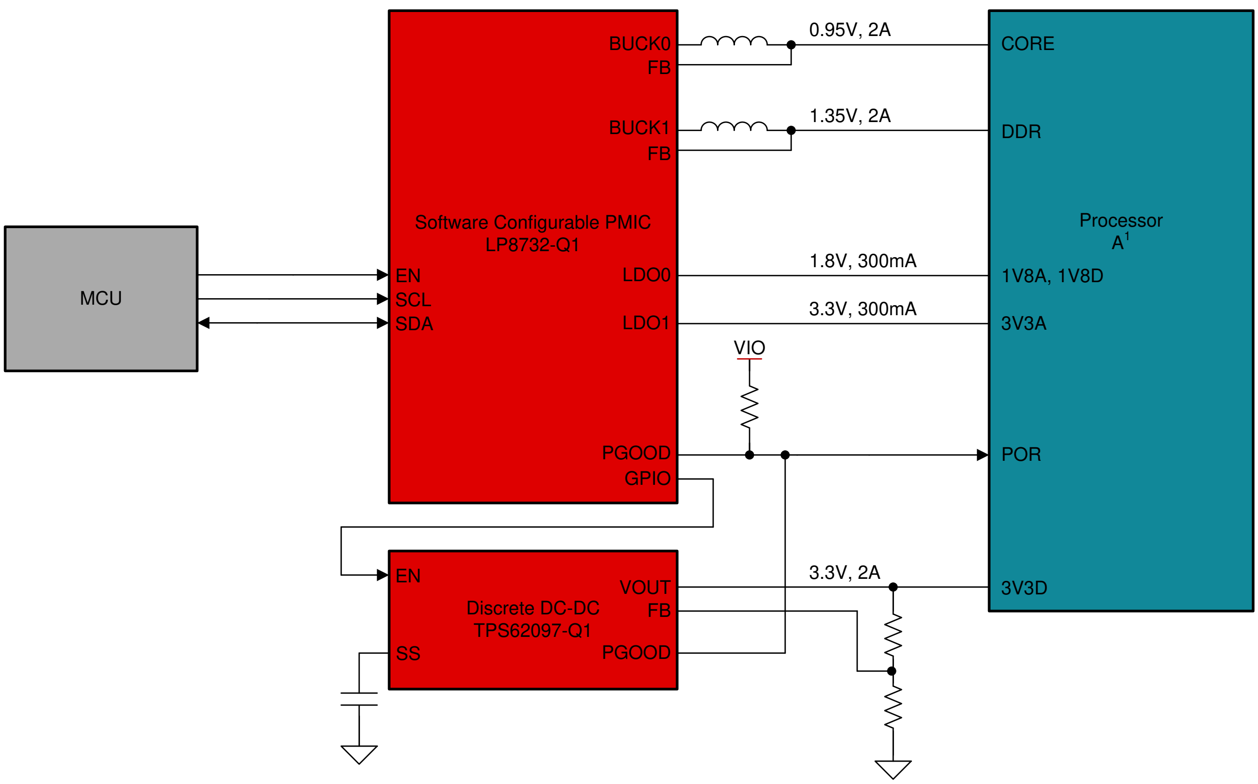

Figure 4. Hybrid (PMIC plus Discrete) Processor Power

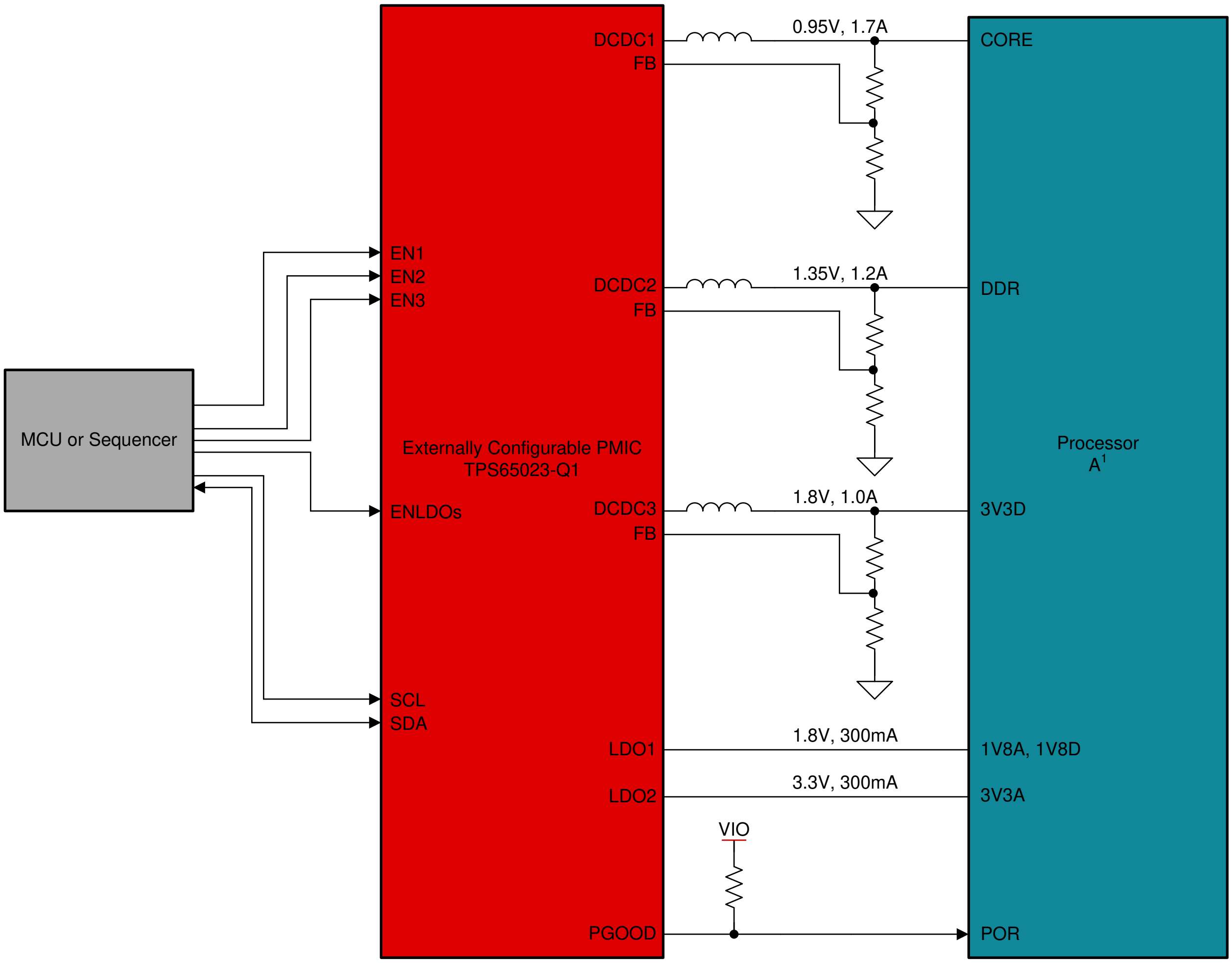

Figure 5. Hybrid (Externally Configurable PMIC) Processor Power

2.3 PMICs

Here, we are considering the example where the core rail current is increased. Figure 6 shows the same PMIC solution from Figure 2 but the device is replaced with a pin-to-pin compatible version of the device that is capable of delivering higher currents from the DC-DC converters.

