AN14887

Automatic Voltage Scaling using Process, Voltage, and Temperature sensor (PVTS) on i.MX RT700 using ML approach

Rev. 1.0 — 13 January 2026 Application note

Document information

Information	Content
Keywords	AN14887, i.MX RT700, Dynamic Voltage Scaling (DVS), Machine Learning
Abstract	This document describes how the PVTS sensor works using a Machine Learning solution and how to use it in an application to extend battery life.

1 Introduction

The i.MX RT700 is a highly integrated MCU designed for low-power applications such as wearables. It is architected for sustained low-power operation while supporting advanced peripherals and secure applications. It has five independent chip domains (Compute, Sense, Common, DSP, Media). Among these, the Compute and Sense domains serve as the primary processing domains, powered by the digital supply rails VDD2 and VDD1, respectively.

This MCU has many features to reduce and optimize power consumption to extend battery life. The i.MX RT700 supports Dynamic Voltage Scaling (DVS) to optimize the active power with the operating frequencies. This document covers an integrated feature in the i.MX RT700, that, when used properly, can potentially reduce power further and lower the voltage below minimum specifications in the i.MX RT700 Crossover Microcontroller Data Sheet (document IMXRT700EC). This feature monitors the Process, Voltage, and Temperature (PVT) variation of each unique device at runtime. On the i.MX RT700, 2 PVTS are present, one in the compute domain PVTS0 and one in the sense domain PVTS1. Using this PVTS can reduce power on average by 20% on the VDD2 or VDD1 supply rail for the digital logic.

For the compute domain where CPU0 and HiFi4 reside, the digital logic rail is called VDD2. For the sense domain where the CPU1 and HIFI1 reside, the digital logic rail is called VDD1. The PVTS is designed to help reduce voltage and power in active modes, including the i.MX RT700 Sleep mode. The PVTS does not improve power consumption in the static low-power modes and is not used in those modes.

The traditional method of using PVTS involves using a delay value that is programmed into the device's fuses during NXP manufacturing. These delay values correspond to specific frequency operation points for a given domain. This document describes how the PVTS works using an ML model that allows the PVTS to be used for any operating frequency and not just the fixed points supported by the traditional method.

The i.MX RT700 SDK demonstrates how to use the PVTS using the ML approach on the compute using the `mimxrt700evk_dvs_pvt_comp_only_ml_method` demo.

2 Process, Voltage, and Temperature (PVT) variation

The power consumption and the performance of a semiconductor can change during its operation when the temperature or voltage varies. Also, there are differences in comparing one specific device to another due to the variation in the manufacturing process. To optimize power consumption and performance of a device like i.MX RT700, it is important to understand how the silicon is affected by these variations.

2.1 Silicon process variation

Modern semiconductors are of such small geometries today that subtle differences in the manufacturing process can lead to noticeable differences in the key properties of each device. Even on the same wafer, each die can have different properties. One property that varies due to the process is the speed of the transistors. In this document, the variation in the speed of each device's transistors is one-dimensional. The speed of both NFETs and PFETs on a specific device can be at a Slow-Slow (SS) limit, at the Fast-Fast (FF) limit, or somewhere in between. When we plot the distribution of this speed for every device manufactured, we have a Gaussian distribution, see Figure 1. The median of these speeds is Typical-Typical (TT). The testing procedure has limits for what speeds are acceptable for production. Outliers that are slower than the SS limit or faster than the FF limit will be discarded.

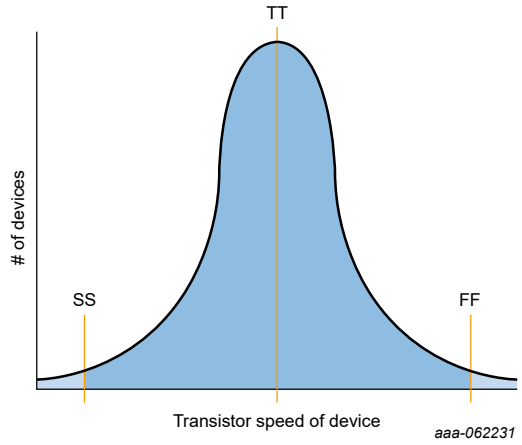

Figure 1. Gaussian distribution of transistor speeds of individual devices

There are tradeoffs in comparing devices with different speeds depending on the process variation: SS devices have lower static leakage current, but FF devices have faster performance when active. At the same clock frequency, FF devices can be powered with a lower voltage than SS devices and still achieve the same performance. Since the SS transistors are slower, they must be turned on harder using a higher supply voltage to keep up with the FF devices.

2.2 Temperature variation

Silicon performance and power consumption vary with temperature. The i.MX RT700 operates within an ambient temperature range of -30°C to 85°C, with a typical value of 30°C.

When temperature decreases, transistor speed slows. For example, at -30°C, transistors operate more slowly than at 30°C. To maintain the same clock frequency at lower temperatures, the supply voltage must be increased to drive the transistors harder. Conversely, as the device warms back to 30°C, the voltage can be reduced while sustaining the same clock frequency.

2.3 Minimum voltage required

i.MX RT700 has the option to power VDD2 and VDD1, respectively the power digital supply of the compute and the sense domain using the internal LDOs or using an external PMIC as the NXP PCA9422 on the i.MX RT700 EVK board. This application note covers the case when the VDD2 and VDD1 rails are powered by the external PMIC, but can be applied when the LDOs are used to power VDD2 and VDD1 as well.

To minimize power consumption in an application that must be clocked at a specific frequency, the supply voltage would be set at the minimum voltage required for that frequency. Increasing the voltage above the minimum required level wastes battery life. The data sheet for the device includes specifications for the minimum voltage. However, these limits are set to guarantee operation of all devices with a wide range of PVT variations.

An example from the i.MX RT700 data sheet is given below.

Table 1 comes from the “General Operating Conditions” table in the data sheet, refer to the data sheet for the latest specifications. For instance, in the Compute domain, when the CPU0 clock, `compute_main_clk` runs at 192 MHz, the supply rail VDD2 requires a minimum voltage of 0.9 V to operate.

Another example in the Sense domain, when the HIFI1 clock, `sense_dsp_clk` runs at 60 MHz, the supply rail VDD1 requires a minimum voltage of 0.8 V to operate.

