Errata

DA1470x Errata

CE0001

Abstract

The document contains the known errata for the DA1470x along with recommended workarounds.

Product	DA1470x
Silicon Revision	0xAB (Register CHIP_REVISION_REG and CHIP_TEST1_REG)
Datasheet Reference	DA1470x_Datasheet.pdf v3.1
Package(s)	VFBGA142 (142 balls) 6.2 mm x 6 mm
Issue Date	29-Sep-2022

Contents

1 Terms and Definitions

ADC	Analog to Digital Converter
BLE	Bluetooth® Low Energy
BOD	Brown-Out Detection Controller
BS	Bit-Stream Controller
CC-CV	Constant Current – Constant Voltage
CM0+	ARM Cortex® M0+
CMAC	Configurable MAC (BLE)
LCDC	LCD Controller
LDO	Low Drop-Out Voltage regulator
OSR	Over Sampling Rate
OTP	One Time Programmable Memory
RFCU	RF Control Unit
RFMON	RF Monitoring Controller
RSSI	Received Signal Strength Indication
SDK	Software Development Kit
SRC	Sample Rate Converter

2 Identifying the Silicon Revision

A readback of registers CHIP_REVISION_REG and CHIP_TEST1_REG will give the device revision information.

3 Workarounds

For the cases where complex software workarounds are recommended in form of concept description, refer to the special section of the SDK release notes that explains which items are addressed and how.

4 Errata Summary

Table 1: Errata Summary

Item No	Item Title
335	GPIOs: P0_28 (Wake-up) Pad Uses Input Enable Control from P0_27
334	Charger: Back Drive Switch Always Enabled
337	SIMO DCDC: Freewheel Mode Increases IQ
338	Booter: QSPI_CLK Throttling Fails
339	WAKEUP: ENABLE_FAST_SWITCH is Triggered on Both Edges
328	DCACHE: Wrong Address Used in Dirty Line Evict when Cacheable and Non-Cacheable Areas are Used
340	DCACHE: MRM Miss Counter (during AHB Write to Dirty Line)

Item No	Item Title
278	Display Controller: Stride Issue
341	Timers: Single Event Capture Not Reset
297	Display Controller: Wrong Partial Refresh
342	V30 Level Increases when Going to Sleep
332	Disabling Bandgap Clamp Can Cause a BOD Event
344	SNC Hready not Properly Connected to the AHB2APB Bridge
-	eMMC Memory Compatibility
302	Charger CC-to-CV Comparator Has Low Hysteresis
362	Supply Dips when Switching over to Retention LDOs
363	LDO Hold Causes Jumps in ADC Reference
364	Boost Converter Output Shows a Sawtooth Disturbance at Certain Loads/Supply Combinations
365	Lower SINAD During SRC Bypass Mode
366	LDO_VSYS Voltage Loop Adds Noise on VSYS Rail
367	LDO_VSYS Headroom Loop Shows Small Oscillation of 400 kHz
368	Missing Retention for Dcache WFLUSH Done Register
369	Booter: UART not reaching 1 Mb
370	Booter: QSPI Flash Memories Larger than 128 Mb Cannot Be Addressed
372	RFMON Fails to Capture GPADC Data when SYS_CLK is Asyncronous to CMAC_CLK
373	Timer4 IRQ Is Not Connected to CMAC SYSPER IRQ
374	Increased XTAL32M Startup Time at Specific OFF Time
375	Wakeup from CMAC Debugger Is Not Functional
376	Enable Signals for the BOD Channels V18P and V18F Are Swapped
377	USB MAC Pclk Frequency Can Go Up To 120 MHz
378	RCLP Glitch
320	V18F High Ohmic Switch
-	Display Controller: Data Read Failure in Read Mode

5 Errata Details

5.1 GPIOs: P0_28 (Wake-up) Pad Uses Input Enable Control from P0_27

5.1.1 Effect

In normal mode, the input state of P0_28 is by mistake connected to the input state of P0_27. If P0_27 is configured to be anything else but a digital input (for example, an ADC input), then P0_28 becomes unusable. Both can be used if they are on the same state (for example, digital inputs).