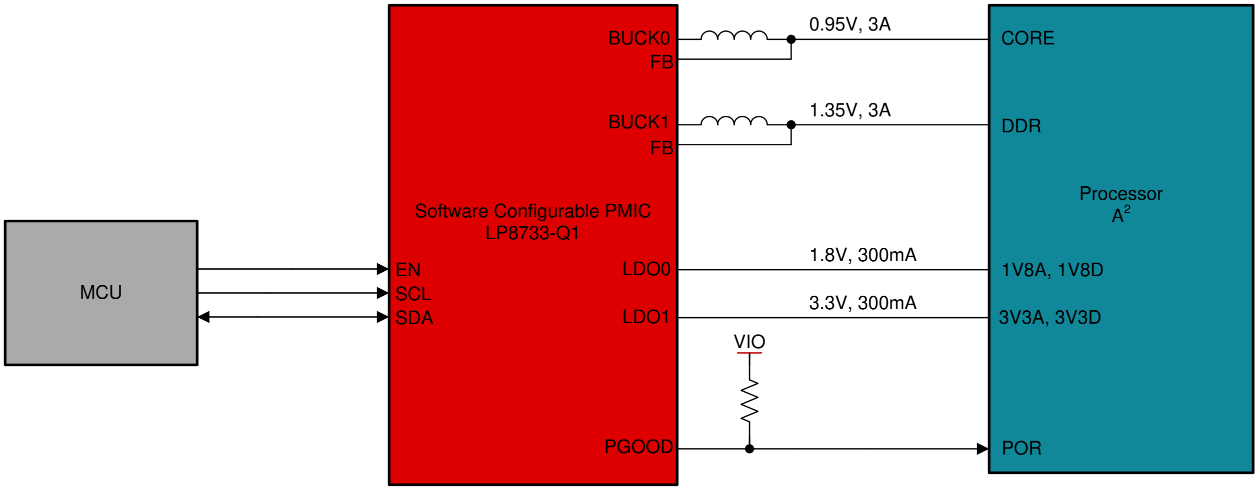

Figure 6. Higher Current PMIC Processor Power

2.3.1 Software Configurable

Some PMICs, such as the LP8732-Q1, LP8733-Q1, and LP87561-Q1, have "blank" factory programmed versions, which have been programmed to the lowest possible settings, and allow you to safely configure the PMIC to the desired settings through I^2C. The caveat to these software configurable variants is their volatile memory; the device must be programmed at every startup as the settings will be reset once powered down. These software configurable PMICs are the ideal solution if there is already a MCU onboard to control the PMIC.

Another advantage of PMICs relating to flexibility is their dynamic voltage scaling (DVS) capability for changing the voltage of an output rail. The processor sometimes refers to this as dynamic voltage/frequency scaling (DVFS), meaning the voltage for a rail is increased or decreased simultaneously with an increase or decrease in the processor clocking frequency. Achieving DVS in a discrete solution is challenging as it requires additional components (transistors and resistors) for the discrete rails to change the voltage to match the changing needs of the processor. The PMIC centralizes all the power rail controls into a single chip, rather than needing individual small drivers for each regulator that requires DVS.

To this point, we have discussed two types of PMICs: externally configurable and software configurable. In addition to these *configurable* PMICs, there are also two types of *programmable* PMICs that we will add to the list:

- **Externally configurable** – Externally configurable PMICs are similar to discrete devices, but with more than one rail and some digital logic integrated.
- **Software configurable** – A software configurable PMIC is controlled by an MCU instead of external passive components, typically using I^2C communication.
- **Factory-programmed** – The traditional type of PMIC is commonly referred to as a factory-programmed, or pre-programmed, PMIC. The programming is already done by TI for a specific use case and the PMIC is not flexible. Notice the word *programmed* is in the past-tense: at one time the PMIC was *programmable*, but the programming is already done and the device is not reprogrammable.
- **User-programmable** – A user-programmable PMIC contains a set of registers in non-volatile memory that are programmed to meet the needs of power a variety of processors, FPGAs, and SoCs. When the target processor is selected for an application, PMIC samples are programmed specifically for the power needs of that processor.

In the next section, we will discuss user-programmable PMICs and how they can provide a similar level of flexibility as externally configurable and software configurable PMICs.

2.3.2 User Programmable (EEPROM and OTP)

It would be a good idea to start our discussion of user-programmable PMICs, and how they differ from factory-programmed PMICs, by using specific parts as examples. The TPS65218D0 is an example of a factory-programmed PMIC that supports the AM335x and AM437x families of Sitara processors, while the TPS6521825 is intended to work in tandem with the LP873347 to power the NXP™ i.MX 8M Mini and Nano processors (both PMICs are factory-programmed).

If all the power capabilities of a factory-programmed PMIC were available and the end-user was allowed to modify the digital settings to determine how the device operated, then this device would be classified as a user-programmable PMIC. A user-programmable PMIC has the same feature set as a factory-programmed PMIC (automatic sequencing, internal feedback to set output voltages, digital glue logic), but the difference is that a user-programmable PMIC has a set of non-volatile memory that can be programmed to meet the needs of powering a variety of SoCs. Programming the nonvolatile memory ensures that the PMIC retains the settings specific to the target processor even after the system is reset, shut-down, or power-cycled. Similar to how the TPS65218D0 is pre-programmed for AM335x and AM437x processors, the TPS6521815 is a user-programmable PMIC that can be used to power NXP i.MX processors, Xilinx FPGAs, and Intel FPGAs. Figure 7 shows the same PMIC solution from Figure 5 but with a single user-programmable TPS6521815 device requiring minimal external components.