Table 1. VDD1 and VDD2 operating conditions from data sheet when using an external PMIC

Digital rail	Conditions	Minimum voltage (V)
VDD1	Active mode; CPU1/HIFI1 max freq = 250 MHz	1.1
	Active mode; CPU1/HIFI1 max freq = 205 MHz	1.0
	Active mode; CPU1/HIFI1 max freq = 160 MHz	0.9
	Active mode; CPU1/HIFI1 max freq = 100 MHz	0.8
	Active mode; CPU1/HIFI1 max freq = 45 MHz	0.7
	Deep Sleep mode	0.63
	Deep Sleep mode async	0.5
VDD2	Active mode; CPU0/HIFI4 max freq = 325 MHz	1.1
	Active mode; CPU0/HIFI4 max freq = 250 MHz	1.0
	Active mode; CPU0/HIFI4 max freq = 192 MHz	0.9
	Active mode; CPU0/HIFI4 max freq = 110 MHz	0.8
	Active mode; CPU0/HIFI4 max freq = 60 MHz	0.7
	Deep Sleep mode	0.63
	Deep Sleep mode async	0.5

The 0.9 V for VDD2 and 0.8 V for VDD1 in the previous example are requirements based on worst-case conditions: an SS device running at a temperature of -30 °C. Since most devices manufactured are faster than SS, most devices can clock at 192 MHz for VDD2 or 60 MHz for VDD1 with a voltage less than respectively 0.9 V and 0.8 V. Furthermore, most devices do not spend much time operating at -30 °C and are closer to room temperature, it enables operating at some voltage less than 0.9 V and 0.8V. It is rare that a device requires 0.9 V to operate at 192 MHz for the Compute domain or 0.8V to operate at 60 MHz for the Sense domain.

To be able to operate at the ideal minimum voltage for a given frequency, each unique device must determine that voltage at runtime, while accounting for temperature changes. The integrated PVTS is the solution to this challenge.

3 Operating with PVTS

This section provides information on operating with PVTS.

3.1 PVTS architecture

i.MX RT700 has two PVTS, one in each domain (compute and sense). Each PVTS instance has 2 delay lines providing the ability to monitor 2 clocks in the same power domain in parallel. Therefore, in the Compute domain with the PVTS0, one delay line monitors the CPU0, and the other delay line monitors the HIFI4. In the Sense domain with the PVTS1, one delay line monitors the CPU1, and the other delay line monitors the HIFI1. The two PVTS instances are independent and can be used separately or in parallel.

The PVTS is a peripheral in the i.MX RT700 that monitors the timing margin of a given clock and triggers an interrupt when this timing margin is not met, which indicates VDD2 and/or VDD1 must be increased to meet the required timing. Figure 3 shows a closed loop system using an external PMIC to adjust the voltage to VDD2 and/or VDD1. When conditions change and the PVTS indicate that VDD2 and/or VDD1 must be increased, the application requests the PMIC to increase this voltage through I2C.

PVTS0 is accessible both by the CPU0 and HIFI4, meaning each core can enable and configure its delay line and the PVTS0 instance by its own. The same mechanism exists for the PVTS1 with CPU1 and HIFI1.

However, only the CPU0 is connected to the PVTS0 interrupt, and only the CPU1 is connected to the PVTS1 interrupt. For instance, if the PVTS0 monitors only the HIFI4 while the CPU0 is in low power mode, then the CPU0 must be woken up to take care of the interrupt generated by the delay line monitoring the HIFI4 and potentially increase VDD2 by a PMIC step.

Also, as VDD2 and VDD1 are controlled by the same external PMIC using an LP_FLEXCOMM as I2C, if both PVTS0 and PVTS1 are enabled then a secure way to access the PMIC must be implemented. Indeed, a mutex can be implemented to ensure that the PMIC using I2C is accessed by one core: CPU0 or CPU1 at a time.

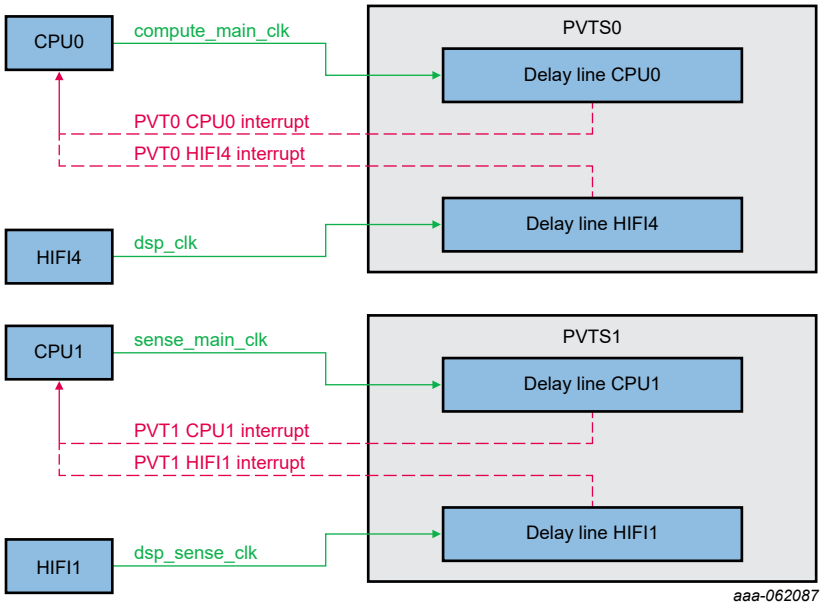

Figure 2. i.MX RT700 PVT architecture

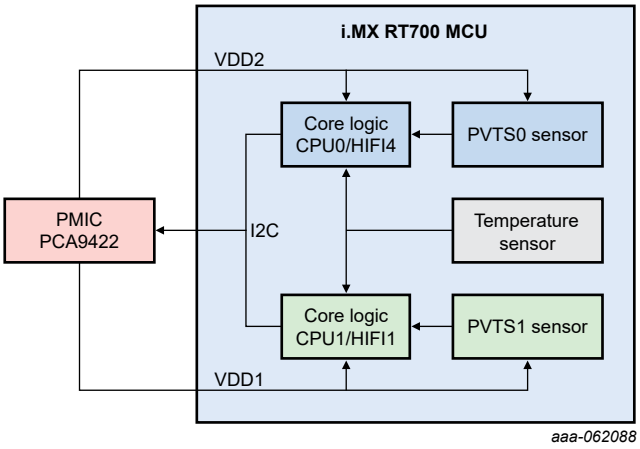

Figure 3. Closed-loop voltage control with PVTS

3.2 Monitor timing margin

In this chapter, CPU0 monitored by the PVTS0 using `compute_main_clk` is used. The same mechanism is used when a PVT instance monitors the CPU1, HIFI1, or HIFI4.

The PVTS monitors the timing margin of `compute_main_clk` using a delay line, see Figure 4. `Compute_main_clk` is divided by two and generates the Launch CLK. The Launch CLK is compared to a delayed version of itself. This delay is programmable and must be set to a value that correlates to the speed of

that unique device (as in SS/TT/FF). The timing comparison is done through a flip-flop that uses Launch CLK to clock the delayed version of itself. If the added delay is too long, the flip-flop output is high, indicating a timing issue and triggering an alarm.

