DA1470x

DA1470x Errata

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

1. Information

Table 1. Information table

Datasheet	DA1470x Multi-Core Wireless SoC Family with PMU and GPU (rev. 3.2)
Package(s)	VFBGA142 (142 balls) 6.2 mm x 6 mm

2. Identifying the Silicon Revision

Silicon Revision is 0xAA, 0xAB (Register CHIP_REVISION_REG and CHIP_TEST1_REG).

3. Errata Summary

Table 2. Errata summary

Issue #	Issue title
1	GPIOs: P0_28 (Wake-up) Pad uses input enable control from P0_27
2	Charger: back drive switch always enabled
3	SIMO DCDC: Freewheel mode increases IQ
4	Booter: QSPI_CLK throttling fails
5	WAKEUP: ENABLE_FAST_SWITCH is triggered on both edges
6	DCACHE: wrong address used in dirty line evict when cacheable and non-cacheable areas are used
7	DCACHE: MRM miss counter (during AHB write to dirty line)
8	Display controller: stride issue
9	Timers: single event capture not reset
10	Display controller: wrong partial refresh
11	V30 level increases when going to sleep
12	Disabling bandgap clamp can cause a BOD event
13	SNC hready not properly connected to the AHB2APB bridge
14	eMMC memory compatibility
15	Charger CC-to-CV comparator has low hysteresis
16	Supply dips when switching over to retention LDOs
17	LDO hold causes jumps in ADC reference
18	Boost converter output shows a sawtooth disturbance at certain loads/supply combinations
19	Lower SINAD during SRC Bypass mode
20	LDO_VSYS voltage loop adds noise on VSYS rail
21	LDO_VSYS headroom loop shows small oscillation of 400 kHz
22	Missing retention for Dcache WFLUSH done register
23	Booter: UART not reaching 1 MB
24	Booter: QSPI flash memories larger than 128 MB cannot be addressed
25	RFMON fails to capture GPADC data when SYS_CLK is asynchronous to CMAC_CLK
26	Timer4 IRQ is not connected to CMAC SYSPER IRQ
27	Increased XTAL32M startup time at specific OFF time
28	Wake-up from CMAC debugger is not functional
29	Enable signals for the BOD channels V18P and V18F are swapped

Issue #	Issue title
30	USB MAC Pclk frequency can go up to 120 MHz
31	RCLP glitch
32	V18F high ohmic switch
33	Display controller: data read failure in Read mode

4. Errata Details

Table 3. GPIOs: P0_28 (wake-up) pad uses input enable control from P0_27

Issue #	Effect
1.	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 in 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.
	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 on).
	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 a 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.
	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.

Table 4. Charger: back drive switch always enabled

Issue #	Effect
2.	There is a constant load of 100 kΩ on the battery while charging.
	Conditions
	Always happening when charging is enabled.
	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.
	Workaround
	There is no workaround available for this because it is only hardware control.

Table 5. SIMO DCDC: Freewheel mode increases I_Q

Issue #	Effect
3.	Quiescent current (I_Q) is increased by ~20 µA.
	Conditions
	Only when the Freewheel mode is selected (BUCK_CTRL_REG[EN_FREEWHEEL] = 1).
	Technical description
	When Freewheel mode is enabled, the converter uses (far) more quiescent current than allowed. This is due to Freewheel 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.
	Workaround
	Do not use this mode when system relies on the DCDC in Sleep mode. By default, this mode is disabled.

Table 6. Booter: QSPI_CLK throttling fails in secure boot

Issue #	Effect
4.	System is hardware reset after trying to secure boot by a FLASH device not able to operate at 96 MHz.
	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.
	Technical description
	When secure boot is enabled, RCHS switches to 96 MHz and QSPI_CLK is set to 96 MHz as well. The booter tries 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 is not lowered after all, which brings the system into the same situation as before. The booter decides the loader is not available and it triggers a hardware reset.
	Workaround
	Add an entry into the Configuration Script (CS) in the OTP to instruct the booter to run on XTAL32M instead of RCHS.

Table 7. WAKEUP: ENABLE_FAST_SWITCH is triggered on both edges

Issue #	Effect
5.	System is woken up accidentally by de-asserting (de-activating) the external wake-up signal.
	Conditions
	If the system is configured for "Fast Wake-up" and was woken up by a signal asserted and went back to sleep before the de-assertion of that signal.
	Technical description
	When enabling the ENABLE_FAST_SWITCH mode for early clock switching, the chip wakes up early, as explained in the datasheet. 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.
	Workaround
	Keep system alive while the wake-up pin is de-asserted.

Table 8. DCACHE: wrong address used in dirty line evict when cacheable and non-cacheable areas are used

Issue #	Effect
6.	Data residing in the data cache will be written on the wrong address in the PSRAM area.
	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.
	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. (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 described sequence 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).
	Workaround
	Either use the complete PSRAM space for a cacheable or non-cacheable access. You 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.