When in Hibernation mode, however, the wake-up function of P0_28 is not affected by the P0_27 state.

5.1.2 Conditions

This issue arises when the state of the P0_27 is different than the state of the P0_28. States are digital input, digital output, analog, and special (for example, ADC input, and so forth).

5.1.3 Technical Description

The Input_Enable signal of pad P0_28 is connected to the wrong place. It connects to the Input_Enable of pad P0_27 instead. This is hard-wired mistake in the schematic.

In non-hibernation mode, the P0_27_MODE_REG[PID] field controls the input path of P0_28. This includes sleep modes and latched states with PadLatchEnable.

5.1.4 Workaround

1. Set the PID for P0_27 and P0_28 always the same.
2. Use only during hibernation, set the P0_27 PID to digital, just before entering hibernation state. Set bit [3] of WAKEUP_HIBERN_REG[WAKEUP_EN].
3. Do not use P0_27 as ADC input but keep it in digital mode. Then both P0_27 and P0_28 input circuits work fine.

5.2 Charger: Back Drive Switch Always Enabled

5.2.1 Effect

There is a constant load of 100 kΩ on the battery while charging.

5.2.2 Conditions

Always happening when charging is enabled.

5.2.3 Technical Description

When the VBUS_AVAILABLE comparator output inside the charger goes high, the backdrive protection PMOS switch in the charger is immediately turned on (= low Ron).

The control circuit to switch this PMOS on applies a 100 kΩ load on VBAT.

5.2.4 Workaround

There is no workaround available for this because it is only HW control.

5.3 SIMO DCDC: Freewheel Mode Increases I_Q

5.3.1 Effect

Quiescent current (I_Q) is increased by ~20 µA.

5.3.2 Conditions

Only when the freewheel mode is selected (BUCK_CTRL_REG[EN_FREEWHEEL] = 1).

5.3.3 Technical Description

When freewheel mode is enabled, the converter uses (far) more quiescent current than allowed. This is due to FW mode enabling the N-switch (low-side switch on the LX pin) when the converter is idle, which in turn enables a resistive divider that controls a PMOS pull-up in the P-switch (high-side switch on the LX pin) driver.

5.3.4 Workaround

Do not use this mode when system relies on the DCDC in sleep mode. By default, this mode is disabled.

5.4 Booter: QSPI_CLK Throttling Fails in Secure Boot

5.4.1 Effect

System will be HW reset after trying to secure boot by a FLASH device not able to operate at 96 MHz.

5.4.2 Conditions

Occurs only when secure booting is enabled, in which case the RCHS clock is switched to 96 MHz frequency and the external FLASH component is not accessible at 96 MHz.

5.4.3 Technical Description

When secure boot is enabled, RCHS switches to 96 MHz and QSPI_CLK is set to 96 MHz as well. The booter will try to access the external FLASH device at this speed. If this is not successful, the booter is supposed to reset the Flash, re-apply the loader to put it in auto mode and lower the QSPI_CLK frequency. The issue is that the QSPI_CLK frequency will not be lowered after all, which brings the system into the same situation as before. The booter will then decide the loader is not available and it will trigger a HW reset.

5.4.4 Workaround

Add an entry into the Configuration Script (CS) in the OTP to instruct the booter to run on XTAL32M instead of RCHS.

5.5 WAKEUP: ENABLE_FAST_SWITCH is Triggered on Both Edges

5.5.1 Effect

System will be woke up accidentally by de-asserting (de-activating) the external wake-up signal.

5.5.2 Conditions

If the system is configured for “Fast Wakeup” and was woken up by a signal asserted and went back to sleep before the de-assertion of that signal.

5.5.3 Technical Description

When enabling the ENABLE_FAST_SWITCH mode for early clock switching, the chip will wake up early, as explained in the datasheet.

However, when the pin is released while the chip is already back in sleep mode, another trigger happens.

This is because an edge detection is done using an XOR, detecting both edges. It should only trigger on a rising edge, not on the falling edge.