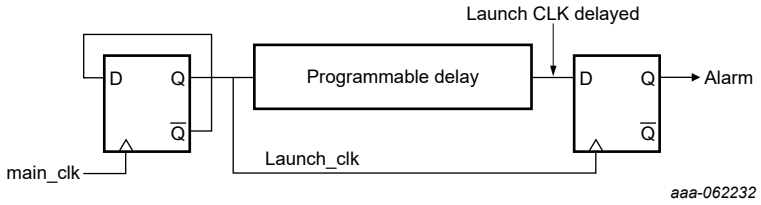

Figure 4. PVT delay line block diagram

To monitor `compute_main_clk` timing, program the PVTS0 delay line to match the worst-case timing path of the logic clocked by `compute_main_clk`.

After programming, the PVTS0 emulates this worst-case path in VDD2 logic. The delay tracks the actual timing across temperature and voltage variations.

- When voltage drops or temperature decreases, the Launch CLK delay increases.
- If the alarm signal triggers, increase the VDD2 voltage. This improves logic performance and reduces Launch CLK delay.

Figure 5 shows the timing of the PVTS0 Delay Line. The PVTS0 Delay Line samples the delayed Launch CLK on every rising edge of Launch CLK.

- When the VDD2 voltage is above the minimum for the device and current temperature, the delay is short. The delayed Launch CLK is low at the rising edge, so the alarm output stays low.
- When the VDD2 voltage is too low, the delay is long. The delayed Launch CLK remains high at the rising edge, and the alarm output goes high. This triggers an interrupt to indicate that the VDD2 voltage must be increased to reduce the timing delay.

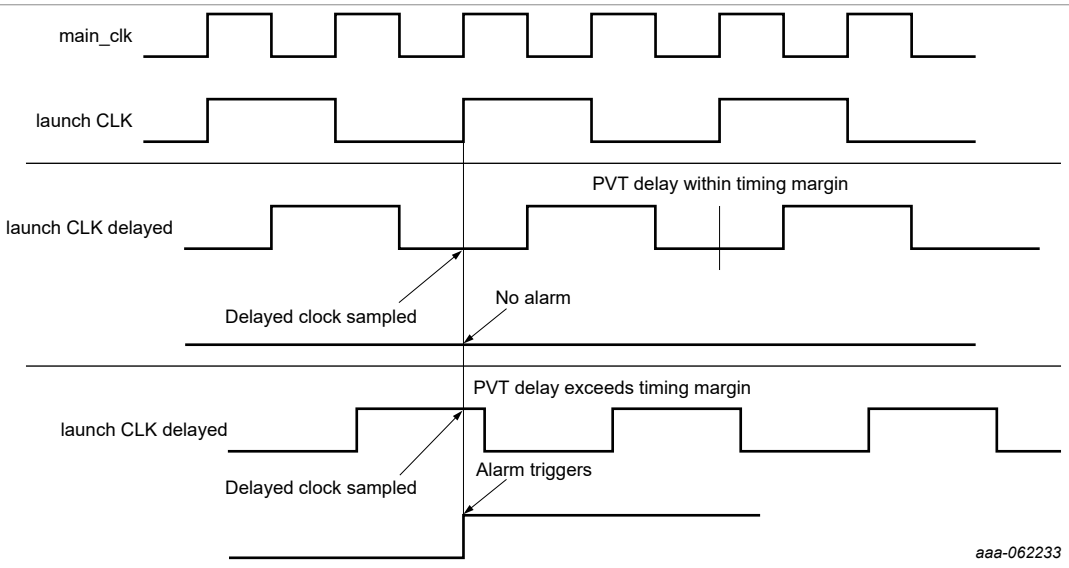

Figure 5. PVT delay line timing with alarm trigger

3.3 Machine Learning method

VF curves, or the minimum operating voltage for a given frequency, are linear. Based on characterized frequencies, it is possible to extrapolate the corresponding linear function to deduce the Vmin at any frequency.

Vint curves when the PVTS triggers the interrupt per characterized frequency follow a different linear regression. For instance, with 8 frequency points:

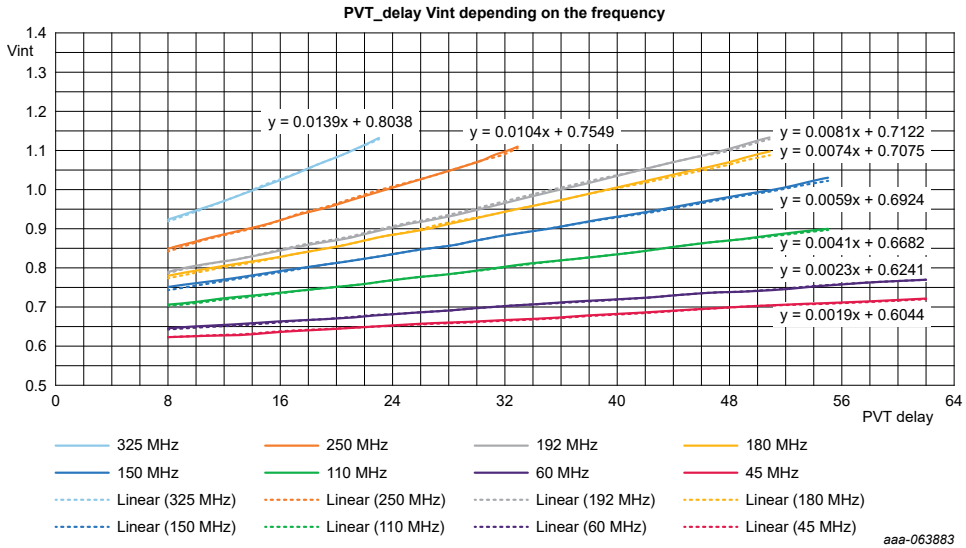

Figure 6. PVT Vint depending on the delay and the frequency

The slope and intercept of each linear regression function follow a 2-order polynomial function.

