RAIL SEQUENCING IN EP71XX INTEGRATED VOLTAGE REGULATORS

AN-EP71xx-1

EMPOWER SEMICONDUCTOR, INC.
FEBRUARY 2024

1. Table of Contents

Programming Rail Sequencing in EP71xx IVRs

2. Summary

Empower Semiconductor's EP71xx series is a family of Step-Down Integrated voltage regulators (IVRs) available in single to quad-output configurations. The EP71xx IVRs offer capability to customize and program output voltage, soft-start times, and power up/down sequencing for each of the outputs independently. This application note will explain how to achieve the desired power up/down sequencing by programming the user-configurable registers, through various examples.

The EP71xx Product Description and Datasheet provide more details on the ratings, electrical characteristics, operating characteristics, and functional features of the product family.

3. Introduction

Many complex systems in the industry use FPGAs, microprocessors, DSPs etc., all usually requiring different rail voltages to power the memory, core, auxiliary outputs, and input/outputs (I/Os). Furthermore, these require stringent power up and down sequencing of the rails.

Power sequencing is the order of the startup and shutdown of the various rails within the system executed in a controlled manner with defined time intervals and dependencies between each rail. To prevent high inrush currents, prevent latch-ups, and to ensure reliable and correct operation, sequencing of the multiple rails is crucial and is an important aspect of the overall system design. Improper sequencing of the rails can cause incorrect operation or even damage the system. FPGA vendors such as Xilinx specify either a recommended or required power-up/down sequence in their datasheets. In addition, timing requirements such as soft-start ramp rates and delays are also listed in the datasheets. In most cases, the recommended power-down sequence is the reverse order of the power-up sequence but the EP71xx is flexible enough to accommodate any order needed.

The EP71xx family features the ability to prototype the sequencing the multiple outputs through software and program the target sequence in the device through MTP (multi-time programmable) memory. This enables *'default'* power-up sequencing in the production system, removing the need for a system micro-controller to control or program the sequence. More information about MTP option can be found in the product datasheet or the MTP programming app note.

4. Programming Power Up/Down Sequencing

The EP71xx features digitally controllable power-up/down sequencing allowing the user to program the order and delay times of the startup and shutdown of the various voltage regulator rails (VRs). Figure 1 show an example of a programmed sequence using EP7144.

Note: EP7144 is a quad-output IVR in the EP71xx series and is used for discussion in this document.

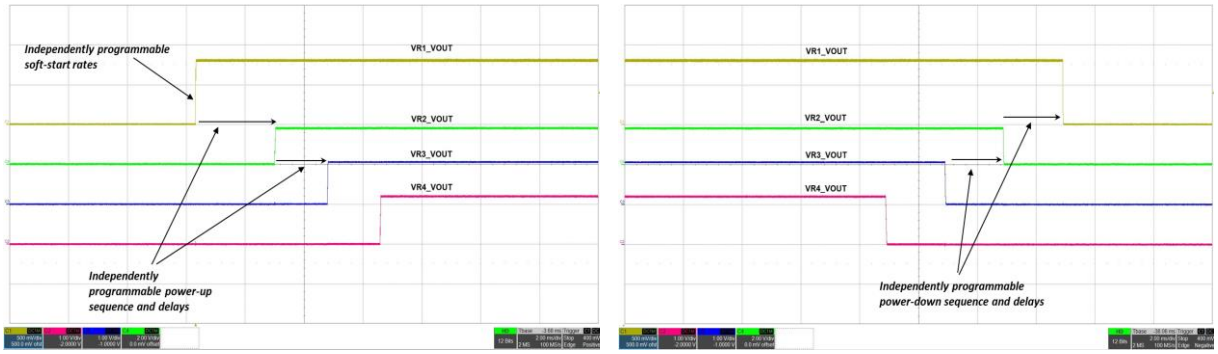

Figure 1. EP7144 Rail Sequencing: Programmable delay and power up/down of VRs

4.1 User-Programmable Registers for Sequencing

Table 1 lists the user-programmable registers to program power-sequencing including delays, GPIO functions, VR output voltages, and soft-start ramp rates. Each of these registers is further explained in this Section 6.1.

During early design/prototype phases, these registers can be used to perform trial runs on sequences either using the EP71xx Demo Board and associated GUI or on an application-specific system board with an on-board controller. For production, the register values determined to suit the application can subsequently be programmed into the MTP memory to provide the default power-up.

Register Name (User-programmable registers)	Bits	Reg Address			
		VR1	VR2	VR3	VR4
VRn_ENABLE - Table 12	[3:0]	0x20			
GPIO0_FUNCTIONS - Table 13	[6:0]	0x23			
GPIO1_FUNCTIONS - Table 13	[6:0]	0x24			
VRn_POWERSEQUENCE - Table 14, Table 15, Table 16	[7:0]	0x25	0x27	0x29	0x2B
VRn_DELAY - Table 14, Table 18	[7:0]	0x26	0x28	0x2A	0x2C
VRn_DAC_TARGET - Table 19	[9:8]	0x41	0x51	0x61	0x71
	[7:0]	0x42	0x52	0x62	0x72
VRn_CNTRL - Table 20	[7:0]	0x45	0x55	0x65	0x75

Table 1: User-Programmable Registers table

In **Figure 1**, the output voltage of each rail is programmed to a different voltage setting. This is set through **VRn_DAC_TARGET** registers. In addition, the soft-start rates can be programmed for each rail independently through **VRn_SOFT_START_RATE** register bit-fields in **VRn_CNTRL** register.

VRn_START_CONDITION and **VRn_HOLD_CONDITION** register bit-fields in **VRn_POWERSEQUENCE** can be used to define any dependency and order between the individual VR rails during power-up/down. Delay times for the startup and shutdown of the various rails can be programmed through Register **VRn_DELAY.**

EP71xx has two general purpose input/output pins (GPIO1, GPIO2) that can be configured to perform functions such as 'Hardware Enable', 'Alternate Power-Good output', 'Alerts output' and 'DVS input' through the registers **GPIO0_FUNCTIONS, GPIO1_FUNCTIONS**. In this application note, GPIOx configured as a hardware enable is discussed.

The following sections explain the power up/down sequencing through examples.

4.2 Enabling or Disabling of VR Rails

The EP71xx's individual VR rails can be enabled or disabled in four different ways as listed below.

It is important to note that the device EN pin (chip enable) should always stay high to keep the VRs enabled. If the EN pin goes low, all rails will shut down irrespective of the programmed power-down sequence.

1. **Standard I²C command (software enable)**: A VRn can be enabled by writing to the **VRn_ENABLE** registers through an I²C command (where, n = 1 to 4, VRn represent each of the VR rails in the EP71xx device).

2. **GPIO0/1 setting (hardware enable)**: GPIOx can be configured as a hardware enable pin to power up one or all the rails. Setting GPIOx high turns on the rail while setting GPIOx low will disable the specified rail.

   In many systems, a micro-controller or similar device is unavailable to coordinate sequencing, and the main system processor needs its core powered up prior to its IO functions being available. Hence, a method for initiating a sequence through hardware is required. This can be accomplished using GPIOx in EP71xx.

   Note: When a GPIOx pin is configured as HW Enable for a VRn, the corresponding **VRn_ENABLE** register bit is not set to 1. The **VRn_ENABLE** bits and the HW Enable functions are logically OR'ed. The VRn can be disabled only by pulling the GPIO voltage low or by setting chip EN pin to low.

3. **Start/hold conditions defined to depend on another VR**: A rail can be powered up based on another VR rail being enabled and in regulation, or the rail can be powered down based on another VR being disabled. A timing associated with the dependent VRn condition can also be programmed for both enable or disable delay.

   Note: When a rail is enabled through a start condition, its corresponding **VRn_ENABLE** register bit is not set to 1. The VRn can be disabled only through a stop condition associated with any other VRn status (refer to example in Figure 5) or by setting the chip EN pin to low.

4. **MTP Enable register**: The **MTP_VRn_ENABLE** register is available for programming through MTP to force a VR to be enabled as soon as EN is pulled high. For more information on this, refer to the datasheet and the MTP programming app note.