5.5.4 Workaround

Keep system alive while the wake-up pin is de-asserted.

5.6 DCACHE: Wrong Address Used in Dirty Line Evict when Cacheable and Non-Cacheable Areas are Used

5.6.1 Effect

Data residing in the data cache will be written on the wrong address in the PSRAM area.

5.6.2 Conditions

A specific sequence of operations listed in the following sections while having the PSRAM address space setup for both cacheable and non-cacheable accesses.

5.6.3 Technical Description

When a “dirty line” is evicted, the data is written to the wrong address when the following events occur in the listed sequence:

1. One or more writes to non-cacheable addresses fills the WBUFFER of the cache controller.

NOTE
Both accessing cacheable as well as non-cacheable are directed through the data cache controller.

2. A read hit happens.
3. A read/write miss is happening to a cacheable region that forces the FSM to evict a dirty line.

If the sequence described above is happening, the address for the write (see #1 above) is constructed wrongly, so that the LSB's (corresponding to the TAG memory address lines are used from the previous access).

5.6.4 Workaround

Either use the complete PSRAM space for a cacheable or non-cacheable access. The user should not mix cacheable and non-cacheable operations on the same address space covered by the data cache controller to avoid triggering the specific sequence of operations described above.

5.7 DCACHE: MRM Miss Counter (during AHB Write to Dirty Line)

5.7.1 Effect

The misses amount counted by the miss rate monitor engine of the data cache controller is not accurate.

5.7.2 Conditions

Always.

5.7.3 Technical Description

The MRM for the DCACHE does count HITS, MISS, and EVICT events. In the case of a AHB write to a dirty location in the DCACHE an EVICT increment is triggered but additionally a MISS increment should happen, because a new cache line needs to be read from the main memory. The MISS increment is not happening.

5.7.4 Workaround

No actual workaround available. If the number of accesses N to the cache is known, the number of missing MISS events can be calculated using the number of HITS, so that MISS = N - HITS.

5.8 Display Controller: Stride Issue

5.8.1 Effect

Artifacts on the display.

5.8.2 Conditions

Changing stride lengths while in continuous mode displaying the LCD controller and before the frame has successfully been transmitted.

5.8.3 Technical Description

Most of the layer registers (LCDC_LAYER0_STARTXY_REG, LCDC_LAYER0_SIZEXY_REG, LCDC_LAYER0_BASEADDR_REG, LCDC_LAYER0_RESXY_REG, and LCDC_LAYER0_OFFSETX_REG) are protected of being changed at inappropriate time. On the contrary, LCDC_LAYER0_MODE_REG and LCDC_LAYER0_STRIDE_REG are not. The protection mechanism applies the input values only when a frame has been completed or when there is a write to LCDC_MODE_REG.

In case of the single frame update, LCDC_MODE_REG is written in every frame and as a result, all layer parameters are applied immediately.

On the other hand, in case of continuous update, LCDC_MODE_REG is only accessed once to set up the mode and as a result the new layer settings are applied with the frame end signal. This has the effect that the stride is taking effect in the next output frame, but rest of the parameters on the second output frame creating a discrepancy and eventually artifacts at the display.

5.8.4 Workaround

1. Store the stride size instead of applying it and then apply it at the frame end interrupt.
2. Forcefully apply the protected registers by performing a dummy write to the LCDC_MODE_REG.
3. Set protected registers and wait for the frame end to apply the new stride value.

Workaround #1 is implemented in the released Software Development Kit (SDK).

5.9 Timers: Single Event Capture Not Reset

5.9.1 Effect

The “single event capture” feature available for Timer and Timer4 only works once, it cannot be set to re-capture unless the respective Timer is reset completely.

5.9.2 Conditions

Always, after a first event was captured.

5.9.3 Technical Description

When single event capture is enabled, only the first event is captured. To reuse this feature, the complete Timer module needs to be reset.

5.9.4 Workaround

A possible workaround would be to trigger a false reset event using software, so that the single event capture FSM resets and can be used again.