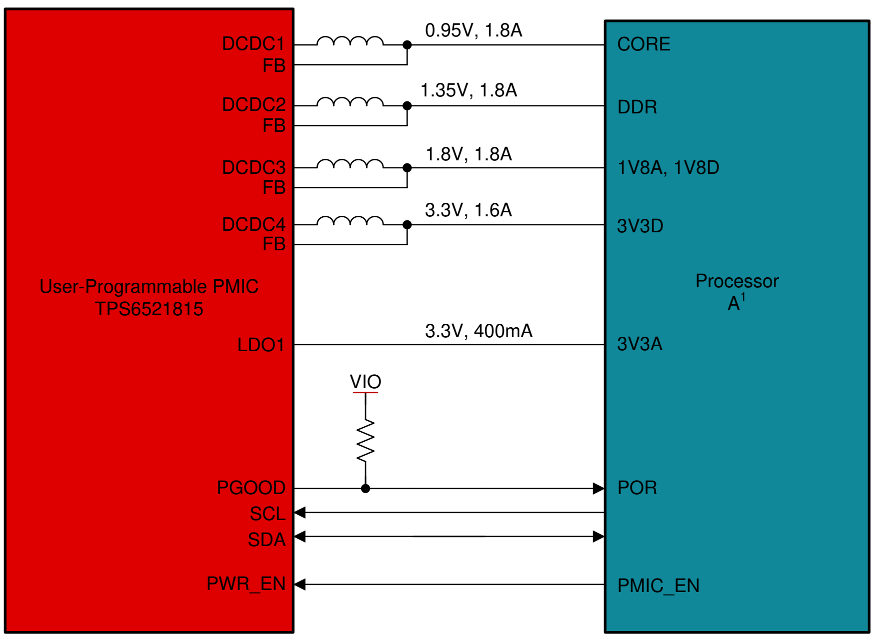

Figure 7. User-Programmable PMIC Processor Power

The TPS6521815 uses a bank of EEPROM memory that is used to program the output voltages, sequence order, sequence timing, and other settings to match the intended processor. Other examples of user-programmable PMICs that use EEPROM are the TPS652170 and the TPS6594-Q1. An example of a user-programmable PMIC that uses one-time programmable (OTP) memory is the TPS650861.

When a user-programmable PMIC is determined to meet the system power requirements, the PMIC offers performance benefits that are unique to the integrated solution and cannot be obtained through any discrete power solution. For example, the PMIC is able to achieve very low supply quiescent current (I_Q) and shutdown current (I_{OFF}) for the system due to the shared bias and control lines. For example, the TPS6521815 device is a PMIC that offers low I_Q, DVS, I^2C-control of individual rails, and warm-reset for the core rails (DCDC1 and DCDC2). All of these features are required to achieve low-power modes of the processor.

3 Integration of Analog and Digital Logic

The most obvious value of using a PMIC in an embedded processor system is the high level of integration, but usually that is interpreted only as reduced PCB area. Sometimes it is difficult to understand all of the value-add from combining a large feature-set into a single IC. Some examples of these often-overlooked PMIC features are power-up and power-down sequencing, external event detection, and system monitoring and fault handling. These features eliminate the need for additional ICs such as sequencers, supervisors, and temperatures sensors that can add to the area and complexity of a discrete power solution.

3.1 External Event Detection

Most systems rely on some external events (a push-button press, application of line-power, or assertion of a GPIO) to wake up the system or to reset (reboot) the system. A PMIC can detect these events and wake up the system or notify the processor of an event using an interrupt and I^2C communication. External event detection is not always limited to digital signals. For example, many PMICs include comparators that can be used to detect early power failure of the main power supply and notify the processor in time to begin the preferred power-down procedure.