In most FPGA vendor datasheets, the recommended or required power-up sequence is in the order of:
Core voltage (0.85V/0.9V) → Auxiliary supply (1V/1.2V) → I/O supplies (1.8V)

Hence, these voltage rails and power-up/down sequence have been used in the examples in this application note. The power-down sequence is typically the reverse order of the power-up sequence.

Scenario 1 and Scenario 2 describe enabling and disabling of VR through I²C and GPIO respectively.

Scenario 3 and Scenario 4 explain power-up/down sequences with start/hold conditions and delays applied to the individual rails.

4.2.1 Scenario 1: Enabling/disabling VR through I²C

In Figure 2, VR1, VR3, VR4 of EP7144 are all enabled individually by writing to each **VRn_ENABLE** register through the I²C bus. No programmable delays have been set in this case. After issuing the 'Execute' command, each VR is enabled with the minimum default delay of 40µs.

To disable the VRs, set **VR1_ENABLE**, **VR3_ENABLE** and **VR4_ENABLE** to zero through the I²C bus.

Loads all registers as shown in Table 3 to achieve this power-up sequence.

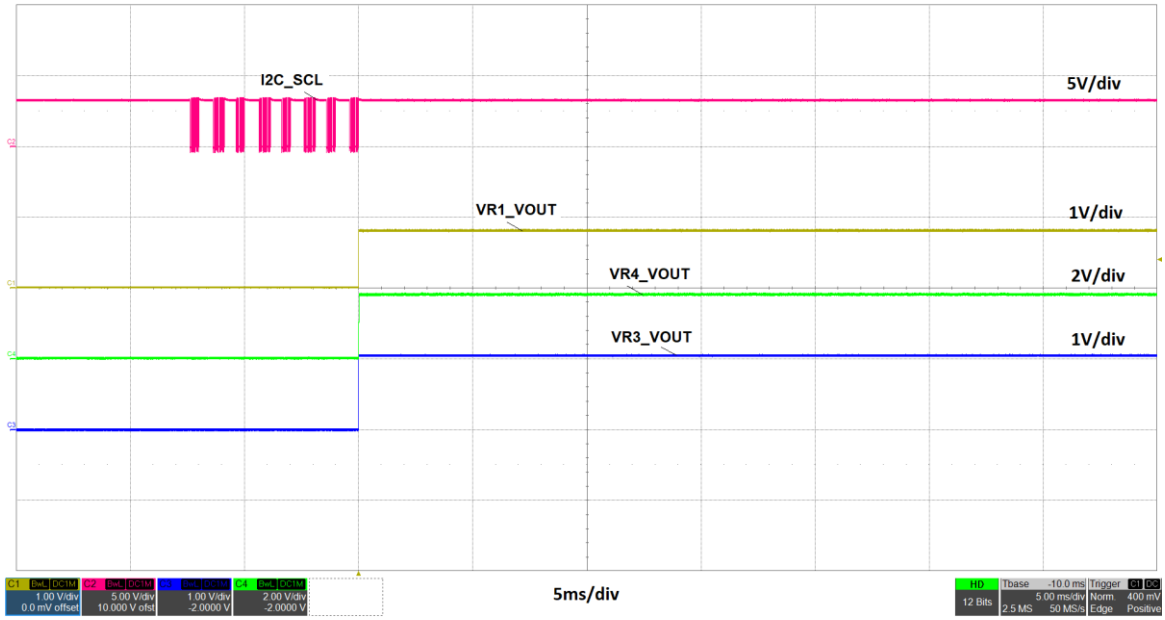

Figure 2. Enabling rails VR1, VR3, VR4 through I²C with no delays

Channel #	Waveform Name	Waveform details	Details	Measured delay
CH2 (magenta)	I²C_SCL	SCL waveform of I²C	I²C command to enable VR1, VR3, VR4	-
CH1 (yellow)	VR1_VOUT	Output Voltage of VR1	Set to 0.8V. Enable VR1 through I²C	No programmable delay added
CH3 (blue)	VR3_VOUT	Output Voltage of VR3	Set to 1V. Enable VR3 through I²C	No programmable delay added
CH4 (green)	VR4_VOUT	Output Voltage of VR4	Set to 1.8V. Enable VR4 through I²C	No programmable delay added

Table 2: Waveform description for Figure 2

Bit field Name	Register Address	Bits	Reg Setting (hex)	Description of changes
VR1_DAC_TARGET	0x41	[9:8]	0x148	Sets VR1 Output Voltage to 0.8V
	0x42	[7:0]		
VR3_DAC_TARGET	0x61	[9:8]	0x19A	Sets VR3 Output Voltage to 1V
	0x62	[7:0]		
VR4_DAC_TARGET	0x71	[9:8]	0x2E2	Sets VR4 Output Voltage to 1.8V
	0x72	[7:0]		
VR1_ENABLE	0x20	[0]	1	Enable VR1
VR3_ENABLE		[1]	1	Enable VR3
VR4_ENABLE		[2]	1	Enable VR4
VR1_DELAY_UP	0x25	[7]	0	No start/hold conditions, default settings
VR1_DELAY_DOWN		[6]	0	
VR1_START_CONDITION		[5:3]	0x00	
VR1_HOLD_CONDITION		[2:0]	0x00	
VR1_DELAY	0x26	[7:0]	0x00	No delay, default settings
VR3_DELAY_UP	0x29	[7]	0	No start/hold conditions, default settings
VR3_DELAY_DOWN		[6]	0	
VR3_START_CONDITION		[5:3]	0x00	
VR3_HOLD_CONDITION		[2:0]	0x00	
VR3_DELAY	0x2A	[7:0]	0x00	No delay, default settings
VR4_DELAY_UP	0x2B	[7]	0	No start/hold conditions, default settings
VR4_DELAY_DOWN		[6]	0	
VR4_START_CONDITION		[5:3]	0x00	
VR4_HOLD_CONDITION		[2:0]	0x00	
VR4_DELAY	0x2C	[7:0]	0x00	No delay, default settings
EXECUTE	0x0F	[7:0]		Write any value to apply register settings

Table 3: Register settings for Figure 2

4.2.2 Scenario 2: Enabling/disabling VR using GPIO as Hardware Enable

In Figure 3, GPIO0 is programmed as HW Enable for VR1, VR3, VR4. No programmable delays have been set in this case, and no start/hold conditions between VRs have been included. The 'Execute' command is issued. When GPIO0

is set HI, the rails VR1, VR3, VR4 power-up with the minimum default delay of 40µs. The rails power-down when GPIO0 goes low.

Table 5 shows the register settings to achieve this power-up sequence described in scenario 2.

Note: When GPIOx is configured as HW Enable for a VRn, the corresponding **VRn_ENABLE** register bit is not set to 1. The VRn can be disabled only by pulling the GPIO voltage low, or by setting chip EN pin to low.