5.10 Display Controller: Wrong Partial Refresh

5.10.1 Effect

Wrong display data.

5.10.2 Conditions

When partially refreshing of the display is programmed and the number of the columns to be refreshed is 1.

5.10.3 Technical Description

In a partial refresh of the LCD can be requested to be refreshed a small number of LCD columns. If the number of LCD columns is 1, the output of the LCDC is wrong. The output is constant and has the value of the first pixel selected.

The bug affects at least these enabled color modes: RGB332, RGB565.

5.10.4 Workaround

Restrict the number of columns of a partial display refresh to anything but 1. In a case of request for the refreshing of 1 column the number of columns should be increased to at least 2 in the software driver of the LCDC.

This fix is implemented in the released low-level driver of the Software Development Kit (SDK).

5.11 V30 Level Increases when Going to Sleep

5.11.1 Effect

V30 increased voltage.

5.11.2 Conditions

Always when switching the system into sleep.

5.11.3 Technical Description

When going to sleep, during the first Bandgap refresh cycle, V30 level increases abnormally. This is due to a very limited time interval (1 clk cycle) between the LDO_V30_RET being enabled and hold mode being asserted. In this case, the LDO kicks the reference that is sampled at the wrong level.

5.11.4 Workaround

Proposed workaround is to keep the LDO_V30_RET always on. During sleep it will be the only LDO on V30 rail. During active it will supply the rail when there is no load. If there is load LDO_V30 will take over.

5.12 Disabling Bandgap Clamp Can Cause a BOD Event

5.12.1 Effect

Reset caused by the BOD.

5.12.2 Conditions

Always when disabling the Bandgap clamp.

5.12.3 Technical Description

Turning off the bandgap clamp (BANDGAP_REG[BANDGAP_ENABLE_CLAMP] = 0) can cause a spike on the voltage references. The BOD can detect a higher reference and thus consider a valid voltage level too low and trigger a POR.

This can also occur after a reset if the bandgap clamp was active.

5.12.4 Workaround

Proposed workaround is to disable the BOD before disabling the bandgap clamp and then enable it again.

5.13 SNC Hready not Properly Connected to the AHB2APB Bridge

5.13.1 Effect

Leads into failing writes when wait cycles are introduced on the bus.

5.13.2 Conditions

The HREADY response output that the AHB-to-APB bridge produces is used as input, thereby it is not considering HREADY that the master uses.

5.13.3 Technical Description

The HREADY signal of the SNC is not connected to the corresponding input of the AHB-to-APB bridge. This can lead into failing register writes when wait cycles are introduced on the bus, because of back-to-back writes that the SNC may perform to memory and then to registers (for example, of the APB32-slow address space).

5.13.4 Workaround

To prevent the failing APB32-slow register writes (while keeping SLOW_PCLK at 32 MHz), a workaround is the interleaving of NOP commands in between the SYSRAM and APB register writes, in the SNC C code.

An alternative is to lower SLOW_PCLK, dividing it by 2 (CLK_AMBA_REG[SLOW_PCLK_DIV]=0x1). This results into a penalty of extending the execution time of the SNC CPU, because the latter uses SLOW_PCLK as clock.

5.14 eMMC Memory Compatibility

5.14.1 Effect

Incompatibility with eMMC devices.

5.14.2 Conditions

The selected eMMC device is not tollerant to clock speed that violates the eMMC specification.

5.14.3 Technical Description

Some eMMC devices require that the identification clock rate f_{od} should be as specified in the standard, that is max 400 kHz. Therefore, these devices are not compatible because the minimum eMMC bus speed is 6 MHz (RCHS96/16) due to the clock divider range limitation (Divide only by 1, 2, 4, 8, and 16).

5.14.4 Workaround

Use eMMC devices that can operate even if f_{od} is higher than 400 kHz.

5.15 Charger CC-to-CV Comparator Has Low Hysteresis

5.15.1 Effect

Charge Cc_mode and cv_mode signals toggle.

5.15.2 Conditions

Always.

