Flash Corruption: Software Bug or Supply Voltage Fault?

Shyam Chandra, Lattice Semiconductor

Answer: Both! Flash memory is commonly used to store firmware in embedded systems. Occasionally, the firmware stored in the Flash memory in some systems is accidentally corrupted, preventing the system from booting up after power-on. Flash corruption is commonly associated with a software bug. However, it is also commonly understood that the probability of Flash corruption increases either during power cycling tests or during margining tests. The Flash corruption problem tends to be more severe when the number of complex ASICs or SOCs used on the board increases. This article examines Flash corruption and its causes beyond a software bug, and suggests methods to minimize the corruption.

How Do The Flash Memory Contents Become Corrupted?

Figure 1 illustrates a typical circuit board's CPU circuitry. When the power is turned on, the reset generator first activates the CPU reset signal. It then waits until the power to the CPU, Flash memory and the DDR memory each reaches its correct operation level, waits for an additional extended period of time (about 150ms) and then deactivates the CPU reset signal. When the reset signal is deactivated, the CPU begins to execute the initialization routine in the Flash memory, transfers the contents of the firmware stored in the Flash memory into the DDR memory and then executes the program from the DDR memory.

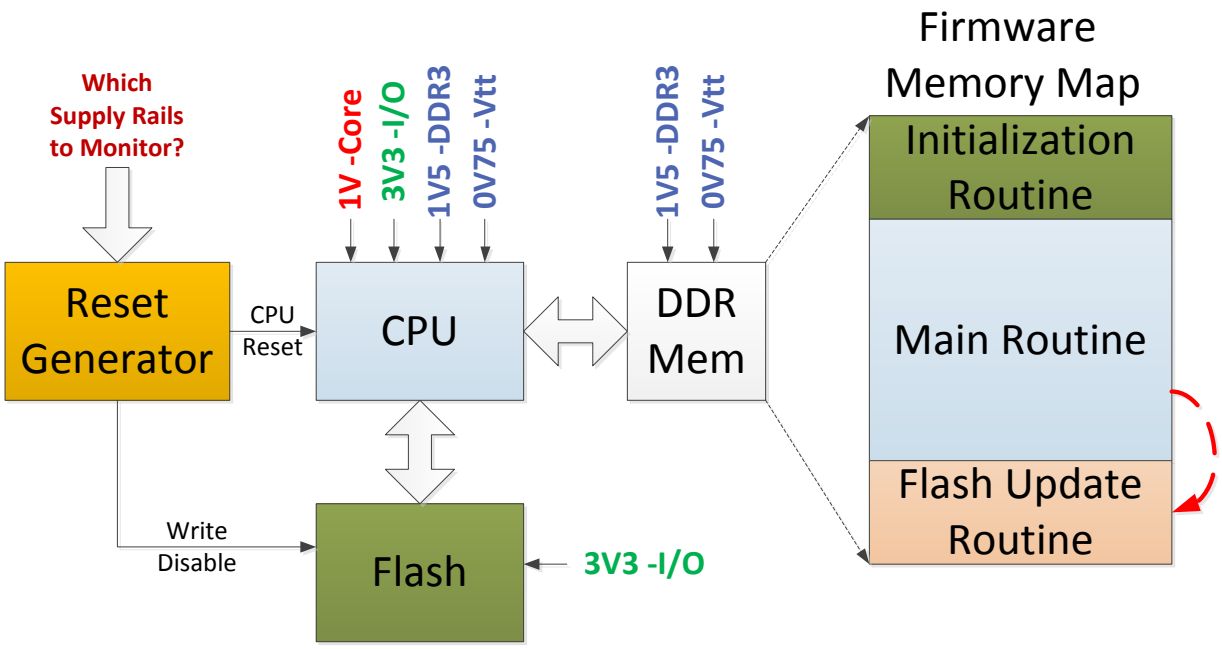

Figure 1 – Typical CPU Section and Firmware Memory Map

The procedure to load firmware into the Flash memory is:

- Firmware is downloaded into the DDR memory through a communication interface.
- Jump to the Flash Update Routine to reprogram the Flash with the new firmware.
- Power to the processor is recycled and the new firmware takes effect.