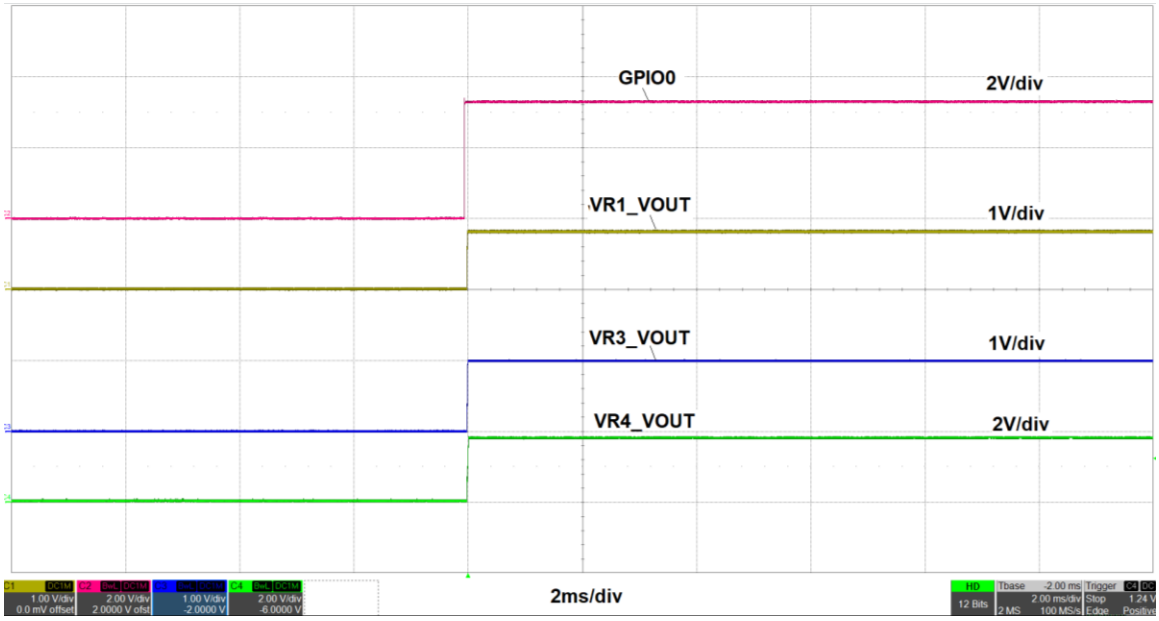

Figure 3. Hardware-enable of VR1, VR3, VR4 through GPIO0 (configured as enable for VR1, VR3, VR4) with no delays

Channel #	Waveform Name	Waveform details	Details	Measured delay
CH2 (magenta)	GPIO0	GPIO0 voltage	GPIO0 is set to function as 'Hardware Enable' for VR1, VR3, VR4.	-
CH1 (yellow)	VR1_VOUT	Output Voltage of VR1	Set to 0.8V. Enabled through GPIO0	No programmable delay added
CH3 (blue)	VR3_VOUT	Output Voltage of VR3	Set to 1V. Enabled through GPIO0	No programmable delay added
CH4 (green)	VR4_VOUT	Output Voltage of VR4	Set to 1.8V. Enabled through GPIO0	No programmable delay added

Table 4: Waveform description Figure 3

Bit field Name	Register Address	Bits	Reg Setting (hex)	Description of changes
VR1_DAC_TARGET	0x41	[9:8]	0x148	

	0x42	[7:0]		Sets VR1 Output Voltage to 0.8V
VR3_DAC_TARGET	0x61	[9:8]	0x19A	Sets VR3 Output Voltage to 1V
	0x62	[7:0]		
VR4_DAC_TARGET	0x71	[9:8]	0x2E2	Sets VR4 Output Voltage to 1.8V
	0x72	[7:0]		
VR1_ENABLE	0x20	[0]	0	Default settings
VR3_ENABLE		[1]	0	Default settings
VR4_ENABLE		[2]	0	Default settings
GPIO0_FUNCTION	0x23	[6:4]	0x04	GPIO0 set as HW Enable
GPIO0_VRS		[3:0]	0x0D	GPIO0 set HW EN for VR1, VR3, VR4
VR1_DELAY_UP	0x25	[7]	0	No start/hold conditions, default settings
VR1_DELAY_DOWN		[6]	0	
VR1_START_CONDITION		[5:3]	0x00	
VR1_HOLD_CONDITION		[2:0]	0x00	
VR1_DELAY	0x26	[7:0]	0x00	No delay, default settings
VR3_DELAY_UP	0x29	[7]	0	No start/hold conditions, default settings
VR3_DELAY_DOWN		[6]	0	
VR3_START_CONDITION		[5:3]	0x00	
VR3_HOLD_CONDITION		[2:0]	0x00	
VR3_DELAY	0x2A	[7:0]	0x00	No delay, default settings
VR4_DELAY_UP	0x2B	[7]	0	No start/hold conditions, default settings
VR4_DELAY_DOWN		[6]	0	
VR4_START_CONDITION		[5:3]	0x00	
VR4_HOLD_CONDITION		[2:0]	0x00	
VR4_DELAY	0x2C	[7:0]	0x00	No delay, default settings
EXECUTE	0x0F	[7:0]		Write any value to apply register settings

Table 5: Register settings for Figure 3

4.2.3 Scenario 3: Enabling/disabling VR through I²C with start/hold conditions and delay

Figure 4 and Figure 5 show the power-up and power-down of three rails with start/hold conditions and programmable delays. The sequence in this case is controlled directly through I²C commands and not through GPIOs.

For the sequences described in scenario 3, load registers as shown in Table 8.

During power-up, Rail VR1 is enabled through the I²C command (VR1_ENABLE = 1). Rail VR3 and Rail VR4 are enabled through start conditions with delays – VR3 power-up depends on VR1 power-up and VR4 depends on VR3.

Note: When a rail is enabled through a start condition, its corresponding **VRn_ENABLE** register bit is not set to 1. In this case, VR3_ENABLE and VR4_ENABLE are not set to 1. The VRn can be disabled only through a stop condition associated with any other VRn status (refer to example in Figure 5) or by setting the chip EN pin to low.

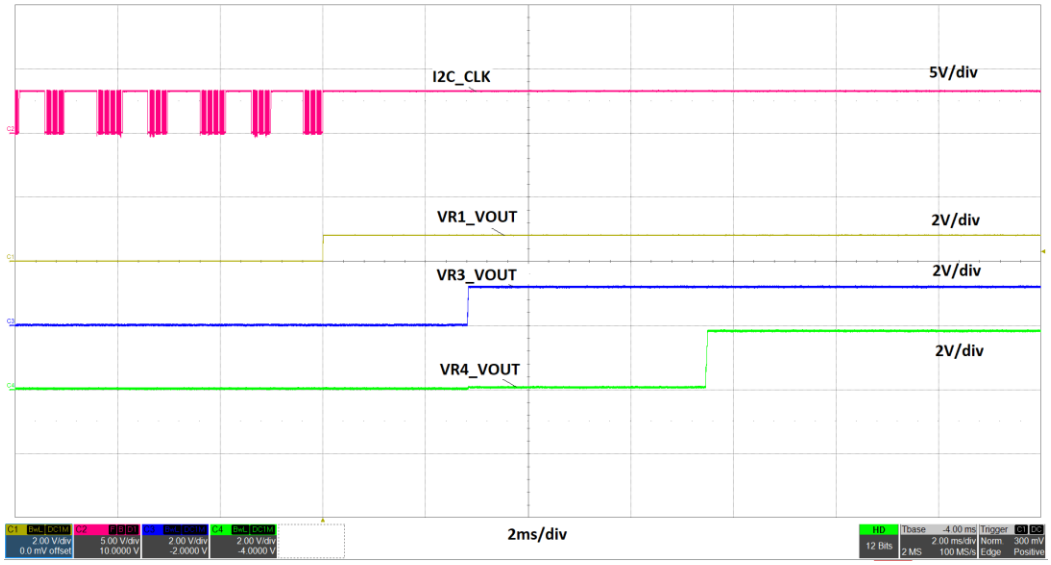

Figure 4. Enabling VR1 through I²C, power-up of VR3, VR4 with start conditions and programmable delays

Channel #	Waveform Name	Waveform details	Details	Measured delay
CH2 (magenta)	I²C_SCL	SCL waveform of I²C	I²C command to enable VR1	-
CH1 (yellow)	VR1_VOUT	Output Voltage of VR1	Set to 0.8V. Enable VR1 through I²C	No programmable delay added
CH3 (blue)	VR3_VOUT	Output Voltage of VR3	Set to 1.2V. Rail VR3 is dependent on VR1 through start condition; it powers-up after rail VR1 is enabled and has reached regulation.	Measured delay = 2.84ms from VR1.
CH4 (green)	VR4_VOUT	Output Voltage of VR4	Set to 1.8V. Rail VR4 powers-up after rail VR3 is enabled and has reached regulation; defined through start condition	Measured delay = 4.66ms from VR3.

Table 6: Waveform description Figure 4

Power-down sequence is the reverse of the power-up sequence. An I²C command is issued to disable VR1 (**VR1_ENABLE** = 0) which initiates a shutdown request. As VR1 has a hold condition dependent on VR3 which in turn depends on VR4 through the hold condition, VR1 continues to stay high. VR4 powers down first as it has no hold conditions, followed by VR3 and finally VR1.