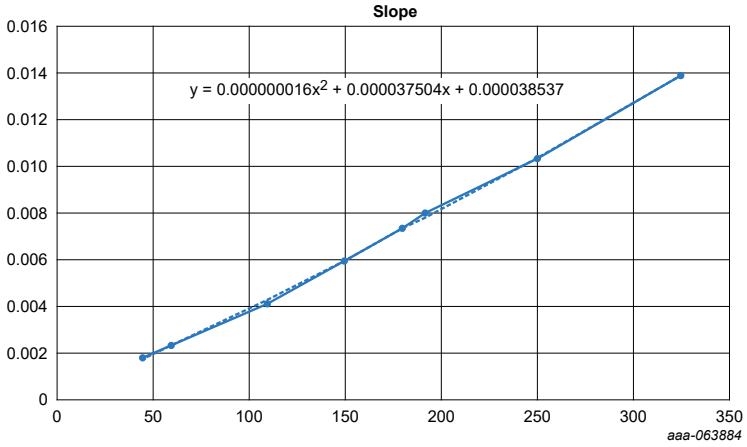

Figure 7. Vint curve slope depending on the frequency

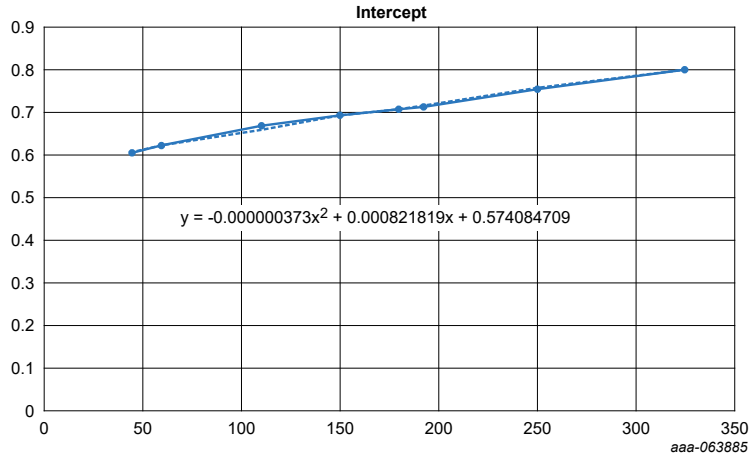

Figure 8. Vint curve intercept depending on the frequency

With Vmin and Vint for characterized frequency points, NXP can predict accurately the behavior at every frequency.

NXP has characterized hundreds of devices across process (SS limit to FF) at worst case temperature (-30C) for the different cores, to create a dataset big enough to be considered generic. This dataset has been used to train and validate the machine learning model.

The developed machine learning model takes the following input parameters:

- Core [CPU0: 1, HIFI4: 2, CPU1: 3, HIFI1: 4],
- Vmin slope; This parameter is read from the OTP fuses provided by NXP,
- Vmin intercept; This parameter is read from the OTP provided by NXP,
- Targeted frequency; Provided by the user,
- Corresponding Vmin; calculated from (slope * targeted frequency) + intercept.

The output of the model is the PVTS delay to use on the specified core at the targeted frequency.

Table 2. Model performance

Dataset	R^2 Score	MAE	Loss
Training	0.9977	0.2961	0.1138
Validation	0.9981	0.2718	0.1015

3.4 Reducing supply voltage at runtime

In this chapter, PVTS0 monitoring the CPU0 at 192MHz will be used. However, the mechanism remains the same for the other cores (CPU1, HIFI1, HIFI4).When using the PVTS0 and reducing VDD2, the PVTS0 alarm indicates when a higher voltage is required. But the PVTS0 does not give any indication of when VDD2 can be further reduced. Instead, the application tests a lower voltage periodically. If that reduced voltage is too low, the PVTS0 immediately triggers the alarm, and the application increases the voltage back to the level it was. But if no alarm is triggered, VDD2 can remain at that reduced voltage.

Figure 9 shows an example of VDD2 dynamically changing as temperature changes. This chart is an example of clocking at 192 MHz, and the VDD2 minimum specification in the data sheet for CPU0 at 192 MHz is 900 mV. This graph starts where VDD2 in blue has already been reduced to 850 mV by using the PVTS0 and is operating at 30C. The PMIC PCA9422 is used, which has voltage steps of 6.25 mV. While the temperature drops, the PVTS0 Delay Line monitoring the CPU0 gets slower, and at some point, the PVTS0 alarm triggers. The application requests the PMIC to increase 1 step to 856.25 mV. The temperature continues to drop, so another PVTS0 alarm triggers, and VDD2 is increased to 862.5 mV. The application is continually testing if a

reduced voltage is acceptable. Typically, reducing the voltage 1 step immediately triggers the PVTS0 interrupt, and the voltage is increased 1 step back to the previous level. As the temperature warms, at some point the reduced voltage test does not trigger an alarm, and the PVTS0 allows VDD2 to reduce 1 step to 856.25 mV. As the temperature continues to warm, this example eventually drops down to 800 mV. And as the temperature then cools back to 30C, the PVTS0 triggers its alarm when voltage must be increased, the VDD2 returns to 850 mV.

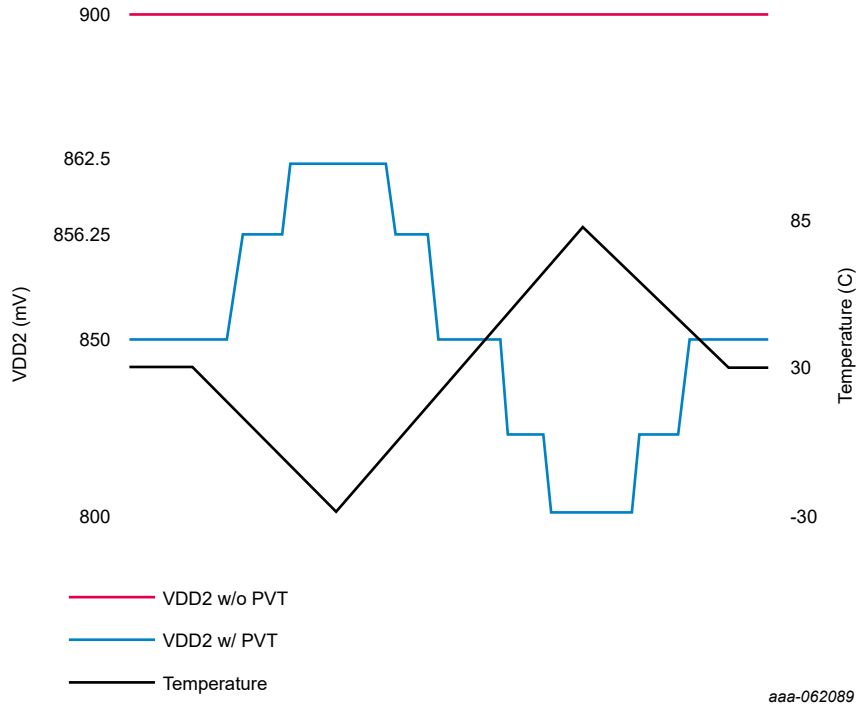

Figure 9. Dynamic VDD2 voltage at runtime

When the application reduces VDD2 using the PVTS, it can test for a lower voltage. This method reduces the voltage 1 step, waits for the voltage to settle, and then repeats. See Figure 10 for an example of this reduction method. When the PVTS0 detects the voltage is too low, it interrupts the application, which raises the voltage 1 step.