5.15.3 Technical Description

There is not enough hysteresis built in the comparators that define the CC or CV mode in the analog loop. Hence, when the battery voltage reaches a threshold from which the transition from CC to CV mode begins, the respective signals (cc_mode, cv_mode) driven by the Chargers analog block towards the digital FSM are toggling.

Because these signals are complementary to each other, this makes the digital FSM transitioning between the CC and CV states, until the battery voltage is sufficiently increased, stabilizing the analog loop in CV mode and also the digital FSM in CV state.

5.15.4 Workaround

The interrupts signaling the transition of the digital FSM from CC to CV state and vice versa can be masked. This is possible by disabling the respective interrupt enable bit-fields, as follows:

- CHARGER_STATE_IRQ_MASK_REG[CC_TO_CV_IRQ_EN] = '0'
- CHARGER_STATE_IRQ_MASK_REG[CV_TO_CC_IRQ_EN] = '0'

This only masks the interrupts due to the specific FSM transitions. Thus, any other Charger interrupt that may occur due to normal charging or because of an error condition during charging is not affected and will be captured, provided that the respective interrupt mask register bit is enabled.

5.16 Supply Dips when Switching over to Retention LDOs

5.16.1 Effect

Supply dips when switching over to retention LDOs.

5.16.2 Conditions

When switching over from normal LDO to Retention LDO.

5.16.3 Technical Description

When switching over from normal LDO to Retention LDO, there are dips in the supply voltage when the nominal output voltage of the normal LDO is larger than the output of the retention LDO. This is caused by the slow startup and recovery behavior of the retention LDOs.

Dips are in the order of 50 to 150 mV and are 50 to 100 µs wide.

When the normal LDO has a higher output voltage and both LDOs (normal and retention) are enabled, the normal LDO will supply the load. The retention LDO will settle to a non-working condition, meaning that the bias-point of the cell will be far away from the normal operating point. When the normal LDO is switched off, the retention LDO has to move quickly to the normal operating point, but this takes some time. If there is some load, the output will drop significantly before the retention LDO is operating normal. When the load is very low, the drop in the supply will be less. When the normal LDO has an output voltage which is below the output voltage of the retention-LDO, there will be no drop. The reason for this is that the retention LDO will not shift to the non-working operating point in that case.

5.16.4 Workaround

A workaround for this behaviour can be used when the normal LDO is trimmed to a value just below the output of the retention LDO.

5.17 LDO Hold Causes Jumps in ADC Reference

5.17.1 Effect

A jump in LDO output (ADC Reference) occurs.

5.17.2 Conditions

When LDO_HOLD is used.

5.17.3 Technical Description

When LDO_HOLD is used, a jump in the LDO output (ADC Reference) occurs. This causes large INL and DNL errors whenever resampling of the LDO reference (using LDO_HOLD) is used. Additionally, the drop in hold is bigger than expected (~ 1 mV/5 s).

Due to the jumps, the gain and offset correction (OTP values) cannot be used along with using LDO_HOLD.

5.17.4 Workaround

Do not use LDO_HOLD.

5.18 Boost Converter Output Shows a Sawtooth Disturbance at Certain Loads/Supply Combinations

5.18.1 Effect

Boost converter output shows a sawtooth disturbance.

5.18.2 Conditions

At certain loads and input voltages.

5.18.3 Technical Description

A sawtooth pattern is present on the output voltage at certain loads and input voltages. This is caused by impedance mismatch between the inputs of the clocked comparator that monitors VLED due to a filter that is present on analog core and coupling of the VOUT_NOK output of the clocked comparator to the reference line.

5.18.4 Workaround

None.

5.19 Lower SINAD During SRC Bypass Mode

5.19.1 Effect

Lower Signal-to-noise and distortion ratio (SINAD).

5.19.2 Conditions

SRC bypass mode.

5.19.3 Technical Description

When the SRC bypass mode is selected, the signal quality is impacted at high input levels (that is full scale). The SINAD measures approximately 86 dB. Modifying the input signal level to -1 dB brings the signal quality back to meeting the 100 dB SINAD specification.