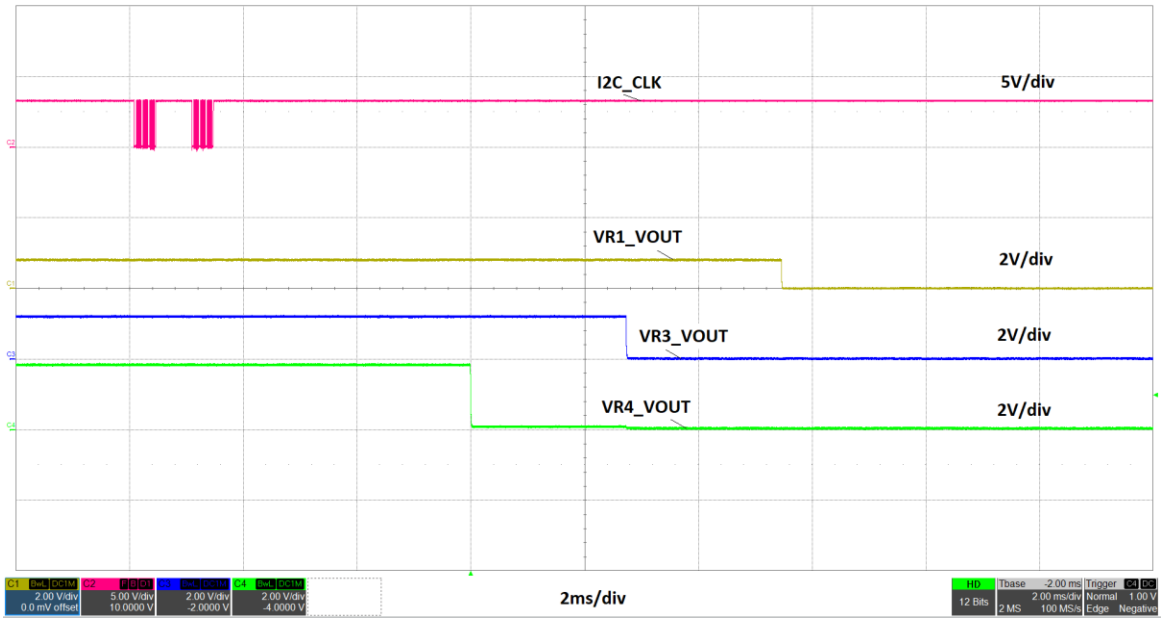

Figure 5. Disabling VR1 through I²C, power-down of VR3, VR4 with hold conditions and programmable delays

Channel #	Waveform Name	Waveform details	Details	Measured delay
CH2 (magenta)	I²C_SCL	SCL waveform of I²C	I²C command to disable VR1 (this initiates shutdown sequence)	-
CH1 (yellow)	VR1_VOUT	Output Voltage of VR1	Rail VR1 will hold the rail ON as long as VR3 is active defined through hold condition. VR1 will shut down only after rail VR3 has shutdown.	Measured delay = 2.73ms from VR3.
CH3 (blue)	VR3_VOUT	Output Voltage of VR3	Rail VR3 will hold the rail ON as long as VR4 is active defined through hold condition. VR3 will shut down only after rail VR4 has shutdown.	Measured delay = 2.73ms from VR4.
CH4 (green)	VR4_VOUT	Output Voltage of VR4	Rail VR4 disables first in the sequence. Set VR1_ENABLE = 0 through I²C command.	Measured delay = 4.7ms from I²C command.

Table 7: Waveform description Figure 5

Bit field Name	Register Address	Bits	Reg Setting (hex)	Description of changes
VR1_DAC_TARGET	0x41	[9:8]	0x148	Sets VR1 Output Voltage to 0.8V
	0x42	[7:0]		
VR3_DAC_TARGET	0x61	[9:8]	0x1EC	Sets VR3 Output Voltage to 1.2V
	0x62	[7:0]		
VR4_DAC_TARGET	0x71	[9:8]	0x2E2	Sets VR4 Output Voltage to 1.8V
	0x72	[7:0]		
VR1_ENABLE	0x20	[0]	1	Enable VR1 (for power-up sequence)
VR3_ENABLE		[1]	0	Default settings
VR4_ENABLE		[2]	0	Default settings
VR1_DELAY_UP	0x25	[7]	0	- No delay during power-up
VR1_DELAY_DOWN		[6]	1	- Delay applied during power-down
VR1_START_CONDITION		[5:3]	0x00	- No start condition
VR1_HOLD_CONDITION		[2:0]	0x02	- Shut down of Rail VR1 dependent on shutdown of Rail VR3
VR1_DELAY	0x26	[7:0]	0x3C	Sets delay time
VR3_DELAY_UP	0x29	[7]	1	- Delay applied during power-up and power-down
VR3_DELAY_DOWN		[6]	1	
VR3_START_CONDITION		[5:3]	0x01	- Startup of Rail VR3 dependent on startup of Rail VR1
VR3_HOLD_CONDITION		[2:0]	0x04	- Shutdown of Rail VR3 depends on shutdown of Rail VR4
VR3_DELAY	0x2A	[7:0]	0x3C	Sets delay time
VR4_DELAY_UP	0x2B	[7]	1	Delay applied during power-up and power-down
VR4_DELAY_DOWN		[6]	1	
VR4_START_CONDITION		[5:3]	0x04	- Startup of Rail VR4 dependent on startup of Rail VR3
VR4_HOLD_CONDITION		[2:0]	0x00	- No hold condition. Shutdown of Rail VR4 depends on I^2C command VR1_ENABLE = 0
VR4_DELAY	0x2C	[7:0]	0x64	Sets delay time
EXECUTE	0x0F	[7:0]		Write any value to register to apply register settings
For power-down sequence				
VR1_ENABLE	0x20	[0]	0	Disable VR1 (for power-down sequence)

EXECUTE	0x0F	[7:0]		Write any value to apply register settings

Table 8: Register settings for Figure 4 and Figure 5

4.2.4 Scenario 4: Enabling/disabling VR using GPIO as HW Enable with start/hold conditions and delay

In Figure 6 and Figure 7: Power-up/down sequence with GPIO0 set as HW Enable for rail VR1.

Setting GPIO0 as high initiates the power-up sequence. VR3 and VR4 are powered-up through start conditions; VR3 depends on VR1 startup and VR4 depends on VR3 startup.

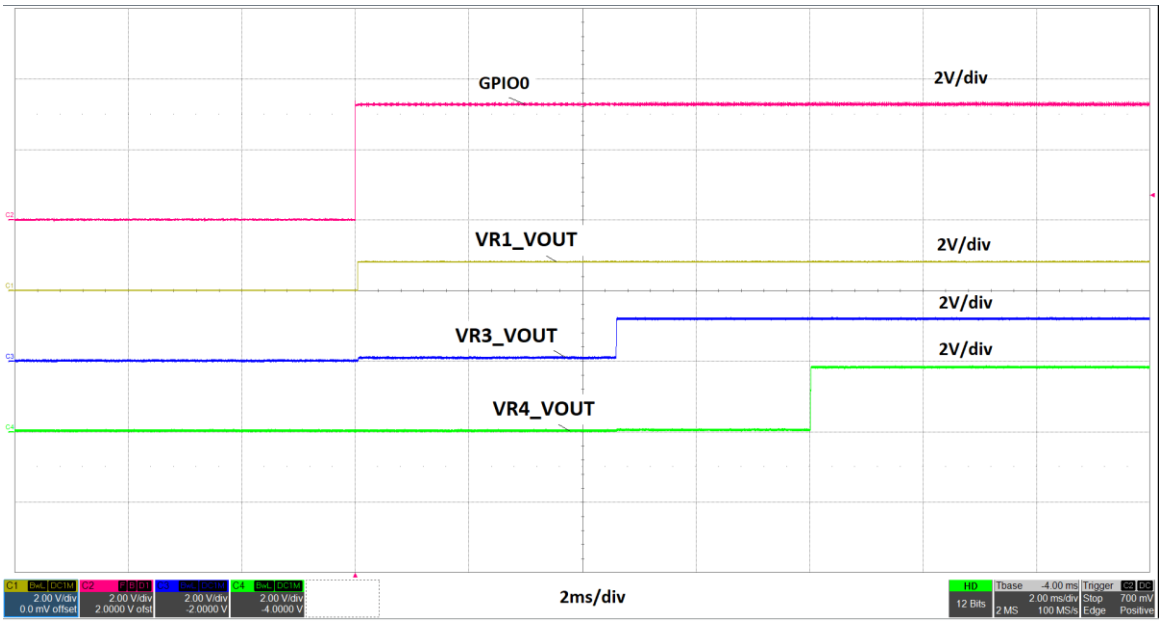

Figure 6. Enable of VR1 through GPIO0, power-up of VR3, VR4 through start conditions and with programmable delays