The Flash memory contents can become corrupted if the code execution jumps to the Flash Update Routine inadvertently. When the board power is cycled, the corrupt version of the code is loaded into the DDR and the board does not function as expected.

The code execution could jump to this Flash Update Routine inadvertently due either to a software bug or to a faulty supply voltage rail (during the power-off event, for example). A software bug can be detected using normal debugging methods. However, a faulty power supply voltage is hard to detect, as the supply voltage error can occur anywhere.

How Does A Supply Voltage Fault Cause The Program To Jump To The Flash Update Routine?

All ICs have both minimum and maximum operating voltage specifications. If the maximum voltage specification is exceeded, the device is damaged, and if the supply drops below the minimum supply level, the device no longer operates as specified. For example, the core voltage specification of the CPU in Figure 1 is 1.2V +/- 5%. If the

voltage drops below this level, the ability of the CPU's internal instruction execution pipeline to reliably transfer instructions and data is compromised, and (depending on the CPU's process and operating temperature) the instruction can be incorrectly executed. For example, a "Move" instruction can be interpreted as a "Pop" instruction, and the code execution then jumps to a random memory location (determined by the contents of the stack). Depending on the contents of that memory location and the error in execution, the processor can either hang or jump to the Flash Update Routine, corrupting the Flash memory and overwriting the Flash memory contents.

A droop in DDR memory voltage or threshold voltage also introduces errors in the instructions and data transferred between the memory and CPU. This erroneous code execution can cause a jump to the Flash Update Routine, corrupting Flash memory.

When does the supply voltage droop?

The power supply voltage droop can occur for the following reasons:

- **Card power down** – When the power to the board is turned off, not all supplies on the board turn off at the same time, because the turn off rate depends on the supply capacity, load, output capacitor, etc. Because the power supply turn-off slew rate is very slow compared to the processor's instruction execution speed, the processor can experience a supply fault, causing it to mis-execute instructions before the supply is fully turned off or before its reset signal is activated.
- **Momentary ground voltage rise**– The power consumption of some processors can fluctuate dynamically, depending on the executed instructions. When such changes occur, the device draws large amounts of current for brief periods from the power source, and dumps these into the ground. As a result, the supply voltage can momentarily droop and the ground voltage may increase. The duration of such a condition depends on the inductance of the supply path.

How can Flash corruption due to supply voltage faults be minimized?

The probability of Flash corruption can be minimized by activating the CPU reset when any supply rail drops below its threshold level. This prevents code execution under faulty power supply conditions. The reset generator activates both the CPU reset signal as well as the write protection signal to the Flash memory. In some cases, the reset generator output is not directly used to reset the CPU. Instead, it is connected to a CPLD, which executes a reset distribution algorithm. In such cases, the write protection signal for the Flash should be activated because the CPU may not be reset as soon as the power supply voltage becomes faulty. The reset generator IC in Figure 1 monitors all CPU rails – 1.0V, 3.3V, 1.5V and 0.75V – and activates the reset signal and Flash write protect signals when any one of them drops below their operating threshold levels.

Selecting a Reset Generator IC

The criteria for selecting a reset IC include the number of voltage monitoring inputs, glitch filtering, hysteresis, fault detection accuracy (across operating temperature and voltage) and fault detection speed.

Number of voltage monitoring inputs: The reset generator IC must monitor all voltage rails related to the CPU for faults (voltage excursions below corresponding operating threshold levels). In the case of Figure 1, four inputs are required with thresholds set at 5% below the nominal operating voltage levels. For example, Lattice power management ICs support 6 to 12 voltage rail monitoring inputs, and the reset generation threshold levels can be programmed from -.5% to -20%.

Errors to avoid: Some designs use a single rail reset generator that usually monitors only, for example, 3.3V. This will not be sufficient, because the 3.3V rail may turn off at a different rate than the core voltage or the DDR voltage. This arrangement could work only if all supplies used the 3.3V as their input supply. In most circuit boards, however, the power supply for the core and DDR use different input voltage sources (due to power dissipation), and so reset generation using only 3.3V cannot avoid Flash corruption. The same argument is true if the reset generator monitors only the core supply rail.