Section 2.3 discusses minimum voltage requirements for the PVTS. If the PVTS0 has already reduced the voltage to the minimum allowed by the application, then the application can stop testing for a lower VDD2, until after VDD2 is increased again.

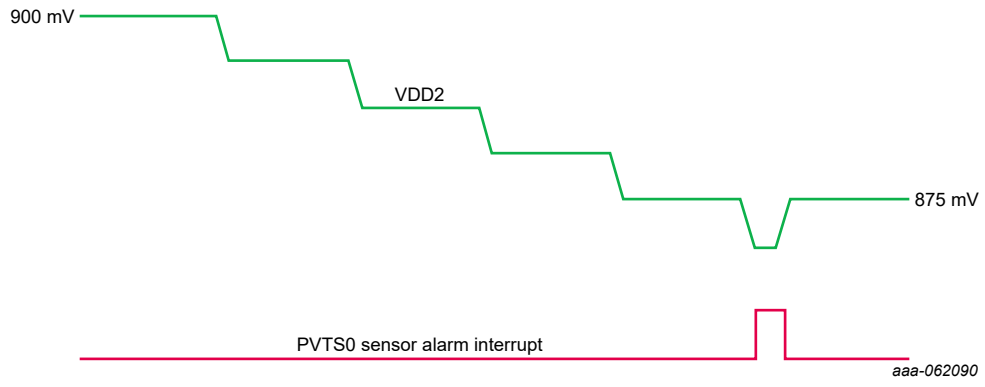

Figure 10. Finding optimal VDD2 voltage with the PVTS0

After initializing the PVTS, the goal is to quickly find the minimum acceptable voltage in the current conditions. With each reduction in voltage, the application must give the PMIC and supply voltage rail enough time to settle at the new voltage. NXP's testing with the PCA9422 PMIC found once the PMIC changes the voltage 1 step, the supply rail settles quickly. But the application needed more time to control the PMIC using I2C and send the new register settings. The time measured for this PMIC communication was about 2 ms from calling the API to update the PMIC, to when the PMIC changed VDD2 voltage. Therefore, with some added headroom, the SDK example uses a 5 ms delay for the PMIC during the initial voltage reduction. This delay can be affected by the source clock of the I2C used to communicate with the external PMIC.

For applications using the PVTS, the initial voltage reduction ends when the PVTS alarm triggers. The application then increases the voltage 1 step, and has now found the minimum acceptable voltage in the present conditions. The example above in Figure 10 assumes clocking at 192 MHz, and starts at the VDD2 safe voltage of 900 mV. The voltage decrements 6.25 mV per step, and waits for the VDD2 voltage to settle. When VDD2 is reduced to 868.75 mV, the PVTS interrupt immediately triggers. The application increases VDD2 by 1 step, and has now found the optimal voltage of 875 mV.

The next stage for the application is to check if the temperature has changed enough that a lower voltage is now acceptable. One method to check this is to periodically reduce the voltage. Based on analyzing the thermal mass of the system, and testing applications with different delay lengths, NXP found that testing for a lower voltage every 10 s is a good option for most applications. This application example uses a task timer to test every 10 s. Other applications can change this test period to optimize. Shorter periods mean that the app spends more time adjusting the PMIC and testing the voltage; the downside of longer periods is that the app may execute at a higher power consumption longer than necessary.

3.5 PVT delays stored in OTP fuses

NXP programs each i.MX RT700 device during manufacturing with the unique characterized Vmin slope and intercept of the device. NXP recommends reading this value using the PVTS driver and API `PVTS_ReadParametersFromOTP()`. PVTS delay can be deduced using the ML model using `PVTS0DeduceDelayUsingML()` and programmed using `PVTS_SetDelay()`.

Starting with a specific manufacturing date code, NXP programs each i.MX RT700 device during manufacturing with the ideal PVT slope and intercept values enabling the ML feature. This starting date code is 2551 (Work Week 51 in 2025).

Table 3. PVT parameters for i.MX RT700

Cores	OTP address	Slope	Intercept
CPU0/HIFI4	452	Bits 0-15	Bits 16-31
CPU1/HIFI1	451		

- Slope and intercept stored in the OTPs are encoded using integer values. However, float values are used for the ML model. Decoding the slope and parameters is required before passing the two parameters as input to the model. This is done in the `PVTS0DeduceDelayUsingML()` function.
- Valid ranges of this PVTS delay value are 2 to 62.
- Be aware of the requirements before reading the fuses in OTP. The data sheet specifies a minimum VDD2 voltage and frequency for reading the OTP, and the application may need to increase VDD2 first. Refer to the data sheet for the latest requirements.
- The OTP driver must be initialized before reading the fuses. `PVTS_ReadParametersFromOTP()` will initialize the driver for the application if the otp_initialized argument is false.
- The MCUXpresso SDK includes an OTP driver example that demonstrates initializing the OTP and reading the fuses.
  On i.MX RT700, the sense domain cannot access the OTP fuses location where the Vmin slope and intercept are stored. Therefore when the PVTS1 is enabled on CPU1 and/or HIFI1, the CPU0 must retrieve the

parameters of the CPU1/HIFI1 in the OTP location 451, execute the ML model with proper parameters and send the output, the PVTS1 delay to the CPU1. Once received, the CPU1 can configure and start the PVTS1. One possible implementation is to use the Messaging Unit to transfer data, in this case the PVT delays, from the CPU0 to the CPU1.

4 Using PVTS in application

The PVTS enables dynamic voltage scaling to optimize power consumption while maintaining timing integrity. However, its integration introduces design and debugging challenges that require careful planning. This section outlines key considerations for safe and efficient PVTS usage.

4.1 Requirements for using the PVTS

Proper usage of the PVTS can allow applications to operate at VDD2 and/or VDD1 levels below data sheet minimum specifications. There are various requirements that must be carefully considered, and followed, for an application to run safely.

VDD2 and VDD1 data sheet minimum voltages are not necessarily limited by compute_main_clk, dsp_clk, sense_main_clk and sense_dsp_clk frequency alone. There are more data sheet restrictions on VDD2 and VDD1 for specific peripherals, features, or use-cases of the MCU that can be found in the data sheet.

The PVTS were designed to monitor the timing margin in the VDD2 and VDD1 domains. However, do not use it when other VDD2 or VDD1 data sheet restrictions are in play.

The following items and discussions present more limitations to consider. Always refer to the data sheet for the full details, latest specifications, and any other requirements.