Channel #	Waveform Name	Waveform details	Details	Measured delay
CH2 (magenta)	GPIO0	GPIO0 voltage	GPIO0 is set to function as 'Hardware Enable' for VR1. Set high.	-
CH1 (yellow)	VR1_VOUT	Output Voltage of VR1	Set to 0.8V. Enabled using GPIO0 HW Enable.	No programmable delay added
CH3 (blue)	VR3_VOUT	Output Voltage of VR3	Set to 1.2V. Rail VR3 depends on VR1 through start condition; it will power-up only after rail VR1 is enabled and has reached regulation.	Measured delay = 4.6ms from VR1.

CH4 (green)	VR4_VOUT	Output Voltage of VR4	Set to 1.8V. Rail VR4 will power-up only after rail VR3 is enabled and has reached regulation; defined through start condition	Measured delay = 8ms from VR3.

Table 9: Waveform description Figure 6

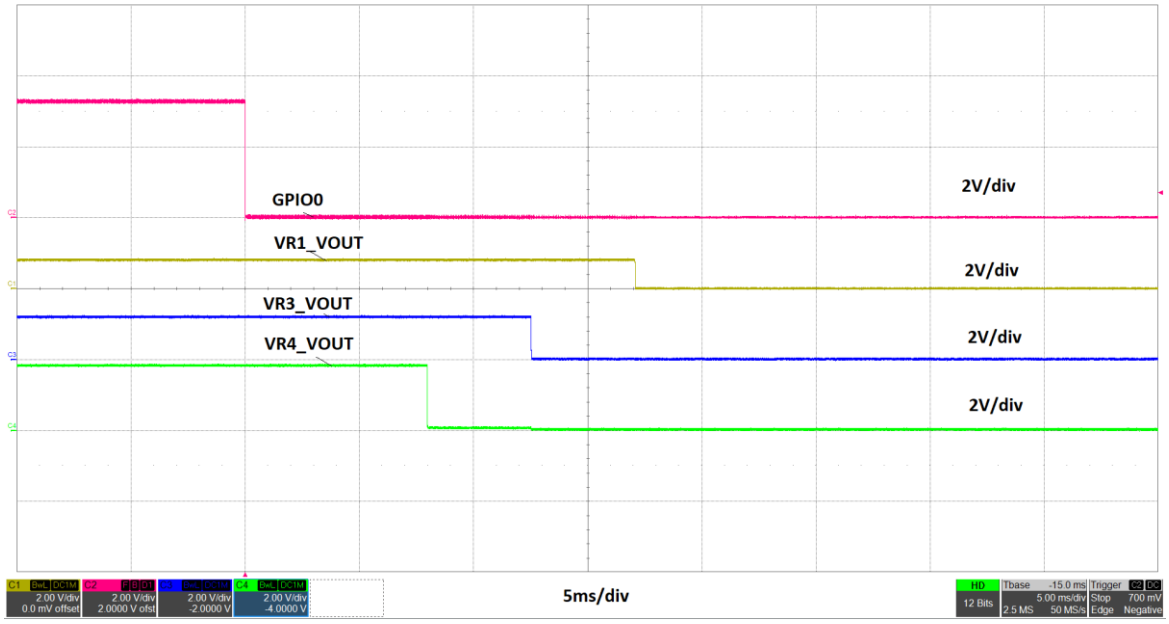

Figure 7. GPIO0 configured as HW-enable for rails VR1: here, GPIO0 is disabled and rails power-down with programmable delays and hold conditions

When GPIO0 is configured as HW Enable for a VR1, the corresponding **VR1_ENABLE** register bit is not set to 1. Power-down sequence starts when GPIO0 voltage is pulled low. VR1 will not power-down first as it has a hold condition dependent on VR3 rail. The shutdown request causes VR4 to power-down first as it has no dependencies. VR3 powers down based on the hold and delay conditions followed by VR1 powering down.

Channel #	Waveform Name	Waveform details	Details	Measured delay
CH2 (magenta)	GPIO0	GPIO0 voltage	GPIO0 is pulled low.	-
CH1 (yellow)	VR1_VOUT	Output Voltage of VR1	Rail VR1 shuts down only after rail VR3 shuts down; defined through hold condition.	Measured delay = 4.6ms from VR3.
CH3 (blue)	VR3_VOUT	Output Voltage of VR3	Rail VR3 shuts down only after rail VR4 has shutdown, defined in hold condition.	Measured delay = 4.5ms from VR4.

CH4 (green)	VR4_VOUT	Output Voltage of VR4	Rail VR4 disables when GPIO0 is pulled low.	Measured delay = 8ms from GPIO0 Disable.

Table 10: Waveform description Figure 7

Bit field Name	Register Address	Bits	Reg Setting (hex)	Description of changes
VR1_DAC_TARGET	0x41	[9:8]	0x148	Sets VR1 Output Voltage to 0.8V
	0x42	[7:0]		
VR3_DAC_TARGET	0x61	[9:8]	0x1EC	Sets VR3 Output Voltage to 1.2V
	0x62	[7:0]		
VR4_DAC_TARGET	0x71	[9:8]	0x2E2	Sets VR4 Output Voltage to 1.8V
	0x72	[7:0]		
VR1_ENABLE	0x20	[0]	0	Default settings
VR3_ENABLE		[1]	0	Default settings
VR4_ENABLE		[2]	0	Default settings
GPIO0_FUNCTION	0x23	[6:4]	0x04	GPIO0 set as HW Enable
GPIO0_VRS		[3:0]	0x01	GPIO0 set HW EN for VR1
VR1_DELAY_UP	0x25	[7]	0	- No delay during power-up - Delay applied during power-down - No start condition - Shut down of Rail VR1 dependent on shutdown of Rail VR3
VR1_DELAY_DOWN		[6]	1	
VR1_START_CONDITION		[5:3]	0x00	
VR1_HOLD_CONDITION		[2:0]	0x02	
VR1_DELAY	0x26	[7:0]	0x64	Sets delay time
VR3_DELAY_UP	0x29	[7]	1	- Delay applied during power-up and power-down - Startup of Rail VR3 dependent on startup of Rail VR1 - Shut down of Rail VR3 dependent on shutdown of Rail VR4
VR3_DELAY_DOWN		[6]	1	
VR3_START_CONDITION		[5:3]	0x01	
VR3_HOLD_CONDITION		[2:0]	0x04	
VR3_DELAY	0x2A	[7:0]	0x64	Sets delay time
VR4_DELAY_UP	0x2B	[7]	1	Delay applied during power-up and power-down - Startup of Rail VR4 dependent on startup of Rail VR3
VR4_DELAY_DOWN		[6]	1	
VR4_START_CONDITION		[5:3]	0x04	
VR4_HOLD_CONDITION		[2:0]	0x00	

				- No hold condition. Shut down of Rail VR4 dependent on GPIO0 disable. (GPIO0 set low).
VR4_DELAY	0x2C	[7:0]	0xAF	Sets delay time
EXECUTE	0x0F	[7:0]		Write any value to the register to apply register settings

Table 11: Register settings for Figure 6 and Figure 7

4.3 Relative Rail Sequencing

Most FPGAs have requirements on voltage ramp-up time for rails to prevent excess inrush current and ensure correct operation. A power-supply with adjustable soft-start ramp rates is usually preferred in FPGA applications.

There are typically three common types of multi-rail sequencing (discussed below). The EP71xx series can support all these rail sequencing methods.