5.19.4 Workaround

Do not use output bypass mode for signals above -1 dB FS when the signal degradation to 86 dB SINAD is unacceptable.

5.20 LDO_VSYS Voltage Loop Adds Noise on VSYS Rail

5.20.1 Effect

Noise on VSYS rail.

5.20.2 Conditions

LDO_VSYS in regulating mode.

5.20.3 Technical Description

An unexpected high noise level on node VSYS exists when the voltage loop of the LDO_VSYS is in regulating mode. This noise has LF characteristics.

5.20.4 Workaround

None.

5.21 LDO_VSYS Headroom Loop Shows Small Oscillation of 400 kHz

5.21.1 Effect

A 400 kHz signal is detected in the VSYS rail.

5.21.2 Conditions

The headroom loop is in command in LDO_VSYS.

5.21.3 Technical Description

When the headroom loop is active in LDO_VSYS, a small 400 kHz signal is detected in the VSYS rail which is a loop oscillation. This oscillation is load-current dependent.

5.21.4 Workaround

The oscillation condition improves by adding a series resistor in the VBUS Cap-lead. A series resistor of 0.3 to 0.5 Ω with a 10 µF cap is recommended.

5.22 Missing Retention for Dcache WFLUSH Done Register

5.22.1 Effect

An unintended data cache flush operation might be started.

5.22.2 Conditions

Certain conditions after a (retained) sleep cycle.

5.22.3 Technical Description

The write flush operation is started by software (writing a 1 into the DCACHE_CTRL_REG[DCACHE_WFLUSH] bit). This action will force a toggle of an internal (cfg_wflush_reg) register. On completion of the write flush operation, the cfg_wflush_done signal is updated to the same value of wflush_reg. So, a XOR of both signals does indicate if a write flush needs to be started and/or is active.

The problem, however, is that cfg_wflush_reg is retained during a sleep cycle while the cfg_wflush_done is not. This means that after a (retained) sleep cycle, when PD_SYS is off, an unintended write flush operation might be started.

Additionally, the process used to generate/update the WFLUSHED status bit is connected to the retention reset, but the corresponding registers are not of retainable type. So the dcache_wflushed_reg and the cfg_wflush_prev registers are undefined after a sleep cycle, but will resolve to a defined value after 2 HCLK cycles.

5.22.4 Workaround

A software workaround can be used. To perform a write flush, the following sequence is used:

```
do{
DCACHE_CTRL_REG[DCACHE_WFLUSH] = 1;
}
```

```
while (DCACHE_CTRL_REG[DCACHE_WFLUSHSED] == 0);
```

To avoid unintended write flush operations after a power cycles (with data retention enabled) execute the above sequence two times.

NOTE
The second write flush will ensure that the write flush trigger bit is toggled twice (that is returning to 0) thereby avoiding unexpected write flush after a power cycle.

5.23 Booter: UART not reaching 1 Mb

5.23.1 Effect

The booter misses intermediate bytes.

5.23.2 Conditions

UART boot at 1 MBaud rate.

5.23.3 Technical Description

The booter runs out of cycle budget at 1 MBaud UART speeds due to way that reads out the data provided over the UART interface. This causes the booter to miss intermediate bytes.

5.23.4 Workaround

Use speeds up to 500 kb.

5.24 Booter: QSPI Flash Memories Larger than 128 Mb Cannot Be Addressed

5.24.1 Effect

QSPI Flash Memories larger than 128 Mb cannot be addressed by the Booter.

5.24.2 Conditions

Always.

5.24.3 Technical Description

Due to legacy booter code (where only particular flash types were considered) the booter only uses the 0x3 SPI read command. This command has a 24-bit address limitation (and thus 128 Mb). Larger Flash devices also support the 0x13 read command which takes 32-bit addresses. Booter does not support 0x13 read command.

5.24.4 Workaround

Use QSPI Flash devices up to 128 Mb.

5.25 RFMON Fails to Capture GPADC Data when SYS_CLK is Asyncronous to CMAC_CLK

5.25.1 Effect

RFMON fails to capture GPADC data.