- If there are multiple VDD2 and/or VDD1 minimum requirements that apply, the highest minimum voltage must be used, regardless of what the PVTS reports.
- Clock sources can have minimum VDD2 and VDD1 voltages requirements based on the trim frequency and dividers used.
- Reading OTP fuses requires a high VDD2 voltage. It is critical when reading the PVT delay value stored in OTP.
- TRNG and ELS cannot be used with PVT. The data sheet minimum voltage for VDD2 must be met when using the TRNG and ELS.

The application can also place maximum VDD2 and/or VDD1 voltage limits used with the PVTS. Consider a use case like this example application: compute_main_clk is 192 MHz clocked from the FRO0. The application is not using other features that require higher VDD2 voltage, the data sheet's "General Operating conditions" minimum specification of 0.9 V applies for 192 MHz. It is the starting voltage for the app before it enables the PVTS0.

Once enabled, the app tries to reduce the voltage. While using the PVTS0, if the voltage increases to 0.9 V and the PVTS0 triggers another alarm, the voltage does not need to be increased. The app can keep VDD2 at 0.9 V because the data sheet confirms this voltage works for all devices and across the full temperature range.

The VDD2 and VDD1 voltages requirements in the data sheet and this document are for the voltages at the VDD2 and VDD1 pins of the MCU. If setting the PMIC to these same minimum VDD2 and VDD1 limits, there is potential for the voltage at the VDD2 and VDD1 pins to be under the requirement, which could lead to issues. For example, if the minimum requirement for VDD2 is 0.8V, and the PMIC is set to supply 0.8V, the voltage at the VDD2 pins could be less than 0.8V. When maximizing battery life and optimizing VDD2 with the PVTS0, the application is accepting more risk by reducing the margin in the system. This can lead to the voltage margin for VDD2 operation down in the millivolts level. Therefore, it is important to consider the power supply system and PCB design when driving VDD2 and VDD1.

The PMIC has a voltage accuracy tolerance for the output. And likely the PMIC output voltage will vary with temperature. Typically, a switching regulator is driving VDD2 and VDD1, so that also introduces ripple on the supply rail. And VDD2/VDD1 voltage could also be lowered in the system before the VDD2/VDD1 pins, like IR drop or due to transients. These impacts and others to the VDD2/VDD1 voltages must be considered during the hardware design and for the PMIC voltage setting at runtime, to ensure that the VDD2/VDD1 voltages requirements are met.

Another requirement already discussed is the settling time for the VDD2/VDD1 power rails. Anytime the application is using the PVTS and reduces the VDD2/VDD1 voltage, it must wait long enough for the PMIC to change the voltage, and the supply rail to settle to the new voltage. The exact delay can depend on the board design, like the PMIC/regulator driving VDD2/VDD1 and the capacitance on this supply rail. If the application drops the voltage too quickly before it has settled, the PVTS cannot detect a timing issue before the voltage is dropped again, and a fault occurs.

One final requirement discussed previously is in a use case where both PVTS0 and PVTS1 are used to monitor clocks in the compute and sense domains. As these domains are powered by respectively VDD2 and VDD1 coming from the same external PMIC, if an interrupt is raised by both PVTSs at the same time, then the application must ensure that only access is done at a time to the PMIC. A mutex or semaphore can be used to ensure exclusive access between the CPU0 and CPU1 to manage PMIC accesses via I2C.

4.2 PVTS considerations with low-power modes

Using the low-power modes with the PVTS can add further complications to the application. In this chapter, low power modes are related to CPU0 and/or CPU1 being in static modes.

For different power mode options, Sleep mode does not really have an impact when using the PVTS. In Sleep mode, the clock to the core is gated, but compute_main_clk or sense_main_clk remains clocked, and the PVTS can continue to operate in Sleep mode and can wake the MCU with its interrupt. Also, waking from ultra-low power modes such as DPD (Deep Power Down) and FDPD (Full Deep Power Down) wakes through the reset process, which likely means the application must start at a safe VDD2/VDD1 voltage and restart the PVTS.

The power mode considered here with the PVTS is Deep Sleep mode. Consider a use case similar to this example application, clocking compute_main_clk at 192 MHz. The MCU is in Active mode using the PVTS. VDD2 originally started at 0.9 V, and started dropping with the PVTS0. The temperature is high and the VDD2 drops to a lower voltage. Now, the application enters the Deep Sleep power mode, which also disables clocks. The logic is in a static state, so the PVTS0 is no longer active. The MCU is in this state for a certain amount of time, and during this state the temperature cools. The worst-case example is when the temperature drops to a minimum -30 °C. Now, the device wakes up, and at these conditions the device needs VDD2 up to 0.9 V before it can clock properly at 192 MHz. If the application wakes up at the current VDD2 level, and attempts to clock at 192 MHz, likely the logic is not going to operate correctly, and a fault occurs. The PVTS0 triggers the alarm immediately after the clock reaches a high frequency with a VDD2 voltage that is too low. However, as the logic is not operating correctly, the firmware cannot execute properly, and the application is not able to request higher voltage from the PMIC.

The PVTS can still be used with low-power modes, but situations like the above example should be avoided. The safest option is to wake back to a safe VDD2 voltage. Revisit the example above, except this time when the MCU wakes, the hardware automatically raises the VDD2 voltage to 0.9 V. This can be done before the application resumes execution, and before the FRO ungates the clock at 192 MHz. The app is able to execute at 192 MHz, and it can re-enable the PVTS0, and reduce the voltage again to an optimized level.

The same mechanism applies to CPU1 in the sense domain monitored by PVTS1 and managing the VDD1 power rail.

The PMIC_MODE pins on the MCU enable the above method to wake up at a pre-configured safe voltage. There are two PMIC_MODE pins, which allow the MCU to request up to four different power configurations in hardware. The PMIC_MODE 0b00 is a special mode because the MCU switches to this mode after POR or when waking from DPD or FDPD modes. The other three PMIC_MODE options are arbitrary, and can be used

as desired by the application. The PMIC can be pre-configured to respond to these mode pins with the desired power configuration. When waking from Deep Sleep mode, the PMIC_MODE pins revert to the state in the PDRUNCFG0[PMIC_MODE] bits.

There are 4 PMIC_MODE instances in the i.MX RT700:

Two in the compute domain, part of the PMC0:

- In the register PMC0->PDRUNCFG0 to control the PMIC_MODE configuration when the CPU0 is active,
- In the register PMC0->PDSLEEPCFG0 to control the PMIC_MODE configuration when the CPU0 is in low power mode.

Two in the sense domain, part of the PMC1:

- In the register PMC1->PDRUNCFG0 to control the PMIC_MODE configuration when the CPU1 is active,
- In the register PMC1->PDSLEEPCFG0 to control the PMIC_MODE configuration when the CPU1 is in low power mode.

The final PMIC_MODE configuration pins value is the aggregated result of the register of PMC0->PDxxCFG0 and PMC1->PDxxCFG0 depending on the state of CPU0 and CPU1. For more details, refer to the Reference Manual PMC chapter.