Glitch filtering – When the reset generator has single ended sensing (as opposed to differential sensing) of voltage rails, differences in the ground voltage between the reset IC and the CPU memory can generate false reset signals. To make sure that the reset is actually generated by a fault in the supply voltage, and not by a momentary ground voltage difference, glitch filters are used within the reset ICs. For example, when their input glitch filters are enabled, Lattice power management ICs ensure that the fault persists for 64 microseconds before activating the reset signal.

Errors to avoid: Reset generators using ADC and microcontrollers to monitor voltages implement an ADC sample-averaging algorithm to eliminate the effects of glitches, resulting in false reset activation. The averaging algorithm derives the actual ADC voltage by calculating the average of four ADC voltage samples.

Hysteresis – Most voltage rails are sourced from switched mode power supplies. The output of these supplies usually contain ripple. This ripple can cause a reset signal glitch when the supply level is close to the reset threshold. To avoid this, reset generators must have hysteresis voltage levels ranging from 0.5% to 1% (of the voltage monitored). For example, the hysteresis of Lattice power management ICs is 1% of the monitored voltage. This means that to achieve 1% hysteresis at 3.3V, the hysteresis step size is 30mV, while the hysteresis step size for a monitoring threshold of 1V is 10mV.

Errors to avoid: Reset generators using ADC and microcontrollers to monitor voltages should implement hysteresis in software to prevent glitches in the reset output.

Fault detection Accuracy: *Assumptions: The lowest operating voltage of the CPU is V_{NOM}-5%, where V_{NOM} is the nominal core voltage. When the supply is turned off, V_{NOM} reduces linearly at a rate of 2% per millisecond.*

The accuracy of a reset generator is a measure of uncertainty in its ability to detect a given voltage threshold. For example, a reset generator monitoring Vnom-5% threshold with an error of 2% can activate the reset output anywhere between Vnom-3% (=-5%+2%) to Vnom-7% (=-5%-2%). The processor continues to execute instructions until the reset signal is activated. Figure 2 shows two reset generators (zero propagation delay assumed): one with an accuracy of 2% accuracy (Figure 2a) and the other with 0.7% (Figure 2b). As can be seen, the supervisor with 2% error has a much wider uncertainty range than that of the reset generator with 0.7% accuracy. While the reset output activation within the orange zone prevents the processor from executing even though the supply is healthy (an irritant), the activation in the red zone is its inability to prevent the processor from corrupting the Flash memory. Clearly, the narrower the uncertainty, the less probability of Flash corruption. The accuracy of Lattice power management devices is 0.7%.