5.25.2 Conditions

When SYS_CLK is async to CMAC_CLK.

5.25.3 Technical Description

No synchronization exists between the GPADC_RESULT (sys_clk) and the RFMON (cmac_clk). This means that when the GPADC is running on an asynchronous clock (mainly RCHS, but also SYS_CLK = PLL and CMAC_CLK = XTAL32M) intermediate data will be sampled and can be corrupted.

5.25.4 Workaround

Run GPADC and CMAC on synchronous clocks.

5.26 Timer4 IRQ Is Not Connected to CMAC SYSPER IRQ

5.26.1 Effect

The IRQ line from Timer 4 to CMAC is missing.

5.26.2 Conditions

Always.

5.26.3 Technical Description

The IRQ line from Timer 4 to CMAC is missing. A SW-based COEX interface that needs Timer4 can not be implemented directly.

5.26.4 Workaround

The SW workaround is to let Timer4 to trigger M33 and then M33 to notify CMAC via the SYS2CMAC_IRQ.

5.27 Increased XTAL32M Startup Time at Specific OFF Time

5.27.1 Effect

Increased XTAL32M startup time.

5.27.2 Conditions

Residual energy in the resonator.

5.27.3 Technical Description

The XTAL32M startup time depends on the time that the XTAL32M was previously OFF. This is caused by residual energy in the resonator, that typically speeds up the subsequent startup. In some cases, however, the startup time may be increased instead.

5.27.4 Workaround

The SW workaround is checking the xtal_rdy bit before using the clock. If the xtal_rdy bit is not set when the IRQ fires, the startup time is longer than expected. Furthermore, the xtal amplitude regulator sample-and-hold needs to be forced to TRACK mode.

5.28 Wakeup from CMAC Debugger Is Not Functional

5.28.1 Effect

CMAC cannot wake up from debugger.

5.28.2 Conditions

Always.

5.28.3 Technical Description

The logic that will wake up the system when there is activity on CMAC debugger is not functional. The logic remains on reset while CMAC is in power down because the CMAC Sleep Timer port "cmac_rst_n" signal is connected to the reset of the CMAC Core power domain and not the reset of the CMAC Sleep Timer power domain.

5.28.4 Workaround

None.

5.29 Enable Signals for the BOD Channels V18P and V18F Are Swapped

5.29.1 Effect

The BOD for the channels V18P and V18F cannot be used independently.

5.29.2 Conditions

Always.

5.29.3 Technical Description

Due to misconnected internal signals, the BOD_V18P_EN and BOD_V18P_RST_EN control the V18F comparator and the BOD_V18F_EN and BOD_V18F_RST_EN control the V18P comparator.

To enable BOD on V18P, BOD_V18F_EN and BOD_V18F_RST_EN should be enabled.

To enable BOD on V18F, BOD_V18P_EN and BOD_V18P_RST_EN should be enabled.

However, in sleep mode, these signals are gated with:

- SW_V18F_SLEEP_ON – that gates BOD_V18P
- DCDC_V18P_SLEEP_EN – that gates BOD_V18F

5.29.4 Workaround

Both signals should be enabled for the BOD to be working in sleep mode. In any other case, a reset will not happen, even when the rail is too low.

5.30 USB MAC Pclk Frequency Can Go Up To 120 MHz

5.30.1 Effect

USB MAC miss bytes.

5.30.2 Conditions

Pclk higher than 120 MHz.

5.30.3 Technical Description

USB block is connected to two clock domains, pclk and 48 MHz. When the USB MAC sends a byte, the event is sychronized to the 48 MHz clock domain. An ack is generated at 48 MHz and sychronized to pclk clock domain. The next byte can be sent when the pclk domain receives the ack. If pclk is too fast the next byte event is missed.

5.30.4 Workaround

Pclk shoudl be < 120 MHz.

5.31 RCLP Glitch

5.31.1 Effect

Improper clocking of some digital blocks.

5.31.2 Conditions

Undefined.