4.3.1 Sequential rail sequencing

In this method, one supply rail is turned on first, followed by a set delay before the next rail is turned on. Examples of this method were discussed Section 4.2.

Figure 8 shows an example of sequential rail sequencing. Rails VR2 and VR3 are ramping up at the same soft-start rate. Rail VR2 ramp-up is followed by a fixed delay before rail VR3 ramps up. A consistent soft-start rate for different rails (dV/dt) enables a consistent and manageable inrush current.

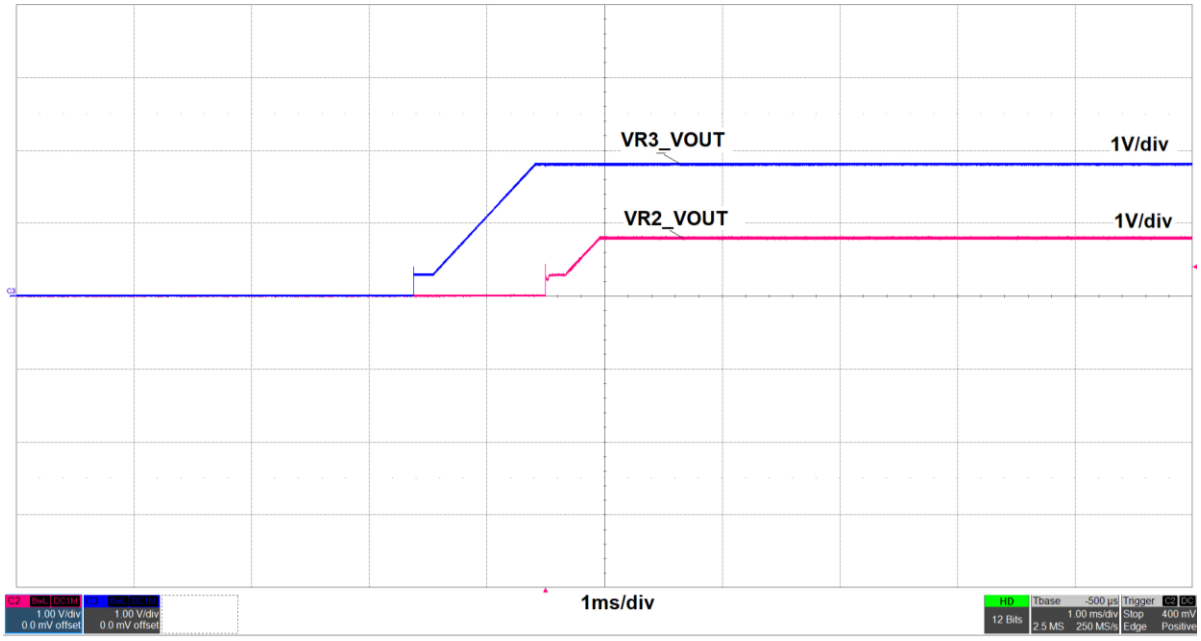

Figure 8. Sequential sequencing of rails VR2 and VR3 with same soft-start ramp rates, VR2 powers-up with delay

4.3.2 Simultaneous rail sequencing

In this method, voltage rails ramp-up together at the same soft-start ramp rate. The core voltage which is the lower voltage reaches its final value before the higher voltage rail (aux, I/O type of rails) has reached its final value.

Figure 9 shows an example of simultaneous rail sequencing. Rails VR2 and VR3 start ramping up at the same time at the same soft-start rate.

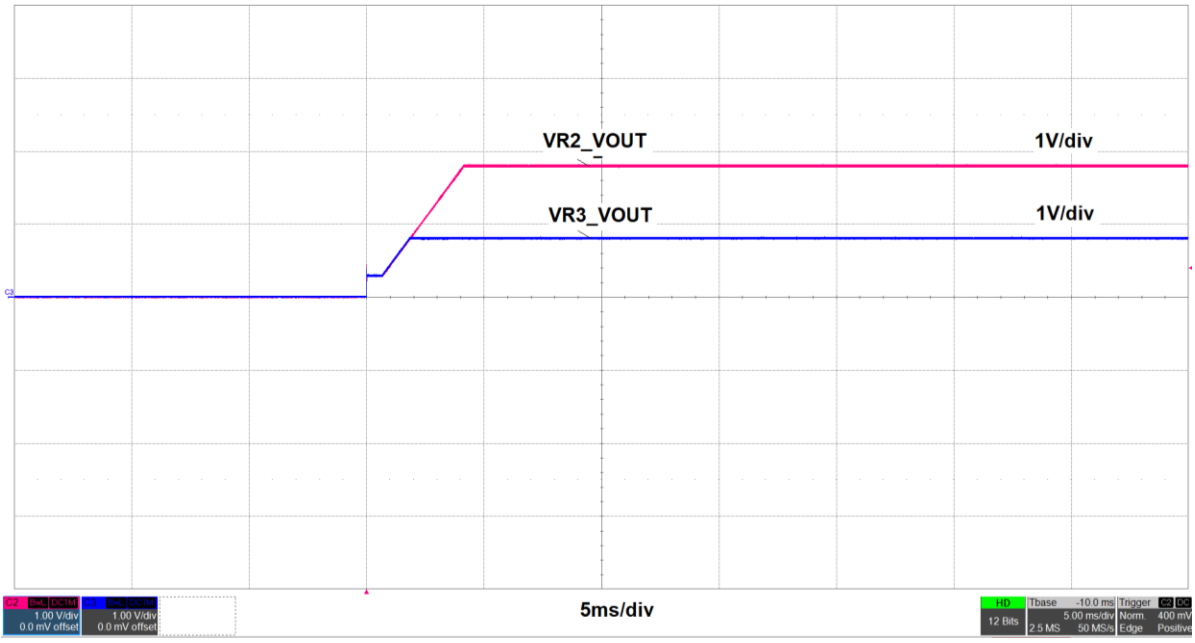

Figure 9. Simultaneous sequencing of rails VR2 and VR3 with same soft-start rates and with no delay

4.3.3 Ratiometric rail sequencing

In this method, the rails start-up at the same time and reach their final voltage levels at the same time. This requires the soft-start ramp rates to be set proportional to the rail voltage to achieve regulation at the same time.

Figure 10 shows an example of simultaneous rail sequencing. Rails VR2 and VR3 start ramping up at the same time at different soft-start rates.

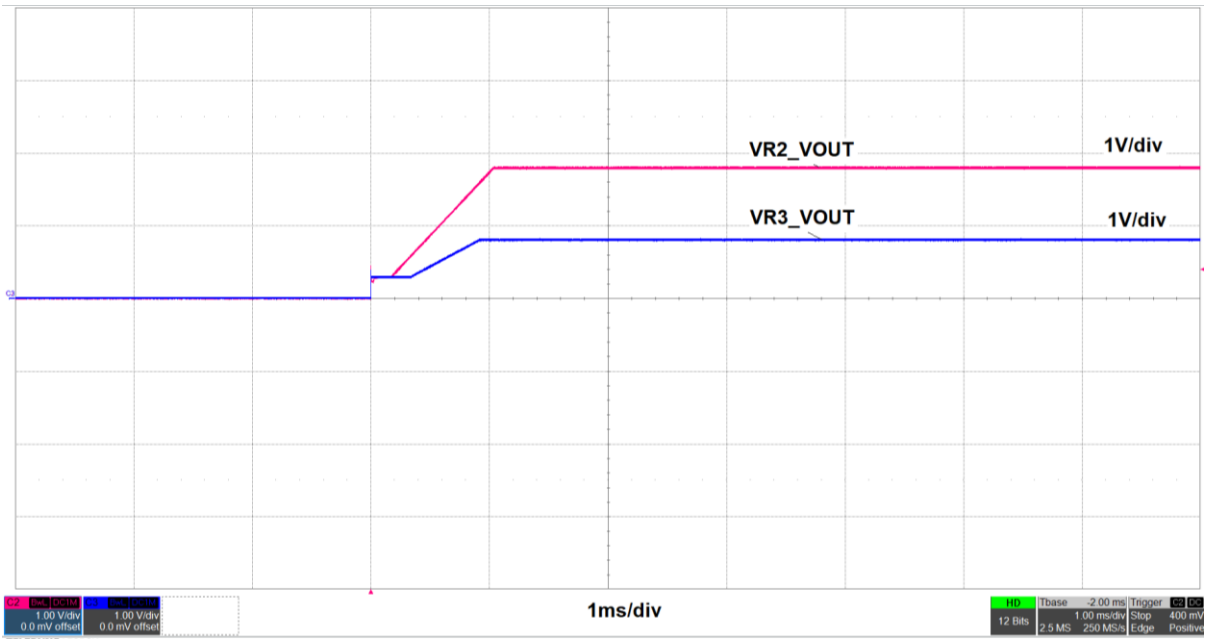

Figure 10. Ratiometric sequencing of rails VR2 and VR3, powering-up with different soft start ramp rates

5. Conclusion

This application note provides details on how the EP71xx family of integrated voltage regulators can address the sequencing requirements of multiple-rail applications in various industry applications.