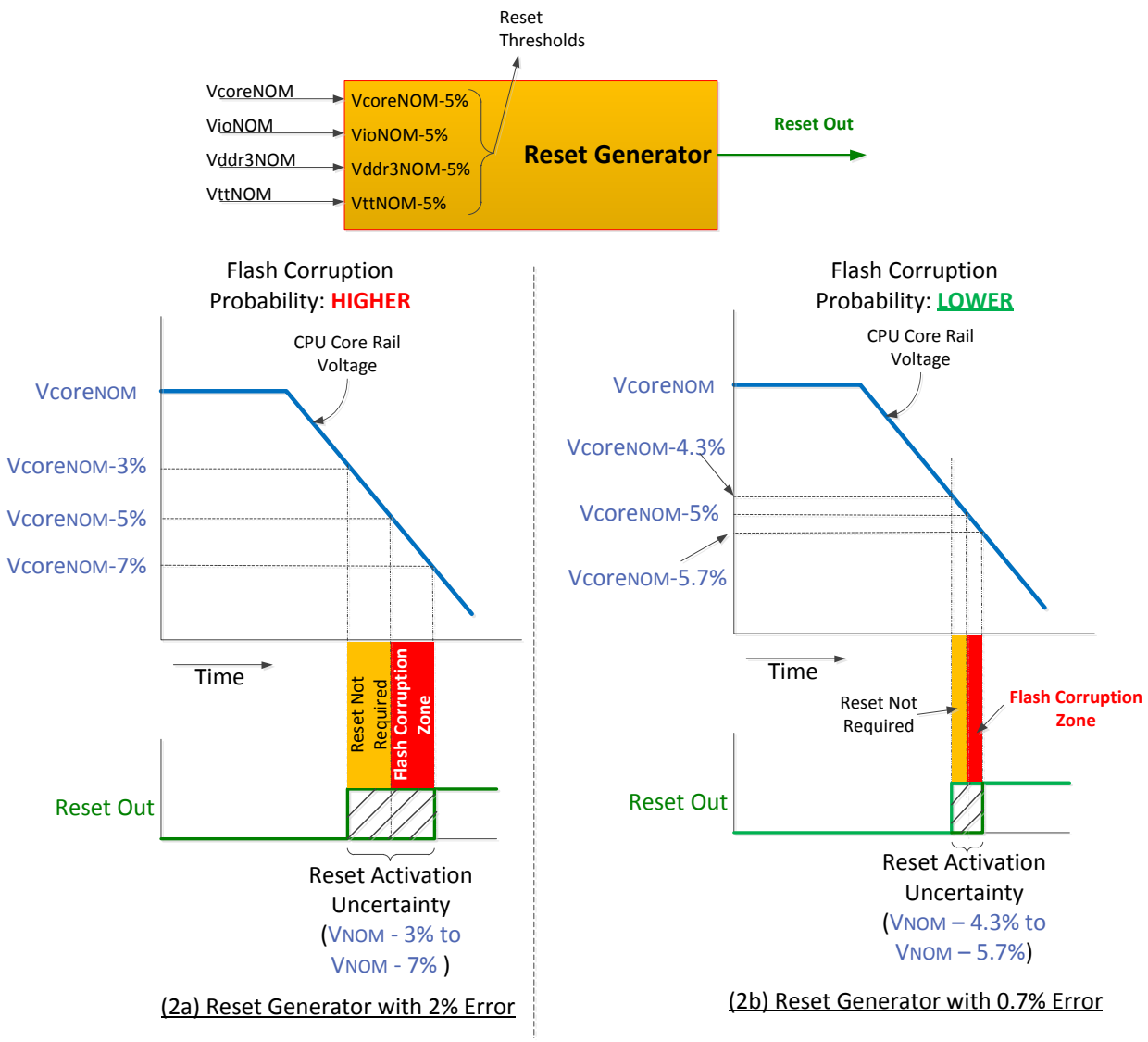

Figure 2 –Flash Corruption Probability vs. Reset Generator Accuracy

Errors to avoid: Some designs use the Power Good from DC-DC converters to determine the health of supplies and use a CPLD to generate reset signals. This method does not reduce the probability of Flash corruption because the accuracy of the Power Good signals of DC-DC converters ranges from 4% to 20%.

Also**,** some designs use low cost comparators to monitor the voltages. In this case, one has to pay attention to voltage reference, resistor accuracy and comparator offset errors. For example, for a 1% accuracy voltage monitoring a circuit across voltage and temperature, one has to use a comparator with <1mV offset error, Vref with an accuracy <0.5% and use 0.1% resistors to set the fault detection threshold.

Fault Detection Speed (Tpd): *Assumption: the voltage monitoring accuracy of the reset generator is 0%.*

Fault detection speed is a measure of the time required for the reset generator to activate the reset output signal from the time the voltage crosses the fault threshold or the reset generator's fault propagation time delay. In Figure 3a, the reset generator requires 1ms to activate the reset signal. The voltage continues to droop and, by the time the reset signal is active, the supply voltage at the CPU is 7% below its nominal operating voltage. This allows about 1ms for the CPU to corrupt the Flash. When the fault detection speed is less than 100us, the voltage at the CPU is V_{NOM}-0.2%, and the probability of Flash corruption is exponentially reduced. Lattice power management devices are able to activate the reset signal in about 64us with the glitch filter turned on, or 16us with the glitch filter turned off.