5.31.3 Technical Description

In a small number of devices, RCLP shows glitches. The problem appears more severe for lower frequencies and higher VDDs. Blocks like the Timers can be affected.

5.31.4 Workaround

RCX should be used instead of RCLP when the Timers are needed and XTAL32K is not available. For the cases where RCLP must be used (for example, cold boot, and so forth), glitches do not affect the device and no application problems appear.

5.32 V18F High Ohmic Switch

5.32.1 Effect

This has impact on the V18F level which supplies the QSPI FLASH chip externally, when the FLASH chip draws ~20 mA during program or erase. The V18F voltage can become too low.

5.32.2 Conditions

Always.

5.32.3 Technical Description

High resistance caused due to a layout issue on the switch that internally supplies V18F through V18P.

5.32.4 Workaround

V18F and V18P rails must be shorted externally and V18F switch must be forced always open.

5.33 Display Controller: Data Read Failure in Read Mode

5.33.1 Effect

Data read failure while display controller is in read mode.

5.33.2 Conditions

Issue occurs when LCDC_STATUS_REG[DBIB_CMD_PENDING] and LCDC_STATUS_REG[DBIB_OUT_TRANS_PENDING] bits are used to synchronize the read transaction (CMD bytes sending or DATA bytes reading).

5.33.3 Technical Description

In SPI3/4, Dual/Quad SPI, and DBIB mode, a user will initiate a read transaction by sending the first Read CMD byte and based on the status of the LCDC_STATUS_REG[DBIB_CMD_PENDING]. LCDC_STATUS_REG[DBIB_OUT_TRANS_PENDING], and LCDC_STATUS_REG[SPI_RD_WR_OP] bits will continue with the following Read CMD bytes or start reading DATA bytes. The normal operation of those bits is:

1. A CMD byte is sent.
2. LCDC_STATUS_REG[DBIB_CMD_PENDING] is set.
3. LCDC_STATUS_REG[DBIB_OUT_TRANS_PENDING] is set.
4. LCDC_STATUS_REG[DBIB_CMD_PENDING] is cleared.
5. LCDC_STATUS_REG[SPI_RD_WR_OP] is set.
6. LCDC_STATUS_REG[DBIB_OUT_TRANS_PENDING] is cleared.
7. LCDC_STATUS_REG[SPI_RD_WR_OP] is cleared.
8. Next CMD byte can be sent or DATA byte can be read.

The issue occurs between step 3 and step 4. The DBIB_CMD_PENDING and DBIB_OUT_TRANS_PENDING bits are not set and cleared with the proper sequence and as such, the software cannot depend on those status bits.

Additionally, the LCDC_STATUS_REG[SPI_RD_WR_OP] bit is not immediately set when setting the read cycles LCDC_DBIB_RDAT_REG[DBIB_READ_C] to perform a read, hence there is not a reliable way to wait until a read operation is complete.

5.33.4 Workaround

- For the SPI3/4 and Dual/Quad SPI modes, a user must use the LCDC_STATUS_REG[SPI_RD_WR_OP] bit only. The user must wait from the point the CMD is added to the FIFO or read cycles are set until the bit is set with a fixed timeout. Then, a user must wait until the bit is cleared. The timeout is calculated using the formula: 2 x SPI clock ticks' period.
- For DBIB mode, the LCDC_STATUS_REG[SPI_RD_WR_OP] bit cannot be used, hence a user must only use a fixed time delay from the point a CMD is added to the FIFO or read cycles are set. The delay is calculated using the formula (bytes + 3) x DBIB clock ticks' period for the CMD bytes and (bytes + 1) x DBIB clock ticks' period for the read operation.

Revision History

Revision	Date	Description
1.1	29-Sep-2022	Section 5.13 to 5.33 added. Applies to final datasheet.
1.0	15-Jun-2022	Initial release. Applies to final datasheet.

Status Definitions

Status	Definition
DRAFT	The content of this document is under review and subject to formal approval, which may result in modifications or additions.
APPROVED or unmarked	The content of this document has been approved for publication.