The highly programmable features of these IVRs can help the user easily customize to their system needs to ensure overall reliable operation.

6. Appendix

6.1 Detailed Description of User-Programmable Registers in EP71xx

6.1.1 VR Enable Registers

EP71xx offers capability to enable each of the output rail (VR) independently through VRn_ENABLE.

Reg Address	Bits	Bit Field Name	Read/ Write	Reset Value	Description
0x20	[7:4]	-	-	-	Not used
	3	VR4_ENABLE	R/W	0x00	VR enable input: Write a 1 to VRn_EN to turn on VRn. Write a 0 to VRn_EN to turn off VRn.
	2	VR3_ENABLE		0x00	
	1	VR2_ENABLE		0x00	
	0	VR1_ENABLE		0x00	

Table 12: VR Enable Register

6.1.2 VR MTP Enable Registers

Reg Address	Bits	Bit Field Name	Read/ Write	Reset Value	Description
0x21	3	MTP_VR4_ENABLE	R/W	0x00	Program this register through MTP (multi-time programming)to force enable VRn by default as soon as chip EN is high. . The new programmed value will take into effect when the device goes through a power cycle (toggle of chip EN pin).
	2	MTP_VR3_ENABLE		0x00	
	1	MTP_VR2_ENABLE		0x00	
	0	MTP_VR1_ENABLE		0x00	

6.1.3 GPIO Functions Register

EP71xx provides two General Purpose Input/Output pins (GPIO0, GPIO1) that can be configured for various functions through the **GPIOx_FUNCTION** registers for selected VR rails defined through the **GPIOx_VRs** registers (listed in **Table 13**).

- **Hardware Enable**: GPIO can be configured as a HW Enable pin for the VRs indicated in the **GPIO_VRs** register.
  Note:
  - It is possible to configure GPIO0 and GPIO1 as HW enables for one rail. In this case, either of the GPIO pins can start the rail.
  - The VR that has been enabled will power up with a default delay of typically 80µs. Additional delay can be included in the startup of the VR through the delay register bits (Table 14).
  - When a GPIO is configured to act as HW Enable for a VRn, the corresponding **VRn_ENABLE** register bit (**Table 12**) is not set to 1. The VRn can be disabled only by pulling the GPIO voltage low or by disabling the device (Chip Enable is disabled or PVIN is pulled low).

Reg Address	Bits	Bit Field Name	Read/ Write	Reset Value (hex)	Function/Description			
					GPIO Setting	FUNCTION	Input/ Output	DESCRIPTION
0x23	[7]	-	-	-	-	-	-	Not used
	[6:4]	GPIO0_FUNCTION	R/W	0x00	000	DVS Function	Input	This bitfield allows the user to set GPIO0 as an input for DVS Function. The logic state of the GPIO pin is fed to the DVSIN signals of each VR included in GPIO_VRS (If a GPIO_VRS bit is configured for both GPIO0 and GPIO1, GPIO0 takes precedence). The LOW state selects the voltage programmed in VRn_DAC_TARGET register and the HIGH state selects the voltage programmed in VRn_DAC_TARGET2 register.
					001	Alt PG	Output	GPIO programmed to function as Power Good. GPIO asserts high when all VRs included in GPIO_VRS are in regulation and output voltage has stabilized.
					010	GPIO Alert	Output	This bitfield allows the GPIOx pins to be used as an output for alert functions for the VRs included in GPIO_VRS. Alert functions for these VRs will cause the GPIO to assert high.
					1xx	HW Enable	Input	Bit 6 sets GPIO as an input to serve as a HW enable for the VRs indicated by GPIO_VRS. It can be configured as a HW Enable for one or all of the VRs.
	[3:0]	GPIO0_VRS			GPIO0 Functions apply to VR4 (Bit 3 is set for VR4)			
					GPIO0 Functions apply to VR3 (Bit 2 is set for VR3)			
					GPIO0 Functions apply to VR2 (Bit 1 is set for VR2)			
					GPIO0 Functions apply to VR1 (LSB is set for VR1)			

Reg Address	Bits	Bit Field Name	Read/ Write	Reset Value (hex)	Function/ Description	Reg Address	Bits	Bit Field Name
0x24	[7]	-	-	-	-	-	-	Not used
	[6:4]	GPIO1_FUNCTION	R/W	0x00	000	DVS Function	Input	This bitfield allows the user to set GPIO0 as an input for DVS Function. The logic state of the GPIO pin is fed to the DVSIN signals of each VR included in GPIO_VRS (If a GPIO_VRS bit is configured for both GPIO0 and GPIO1, GPIO0 takes precedence). The LOW state selects the voltage programmed in VRn_DAC_TARGET register and the HIGH state selects the voltage programmed in VRn_DAC_TARGET2 register.
					001	Alt PG	Output	GPIO programmed to function as Power Good. GPIO asserts high when all VRs included in GPIO_VRS are in regulation and output voltage has stabilized.
					010	GPIO Alert	Output	This bitfield allows the GPIOx pins to be used as an output for alert functions for the VRs included in GPIO_VRS. Alert functions for these VRs will cause the GPIO to assert high.
					1xx	HW Enable	Input	Bit 6 sets GPIO as an input to serve as a HW enable for the VRs indicated by GPIO_VRS. It can be configured as a HW Enable for one or all of the VRs.
	[3:0]	GPIO1_VRS			GPIO0 Functions apply to VR4 (Bit 3 is set for VR4)			
					GPIO0 Functions apply to VR3 (Bit 2 is set for VR3)			
					GPIO0 Functions apply to VR2 (Bit 1 is set for VR2)			

					GPIO0 Functions apply to VR1 (LSB is set for VR1)

Table 13: GPIO Functions Register

6.1.4 VR Power Sequencing

EP71xx offers programmable rail sequencing through which the user can program the order and delay times of the startup and shutdown of the various rails.

Reg Address	Bits	Bit Field Name	Read/Write	Reset Value	Description
0x25 (VR1) 0x27 (VR2) 0x29(VR3) 0x2B(VR4)	7	VRn_DELAY_UP	R/W	0x00	1 = VRn will be enabled after the time delay defined in VRn_DELAY. 0 = The delay defined in VRn_DELAY will not be applied.
	6	VRn_DELAY_DOWN			1 = VRn will be disabled after the time delay defined in VRn_DELAY. 0 = The delay defined in VRn_DELAY will not be applied.
	[5:3]	VRn_START_CONDITION			This rail will start after the rail defined by this register bitfield has been enabled and the output is in regulation.
	[2:0]	VRn_HOLD_CONDITION			This rail will stay enabled until the rail defined by this register bitfield has been disabled.
0x26(VR1) 0x28 (VR2) 0x2A(VR3) 0x2C(VR4)	[7:0]	VRn_DELAY	R/W	0x00	The delay function is programmable and will apply during enabling and disabling of the VRn rail.