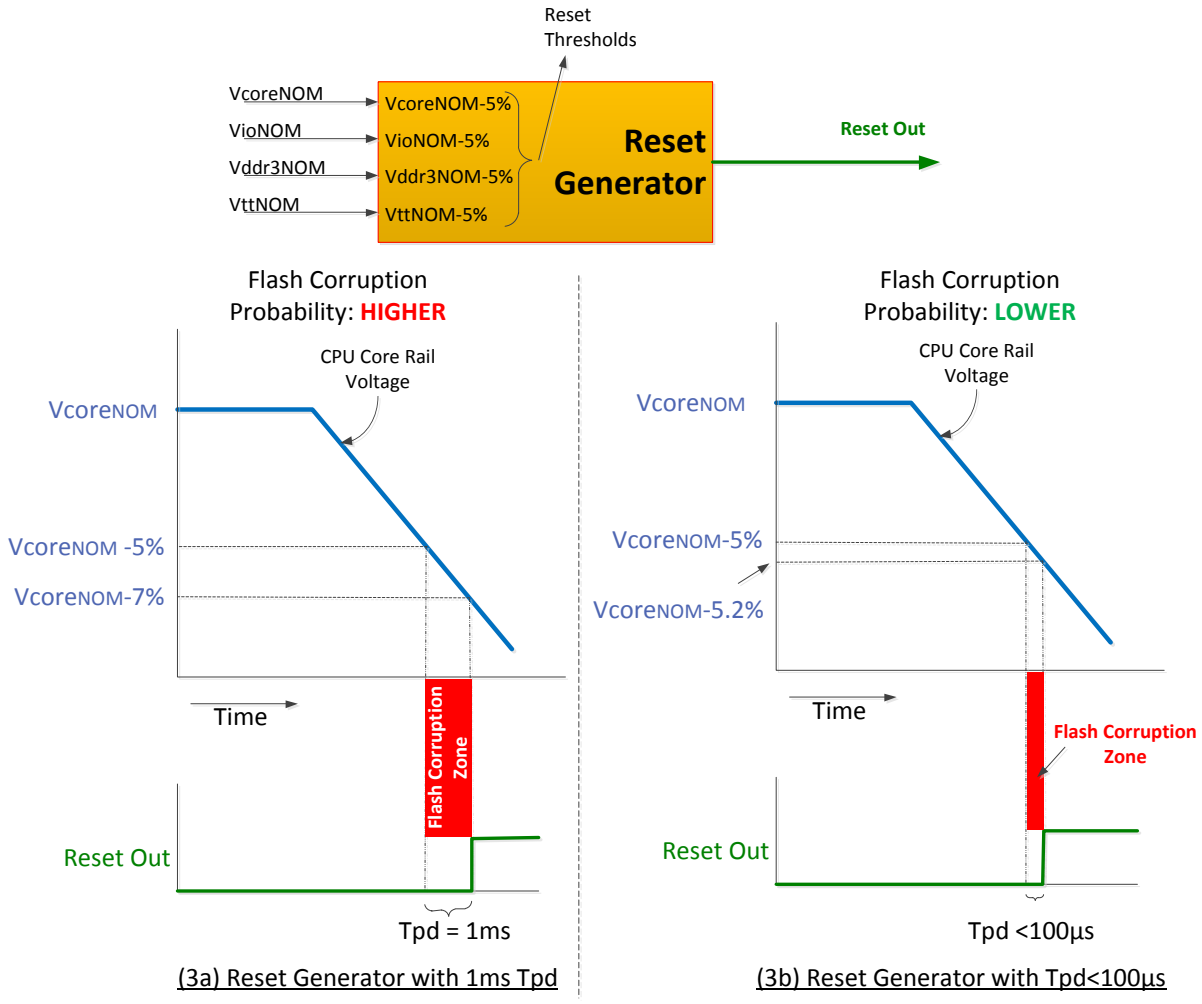

Figure 3 – Flash Corruption Probability vs. Reset Generator Fault Detection Speed

Errors to avoid: Some designs use a microcontroller with an ADC as a reset generator. The reset is activated by its voltage monitoring routine, which is activated by a 10ms to 50ms real time clock Interrupt. Consequently, the reset can be activated with a delay of 10ms to 50ms. Because of this long delay, this method will not be able to prevent Flash corruption. Note that the voltage monitoring accuracy of an ADC is determined by its errors and the error of the on-chip ADC voltage reference. The number of ADC bits is not a measure of its accuracy.

Summary

The conventional thinking that Flash corruption is due only to a software bug results in engineers wasting time looking for one that does not exist. Flash corruption can occur after the power input to the board is disconnected. The only way to minimize the chances of Flash corruption is by holding the processor in reset when there is a supply voltage fault. The probability of Flash corruption can be significantly reduced by using a more accurate (<1% voltage fault detection error) and faster (Fault detection speed <100us) reset generator.