Table 9. DCACHE: MRM miss counter (during AHB write to dirty line)

Issue #	Effect
7.	The misses amount counted by the miss rate monitor engine of the data cache controller is not accurate.
	Conditions
	Always.
	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.
	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.

Table 10. Display controller: stride Issue

Issue #	Effect
8.	Artifacts on the display.
	Conditions
	Changing stride lengths while in Continuous mode displaying the LCD controller and before the frame has successfully been transmitted.
	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 from 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.
	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).

Table 11. Timers: single event capture not reset

Issue #	Effect
9.	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.
	Conditions
	Always, after a first event was captured.
	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.
	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.

Table 12. Display controller: wrong partial refresh

Issue #	Effect
10.	Wrong display data.
	Conditions
	When partially refreshing of the display is programmed and the number of the columns to be refreshed is 1.
	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.
	Workaround
	Restrict the number of columns of a partial display refresh to anything but 1. In the 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).

Table 13. V30 level increases when going to sleep

Issue #	Effect
11.	V30 increased voltage.
	Conditions
	Always when switching the system into sleep.
	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.
	Workaround
	The proposed workaround is to keep the LDO_V30_RET always on. During sleep it is the only LDO on V30 rail. During active it supplies the rail when there is no load. If there is load LDO_V30 takes over.

Table 14. Disabling bandgap clamp can cause a BOD event

Issue #	Effect
12.	Reset caused by the BOD.
	Conditions
	Always when disabling the Bandgap clamp.
	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.
	Workaround
	Proposed workaround is to disable the BOD before disabling the bandgap clamp and then enable it again.

Table 15. SNC hready not properly connected to the AHB2APB bridge

Issue #	Effect
13.	Leads into failing writes when wait cycles are introduced on the bus.
	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.
	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).
	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.

Table 16. eMMC memory compatibility

Issue #	Effect
14.	Incompatibility with eMMC devices.
	Conditions
	The selected eMMC device is not tolerant to identification clock rate higher than 400 kHz.
	Technical description
	Some eMMC devices require that the identification clock rate f_{od} does not exceed the max rate 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).
	Workaround
	Use eMMC devices that can operate even if f_{od} is higher than 400 kHz or temporarily use RCLP as system clock to initialize the eMMC device.

Table 17. Charger CC-to-CV comparator has low hysteresis

Issue #	Effect
15.	Charge Cc_mode and cv_mode signals toggle.
	Conditions
	Always.
	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 toggled. 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.
	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 is captured, provided that the respective interrupt mask register bit is enabled.

Table 18. Supply dips when switching over to retention LDOs

Issue #	Effect
16.	Supplies dips when switching over to retention LDOs.
	Conditions
	When switching over from normal LDO to Retention LDO.
	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 MV150 MV and are 50 µs-100 µs wide. When the normal LDO has a higher output voltage and both LDOs (normal and retention) are enabled, the normal LDO supplies the load. The retention LDO will settle to a non-working condition, meaning that the bias-point of the cell is 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 drops 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 is no drop. The reason for this is that the retention LDO does not shift to the non-working operating point in that case.
	Workaround
	A workaround for this behavior can be used when the normal LDO is trimmed to a value just below the output of the retention LDO.

Table 19. LDO hold causes jumps in ADC reference

Issue #	Effect
17.	A jump in LDO output (ADC Reference) occurs.
	Conditions
	When LDO_HOLD is used.
	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.
	Workaround
	Do not use LDO_HOLD.

Table 20. Boost converter output shows a sawtooth disturbance at certain loads/supply combinations

Issue #	Effect
18.	Boost converter output shows a sawtooth disturbance.
	Conditions
	At certain loads and input voltages.
	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.
	Workaround
	None.

Table 21. Lower SINAD during SRC Bypass mode

Issue #	Effect
19.	Lower Signal-to-noise and distortion ratio (SINAD).
	Conditions
	SRC Bypass mode.
	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.
	Workaround
	Do not use Output Bypass mode for signals above -1 dB FS when the signal degradation to 86 dB SINAD is unacceptable.

Table 22. LDO_VSYS voltage loop adds noise on VSYS rail

Issue #	Effect
20.	Noise on VSYS rail.
	Conditions
	LDO_VSYS in Regulating mode.
	Technical description
	An unexpectedly high noise level on node VSYS exists when the voltage loop of the LDO_VSYS is in Regulating mode. This noise has LF characteristics.
	Workaround
	None.

Table 23. LDO_VSYS headroom loop shows small oscillation of 400 kHz

Issue #	Effect
21.	A 400 kHz signal is detected in the VSYS rail.
	Conditions
	The headroom loop is in command in LDO_VSYS.
	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.
	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.