Table 14: VR Power Sequencing Register

VRn_START_CONDITION: Through **VRn_START_CONDITION** function it is possible to set a VR rail to startup only after a previous VR rail is OK. A VR rail is considered 'OK' when it has been enabled and its output voltage is in regulation. Refer to Table 15 for register settings to program START conditions for the VR rails. For example: If the startup of rail VR3 is dependent on VR1 and VR2 being ON, **VR3_START_CONDITION** must be set to 011 (binary). Rail VR3 will power up only after VR1 and VR2 rails have been enabled and have reached regulation.

Function	Bit 0 (LSB)	Bit 1	Bit 2 (MSB)
VR1_START_CONDITION	when VR2 OK	when VR3 OK	when VR4 OK
VR2_START_CONDITION	when VR1 OK	when VR3 OK	when VR4 OK
VR3_START_CONDITION	when VR1 OK	when VR2 OK	when VR4 OK
VR4_START_CONDITION	when VR1 OK	when VR2 OK	when VR3 OK

Table 15: Power Rail Sequencing – Programmable START Condition

VRn_HOLD_CONDITION: Through **VRn_HOLD_CONDITION** function, it is possible to set the current active rail to shut down only after a previous rail has disabled. Current rail will stay active/enabled as long as the previous rail stays active. Refer to Table 16 for register settings to program HOLD conditions for the VRs.

For example: If the shutdown of rail VR4 is dependent on VR1 and VR3 being shut down, **VR4_HOLD_CONDITION** must be set to 101 (binary). Rail VR4 will hold the rail ON as long as VR1 and VR3 are Active. VR4 will shut down only after both VR1 and VR3 have shutdown.

Function	Bit 0 (LSB)	Bit 1	Bit 2 (MSB)
VR1_HOLD_CONDITION	VR2 Active	VR3 Active	VR4 Active

VR2_HOLD_CONDITION	VR1 Active	VR3 Active	VR4 Active
VR3_HOLD_CONDITION	VR1 Active	VR2 Active	VR4 Active
VR4_HOLD_CONDITION	VR1 Active	VR2 Active	VR3 Active

Table 16: Power Rail Sequencing – Programmable HOLD Condition

DELAY_FUNCTION: The delay function is programmable through an 8-bit register **VRn_DELAY** (n = 1 to 4 for EP71xx). It defines the time between the enabling of the rail and all the START conditions being met and/or the time between all the HOLD conditions being met and disabling the rail. If **VRn_DELAY_UP** for a rail is set to 1, then the rail will startup only after the delay time as specified in **VRn_DELAY**. The rail will shut down only after the delay time specified in the **VRn_DELAY** register if **VRn_DELAY_DOWN** for the rail is set to 1.

Table 18 shows the settings for **VRn_DELAY** vs expected Delay duration in µs. Note, this doesn't include the default startup delay between I2C_EN/GPIO_EN to VR startup.

The delay for a given part number falls into one of two categories: Delay A or Delay B as shown in Table 17. The default startup delay for Category A is typically 80µs and Category B is typically 40µs.

Part Number	Rate
EP7112	A
EP7123	A
EP7124	B
EP7131	B
EP7136	A
EP7139	B
EP7143	A
EP7144	B

Table 17: Part Number vs Category A/B

VRn_DELAY (code setting)		Delay Duration	Delay Duration
(dec)	(hex)	(Delay A)	(Delay B)
0	0x00	0.6µs	0.3µs
1	0x01	92µs	45µs
5	0x05	460µs	230µs
11	0x0B	1.01ms	0.6ms
22	0x16	2ms	1ms
33	0x21	3.02ms	1.5ms
44	0x2C	4.03ms	2.01ms
55	0x37	5.04ms	2.52ms
66	0x42	6.05ms	3ms
77	0x4D	7.05ms	3.5ms
88	0x58	8.07ms	4ms
99	0x63	9.1ms	4.5ms
109	0x6D	10ms	5ms
164	0xA4	15ms	7.5ms
219	0xDB	20.1ms	10ms
251	0xFB	23ms	11.5ms
255	0xFF	23.4ms	12ms

Table 18: Programmable Delay Settings for VRn_DELAY register

6.1.5 VR Output Voltage Regulators

The output voltage control registers set the target output voltage for the corresponding VR between 0.5V and 2.5V as shown in Table 19.

Reg Address	Bits	Bit Field Name	Read/ Write	Reset Value (hex)	Description
0x41	[9:8]	VR1_DAC_TARGET	R/W	0x199	Sets the VR_n target output voltage. VOUTn = $([VR_n\_DAC\_TARGET]_{DECIMAL}/410)$, where VOUTn is the VRn target output voltage in Volts. Write to VRn_DAC_TARGET to set the VRn output voltage between 0.5V (0xCD) and 2.495V (0x3FF), where n is a number between 1 and 4 representing the VR number (VR1, VR2, VR3 and VR4).
0x42	[7:0]				
0x51	[9:8]	VR2_DAC_TARGET	R/W	0x199	
0x52	[7:0]				
0x61	[9:8]	VR3_DAC_TARGET	R/W	0x199	
0x62	[7:0]				
0x71	[9:8]	VR4_DAC_TARGET	R/W	0x199	
0x72	[7:0]				

Table 19: VR Output Voltage Registers

6.1.6 VR Control Registers

The VR Control register, shown in Table 20, sets various features of VRn. VRn Soft-start ramp rate can be set through this register.

Reg Address	Bits	Bit Field Name	Read/ Write	Reset Value (hex)	Description
0x45	[3:0]	VR1_SOFT_START_RATE	R/W	0x09	The SOFT_START_RATE bit-field sets the VR1 soft-start ramp rate.
0x55	[3:0]	VR2_SOFT_START_RATE	R/W	0x09	The SOFT_START_RATE bit-field sets the VR2 soft-start ramp rate.
0x65	[3:0]	VR3_ SOFT_START_RATE	R/W	0x09	The SOFT_START_RATE bit-field sets the VR3 soft-start ramp rate.
0x75	[3:0]	VR4_ SOFT_START_RATE	R/W	0x09	The SOFT_START_RATE bit-field sets the VR4 soft-start ramp rate.

Table 20: VR Control Registers

VRn_SOFT_START_RATE: This controls the soft-start rate for VR*n* where n is a number between 1 and 4 and represents the VR whose soft-start is being controlled. The soft start ramp sets the rate at which the output voltage rises when the VR is first enabled or is re-starting after a short-circuit condition.

The Soft-start slew rates for a give part number fall into one of two categories: Rate A or Rate B. The table below shows the categories by part number:

Part Number	Rate
EP7112	A
EP7123	A
EP7124	B
EP7131	B
EP7136	A
EP7139	B
EP7143	A

EP7144	B

Table 21: Part Number vs Category A/B

Table 22 shows the register bit field **VR*n*_SOFT_START_RATE** setting vs expected Soft-Start Slew Rate. These rates are captured under no load conditions. The numbers may vary slightly with different load conditions.

VR*n*_SOFT_START_RATE (code setting)	**Soft Start Slew Rate A (mV/µs)**	**Soft Start Slew Rate B (mV/µs)**
0	0.05	0.1
1	0.1	0.2
2	0.22	0.4
3	0.44	0.78
4	0.88	1.54
5	1.77	3.1
6	3.5	6.2
7	7.2	12.3
8	14.6	24.6
9	29	49.6
10	59	100
11	118.2	203.2
12	240	415
13	496	870
14	1110	1870
15	1630	2394

Table 22: Output Soft-start Slew Rate settings

6.2 Multi-time Programming

EP71xx has Non-Volatile Memory for Multi-time programming (MTP). It contains 3 MTP pages to support custom configuration and prototyping. The first MTP page is programmed at the factory with standard configuration. The subsequent two pages are available to the customer to program custom configurations. The last MTP page programmed is the active configuration. The device will operate based on the settings in this active configuration.

Refer to Table 1 for the list of registers that can be programmed into device through MTP to set a desired power-up/down sequence. The new custom configuration is saved for future power-ups and resets.

For further information on steps to perform custom configuration, please refer to the application note '**EP71xx Multi-time Programming (MTP) manual**' or contact the Empower Semiconductor.

7. Revision History

#	Date	Description	Version
1.	Oct 30, 2023	Initial release of AN-EP71xx-1 Rail Sequencing application note	1.0
2.	February 9th, 2024	Formatting updates	1.1