For each PMIC_MODE configuration, different programmable output voltages are available. Using the i.MX RT700 EVK with PCA9422, SW1 is connected to VDD2 and SW3 is connected to VDD1. Therefore, the proper voltage should be programmed in the correct output voltage setting.

The downside of the safe voltage method is that the application must first raise VDD2/VDD1 to a safe voltage level before entering Deep Sleep. This process is not the most energy efficient. Some applications can wake and resume execution at the same reduced voltage, but it depends on the use case of each application. When the MCU is in Deep Sleep mode, temperature change is the main condition that can cause an issue with VDD2/VDD1 voltage during wake. Applications where ambient temperature does not change much, can wake to the previous voltage found with the PVTS. Some applications can have large temperature changes but wake up fast enough relative to the slower temperature change. In those cases, the temperature cannot have a chance to change much during the short Deep Sleep duration. If an application chooses to wake with a reduced voltage, these limitations and conditions must be well understood and considered.

The application can measure the temperature with the integrated temperature sensor and compare it with a stored temperature from before sleeping. Or a better option is to use a hardware timer and measure its duration in Deep Sleep. Either way, if the temperature change or sleep duration is large enough, the application increases VDD2/VDD1 to a safe voltage first, before switching to the fast clock frequency. If the application determines that the change is minimal compared to conditions before entering Deep Sleep, it can resume at the reduced voltage found by the PVTS before sleeping.

4.3 Potential implementation

The i.MX RT700 architecture allows independent control of Compute and Sense domains, and PVTS instances operate independently. For example:

- Sense domain can enter Deep Sleep while CPU0 remains in Sleep and HiFi4 runs, monitored by PVTS0.
- Both domains can be active with CPU0 and CPU1 running, monitored by PVTS0 and PVTS1 respectively.

As a general guideline, here are some elements to consider while architecting the application:

- OTP fuses are accessible only by the CPU0 and/or HIFI4. If the CPU1 or HIFI1 PVTS1 in the sense domain is enabled, use the Messaging Unit to transfer the PVT delays values between domains.
- The PVTS configuration and start can be done by all cores in their respective domain, in other words, CPU0, HIFI4 can configure PVTS0, and CPU1 and HIFI1 can configure PVTS1. As the same PVTS is used to monitor the CPU0 and HIFI4, PVTS0 and related delay lines can be programmed by the CPU0, or HIFI4, or both depending on the application. The same concept applies to CPU1 and HIFI1 with PVTS1.

- The PVTS interrupts are connected only to the CPUs cores. Both interrupts related to the two delays lines of the same PVTS must be handled by the CPU core.
- When the two PVTS0 and PVTS1 are used in parallel, implement a mutex or semaphore to secure the access to the PMIC via I2C while modifying the digital rail.
- When in a domain only the Hi-Fi is enabled and monitored, for instance the HIFI4 monitored by the PVTS0 in the compute domain, the related CPU (CPU0) must be in sleep mode to allow the PVT to run properly.

4.4 Configuring low-voltage detectors (LVDs)

The i.MX RT700 integrates low-voltage detectors (LVDs) that monitor separately the VDD2 and VDD1 supply rails. For VDD2, the low voltage detection threshold is programmable using PMCx->PD*CFG0[LDO2_VSEL] and PMCx->LVDVDD2CTRL. For VDD1, the low voltage detection threshold is programmable using PMCx->PD*CFG0[LDO1_VSEL] and PMCx->LVDVDD1CTRL.

LVDVDDxCTL register sets the LVD falling trip voltage for 4 levels, with the falling trip = 0.5 +10 mV * value of the field.

LDOx_VSEL selects one of the 4 LVD trip points in LVDVDDx_CTRL.

These registers are present in the PMC0 and PMC1 in the compute and sense domain, meaning that the result is the aggregated value of these registers.

Key considerations for LVD with DFVS and the PVTS

The LVD provides two primary functions as follows:

- LVD can trigger when VDD2/VDD1 falls below the threshold. It can interrupt the application as an early warning, or it can reset the MCU.
- On initial power up and on wake-up from Deep Sleep, the LVD determines when VDD2/VDD1 is high enough to allow the MCU to operate.

The LVDs minimum falling threshold is 500 mV, and the application can increment in 10 mV steps using LVDVDDxCTL with a maximum of 1.13V. The rising threshold is 30 mV above the falling threshold for hysteresis. However, there are tolerances for the threshold across the PVT variation. It means that each LVDVDDxCTL setting has ranges for each threshold, shown in Table 3. This table shows the LVD threshold ranges for a level [LVL3], but it is the same for all 4 levels [LVL0] [LVL1] [LVL2] [LVL3] in LVDVDDxCTL.

Table 4. i.MXRT 700 LVDs threshold ranges

LVDVDDxCTL	Falling voltage (mV)		Rising voltage (mV)	
	Min	Max	Min	Max
0	483	517	512	548
1	493	527	522	558
2	503	537	532	568
3	512	548	542	578
4	522	558	552	588
5	532	568	561	599
6	542	578	571	609
7	552	588	581	619
8	561	599	591	629
9	571	609	601	639
10	581	619	610	650

Table 4. i.MXRT 700 LVDs threshold ranges*...continued*

LVDVDDxCTL	Falling voltage (mV)		Rising voltage (mV)	
	Min	Max	Min	Max
11	591	629	620	660
12	601	639	630	670
13	610	650	640	680
14	620	660	650	690
15	630	670	659	701
16	640	680	669	711
17	650	690	679	721
18	659	701	689	731
19	669	711	699	741
20	679	721	708	752
21	689	731	718	762
22	699	741	728	772
23	708	752	738	782
24	718	762	748	792
25	728	772	757	803
26	738	782	767	813
27	748	792	777	823
28	757	803	787	833
29	767	813	797	843
30	777	823	806	854
31	787	833	916	864
32	797	843	826	874
33	806	854	836	884
34	816	864	846	894
35	826	874	855	905
36	836	884	865	915
37	846	894	875	925
38	855	905	885	935
39	865	915	895	945
40	875	925	904	956
41	885	935	914	966
42	895	945	924	976
43	904	956	934	986
44	914	966	944	996
45	924	976	953	1007

Table 4. i.MXRT 700 LVDs threshold ranges*...continued*

LVDVDDxCTL	Falling voltage (mV)		Rising voltage (mV)	
	Min	Max	Min	Max
46	934	986	963	1017
47	944	996	973	1027
48	953	1007	983	1037
49	963	1017	993	1047
50	973	1027	1002	1058
51	983	1037	1012	1068
52	993	1047	1022	1078
53	1002	1058	1032	1088
54	1012	1068	1042	1098
55	1022	1078	1051	1109
56	1032	1088	1061	1119
57	1042	1098	1071	1129
58	1051	1109	1081	1139
59	1061	1119	1091	1149
60	1071	1129	1100	1160
61	1081	1139	1110	1170
62	1091	1149	1120	1180
63	1100	1160	1130	1190

The LVD interrupt and reset are optional. In Active mode, the application can disable these features and ignore the LVD. If the application uses the LVD for its monitoring feature while dynamically changing the VDD2/VDD1 voltages, the LVDs threshold must be configured. One option is to consider the minimum VDD2/VDD1 voltages allowed by the application for all devices; 800 mV is a good example. The application can set LVDVDDxCTL to the falling threshold range that is below the minimum voltage at the pins of the rail. Another option for applications is to adjust the LVDs threshold when changing VDD2/VDD1. Therefore, keeping them more tightly aligned.