Table 24. Missing retention for Dcache WFLUSH done register

Issue #	Effect
22.	An unintended data cache flush operation might be started.
	Conditions
	Certain conditions after a (retained) sleep cycle.
	Technical description
	The write flush operation is started by software (writing a 1 into the DCACHE_CTRL_REG[DCACHE_WFLUSH] bit). This action forces 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, , 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. The dcache_wflushed_reg and the cfg_wflush_prev registers are undefined after a sleep cycle but resolves to a defined value after 2 HCLK cycles.
	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 cycle (with data retention enabled) execute the above sequence two times.
	The second write flush ensures that the write flush trigger bit is toggled twice (that is returning to 0) thereby avoiding unexpected write flush after a power cycle.

Table 25. Booter: UART not reaching 1 Mbaud

Issue #	Effect
23.	The booter misses intermediate bytes.
	Conditions
	UART boot at 1 Mbaud rate.
	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.
	Workaround
	Use speeds up to 500 kB.

Table 26. Booter: QSPI flash memories larger than 128 MB cannot be addressed

Issue #	Effect
24.	QSPI Flash Memories larger than 128 MB cannot be addressed by the Booter.
	Conditions
	Always.
	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.
	Workaround
	Use QSPI Flash devices up to 128 MB.

Table 27. RFMON fails to capture GPADC data when SYS_CLK is asynchronous to CMAC_CLK

Issue #	Effect
25.	RFMON fails to capture GPADC data.
	Conditions
	When SYS_CLK is async to CMAC_CLK.
	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 is sampled and can be corrupted.
	Workaround
	Run GPADC and CMAC on synchronous clocks.

Table 28. Timer4 IRQ is not connected to CMAC SYSPER IRQ

Issue #	Effect
26.	The IRQ line from Timer 4 to CMAC is missing.
	Conditions
	Always.
	Technical description
	The IRQ line from Timer 4 to CMAC is missing. A SW-based COEX interface that needs Timer4 cannot be implemented directly.
	Workaround
	The software workaround is to let Timer4 to trigger M33 and then M33 to notify CMAC through the SYS2CMAC_IRQ.

Table 29. Increased XTAL32M startup time at specific OFF time

Issue #	Effect
27.	Increased XTAL32M startup time.
	Conditions
	Residual energy in the resonator.
	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.
	Workaround
	The software 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.

Table 30. Wake-up from CMAC debugger is not functional

Issue #	Effect
28.	CMAC cannot wake up from debugger.
	Conditions
	Always.
	Technical description
	The logic that wakes 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.
	Workaround
	None.

Table 31. Enable signals for the BOD channels V18P and V18F are swapped

Issue #	Effect
29.	The BOD for the channels V18P and V18F cannot be used independently.
	Conditions
	Always.
	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
	Workaround
	Both signals should be enabled for the BOD to be working in Sleep mode. In any other case, a reset does not happen, even when the rail is too low.

Table 32. USB MAC Pclk frequency can go up to 120 MHz

Issue #	Effect
30.	USB MAC miss bytes.
	Conditions
	Pclk higher than 120 MHz.
	Technical description
	USB block is connected to two clock domains, pclk and 48 MHz. When the USB MAC sends a byte, the event is synchronized to the 48 MHz clock domain. An ack is generated at 48 MHz and synchronized 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.
	Workaround
	Pclk should be <120 MHz.

Table 33. RCLP glitch

Issue #	Effect
31.	Improper clocking of some digital blocks.
	Conditions
	Undefined.
	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.
	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.

Table 34. V18F high ohmic switch

Issue #	Effect
32.	This has an 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.
	Conditions
	Always.
	Technical description
	High resistance caused due to a layout issue on the switch that internally supplies V18F through V18P.
	Workaround
	V18F and V18P rails must be shorted externally and V18F switch must be forced always open.

Table 35. Display controller: data read failure in Read mode

Issue #	Effect
33.	Data read failure while display controller is in Read mode.
	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).
	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 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.
	Workaround
	● For the SPI3/4 and Dual/Quad SPI modes, a user must use the LCDC_STATUS_REG[SPI_RD_WR_OP] bit only. You 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.

5. Revision History

Revision	Date	Description
1.2	Oct 28, 2024	Converted into Renesas template. Table 16. is updated.
1.1	Sep 29, 2022	Sections 5.13-5.33 are added. Applies to final datasheet.
1.0	Jun 15, 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.

ROHS COMPLIANCE

Renesas Electronics' suppliers certify that its products are in compliance with the requirements of Directive 2011/65/EU of the European Parliament on the restriction of the use of certain hazardous substances in electrical and electronic equipment. RoHS certificates from our suppliers are available on request.