3.2 Power-Up and Power-Down Sequencing

Power-up and power-down control are particularly important for the application processor because multiple functional blocks within the processor have critical timing dependencies. Intelligent power management must also handle the increasing number of channels. The sequence-up phase is typically initiated by a single enable signal or a combination of external events. The controlled supplies of the PMIC sequence-up with the correct order and timing.

All power supplies must exceed a power-good threshold within the configured time-out value. If any individual power rail fails to turn on properly, a sequence fault occurs, and all controlled supplies are shut down. When all supplies reach their sequence-up threshold, the supply monitor begins. PMICs integrate the power sequencing into their digital core and only a single power-good (PGOOD, or nPOR) signal is required for all of the rails in the system. This integrated feature eliminates the need for external sequencers and supervisors to control the sequencing for the system.

3.3 System Monitoring and Fault Handling

Supply and fault monitoring is critical for any system. Faults like over-current, under-voltage (PGOOD threshold), over-voltage, over-temperature, and main supply UVLO thresholds can be damaging to the system. A PMIC can detect all these faults and take immediate action without waiting for the main processor to take control to avoid system or power failure. When a fault is detected, it is communicated to the main processor; in parallel, a graceful shutdown sequence is initiated by the PMIC to prevent damage to the system.

4 Summary

In this application report, we have shown and discussed a variety of PMIC options that can be used to power embedded processors, even as the power requirements of the processor change. As a reference, Table 1 lists the high-level differences between the four different categories of PMICs in this application report.

Table 1. PMIC Categories

	Externally Configurable	Software Configurable	Factory Programmed	User Programmable
DVS, I^2C, multi-channel PGOOD, other analog and digital logic	✓(1)	✓	✓	✓
Internal feedback network		✓	✓	✓
Operate without MCU control	✓		✓	✓
Supports multiple SoCs (processors, FPGAs)	✓	✓		✓
Automatic sequencing			✓	✓
Non-volatile memory (NVM)			✓	✓
End-user allowed to re-program NVM				✓

(1) Some externally configurable PMICs are completely analog ICs and do not support DVS or I^2C

Designing power for embedded processor systems can be a difficult task due to area constraints and changes in the processor requirements. PMICs offer an obvious advantage when it comes to reducing the area of the PCB, but are not commonly seen as being flexible or scalable when the power requirements of the system are modified. This stigma is associated with the factory-programmed PMIC that only pairs well with a single processor. When flexibility and scalability is a concern, a discrete power implementation is frequently implemented. However, there are many different types of do-it-yourself (DIY) PMICs available to overcome this challenge: externally configurable, software configurable, and user-programmable. Once a PMIC can be seen as a flexible or scalable power device, then the multitude of other PMIC features (I^2C control, DVS, event detection, sequencing, fault handling) can also be used to design more impressive embedded processor systems.

5 References and Related Documentation

TI Power management multi-channel IC (PMIC) solutions

Power management for FPGAs and processors

Texas Instruments, *LP8756x-Q1 16-A Buck Converter With Integrated Switches* Data Sheet

Texas Instruments, *LP8733xx-Q1 Dual High-Current Buck Converter and Dual Linear Regulator* Data Sheet

Texas Instruments, *LP8732xx-Q1 Dual High-Current Buck Converter and Dual Linear Regulator* Data Sheet

Texas Instruments, *TPS65023-Q1 Power Management IC (PMIC) With 3 DC/DCs, 3 LDOs, I 2C Interface and DVS* Data Sheet

Texas Instruments, *TPS6521815 User-Programmable Power Management IC (PMIC) With 6 DC/DC Converters, 1 LDO, and 3 Load Switches* Data Sheet

Texas Instruments, *TPS652170 Programmable PMIC for Battery-Powered Systems* Data Sheet

Texas Instruments, *TPS650861 Programmable Multirail PMU for Multicore Processors, FPGAs, and Systems* Data Sheet

Texas Instruments, *TPS6594-Q1 Power Management IC (PMIC) for Processors with 5 Bucks and 4 LDOs* Data Sheet

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Original (June 2015) to B Revision	Page
• Added List of Figures	1
• Updated *Area* section	2
• Deleted *Cost* section	2
• Updated *Flexibility and Scalability* section to include *Discrete Hybrid* and *PMICs* sections	4
• Added *Summary* section	9

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265