If using Deep Sleep mode and reducing VDD2/VDD1, the LVDs have another considerations. First, if the application lowers VDD2/VDD1 below the LVD falling threshold, then the LVD reset, and interrupt must be disabled. The LVD rising threshold must also be set above the minimum VDD2/VDD1 voltage needed at wake-up. And this LVD value must be set before entering Deep Sleep. The example of waking to a safe voltage, as discussed in Section 4.2 considers a minimum of 900 mV for VDD2. To guarantee the voltage at the VDD2 pins is at least 900 mV before the wake-up process starts, LVDVDD2CTL = 40 can be used. However, the higher limit with option 40 is 956 mV. It requires the PMIC to drive VDD2 to at least 956 mV to ensure that the MCU wakes. After waking, the application can reduce LVDVDD2CTL and lower VDD2.

If the application does not use the LVDs, they can be disabled. Use PMC0->CTRL[LVD2RE] to disable the LVD for the VDD2. Use PMC0->CTRL[LVD1RE] to disable the LVD for the VDD1. The CTRL register can only be written by the PMC0 in the Compute domain. The CTRL register in the PMC1 in the Sense domain is read-only.

4.5 Troubleshooting

Using the PVTS can greatly benefit the power consumption. But it can create some challenges when diagnosing issues in the application. As discussed in this document, there are several requirements and limitations to consider when reducing VDD2/VDD1 with the PVTS. When issues occur related to the PVTS, it is usually because the VDD2/VDD1 voltages are too low for the current conditions and clock frequencies.

Common symptoms when the logic cannot keep up with the clock frequency include the following:

- Application appearing to lockup or be unresponsive
- Erratic and unpredictable faults that are hard to replicate consistently
- Problems only occurring on a small percentage of devices because they are SS or FF devices
- Problems only occurring at certain temperatures
- Issues that happen while waking from Deep Sleep

If issues like these occur and the PVTS is used in the application, the first step is to disable the PVTS. Keep VDD2/VDD1 at a fixed safe voltage. As it is during debugging, the maximum 1.155 V can be used. If the issue goes away, then it is likely related to VDD2/VDD1 going too low at some point. From there, the application can be debugged to track down the point where the voltage rail is too low.

If using the PVTS with low-power modes, try testing by disabling entry into the low-power modes to see if the issue goes away. If so, the issue is likely related to the entry/exit of a low-power mode. If the application is not waking to a safe voltage level, try testing that method to rule out VDD2/VDD1 during wake-up.

The LVDs can also be related to issues with using DVFS or Deep Sleep mode. If unexpected resets are occurring, LVDs can trigger them. While debugging, try disabling the LVDs reset to see if it impacts the issue. Potentially, the LVDs falling threshold is set too high relative to the rail supplied, and occasionally it triggers unwanted resets. However, consider that if the LVDs are triggering, perhaps the issue is that VDD2/VDD1 is dropping lower than expected.

If issues are related to using Deep Sleep mode, the LVDs can also be involved. For debugging, try disabling the LVDs reset and interrupt before entering Deep Sleep mode. The issue can also be related to VDD2/VDD1 and LVDs during wake-up. To rule out issues, increase LVDVDDxCTL to a high setting that ensures VDD2/VDD1 are at a high and safe voltage before MCU starts the wake-up process. Then, ensure that the PMIC is increasing VDD2/VDD1 above the upper limit of the LVDs rising threshold to trigger a wake-up.

Another method that can be helpful is using some output pins to help capture the state of the MCU during the application. The CLKOUT_VDD2 and CLKOUT_VDD1 signals can be connected to a pin, and compute_main_clk (or a divided down main_clk, or other clocks) and/or sense_main_clk can be measured. Also, a GPIO can be used to track entry/exit of low-power modes. The CLKOUT, wake GPIO, PMIC_MODE pins, and VDD2/VDD1 voltages can be monitored with test tools. It can help identify when VDD2/VDD1 have issues during wake-up or with changing clock frequencies. When measuring VDD2/VDD1 voltages, try to measure as close to the MCU's VDD2/VDD1 pins as possible, for example, on the bypass capacitors. Measuring the output at the PMIC cannot expose how the VDD2/VDD1 pins are supplied.

5 Conclusion

The PVTS integrated in the i.MX RT700 improves battery life in power-sensitive applications. Compliance with the operating requirements outlined in this document is critical to avoid issues and maximize performance. NXP has developed an enhanced method leveraging the usage of machine learning to enable the PVTS on every frequency, beyond those supported by traditional PVTS techniques. The power consumption on the VDD2 and/or VDD1 supply rails decreased on average 20% depending on the use case compared to devices running at the voltage required by the data sheet and the targeted frequency. The resulting battery life benefits can be substantial. For applications where battery longevity is a priority, consider implementing the PVTS to optimize efficiency.

6 Acronyms and abbreviations

Table 5. Acronyms and abbreviations

Term	Meaning
PVT	Process, Voltage, Temperature
MCU	Microcontroller
AVS	Automatic Voltage Scaling
DSP	Digital Signal Processor
PMIC	Power management IC
SDK	Software Development Kit
I2C	Inter-Integrated Circuit
ADC	Analog-to-digital converter
OTP	One-time programmable
API	Application programming interface
FRO	Free-running oscillator
IR drop	I*R drop, a voltage drop that occurs across a resistance
POR	Power-on reset
DPD	Deep Power Down
FDPD	Full Deep Power Down
LVD	Low-voltage detect
DVFS	Dynamic Voltage Frequency Scaling

7 Revision history

Table 6. Revision history

Document ID	Release date	Description
AN14887 v.1.0	13 January 2026	Initial version

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Microsoft, Azure, and ThreadX — are trademarks of the Microsoft group of companies.

Automatic Voltage Scaling using Process, Voltage, and Temperature sensor (PVTS) on i.MX RT700 using ML approach

Contents