STR91xF
Errata sheet

Rev B and D limitations and corrections

Silicon identification

STR9 microcontrollers have two major salestype groups, as follows:

- The initial group consists of devices STR91xFxxxxx, for example STR912FW44X6, with silicon revisions B, and D. Detailed technical information for these devices may be found in the STR91xF datasheet and STR91xF reference manual documents.
- The second group consists of devices STR91xFAxxxxx, for example STR912FAW44X6, with a later silicon revision. Detailed technical information for these devices may be found in the STR91xFA datasheet and STR91xFA reference manual documents.

This errata document covers the first group with rev B and rev D.

Table 1 summarizes the marking to assist identification. *Figure 1* through *Figure 3* represent where the physical marking may be found on the devices.

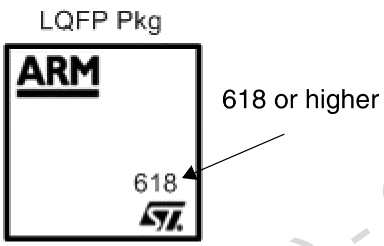
Table 1. Device identification

Salestype Group, External marking	Internal Silicon Revision	External Marking	Note
STR91xFxxxxx	B	ES	see *Figure 1*. Initial Engineering Samples. LQFP128 pkgs.
	D	ES_D	see *Figure 2*. Rev D Engineering Samples before week 18 of 2006. LQFP128 packages.
		Date code 618 or later	see *Figure 3*. Production Rev D Devices on or after week 18 of 2006. LQFP80 and LQFP128 packages. Date code format = YMM.

Figure 1. Device marking for Revision B Engineering Samples, LQFP128 packages

Figure 2. Device marking for Revision D Engineering Samples, LQFP128 packages

Figure 3. Device marking for Revision D Production Devices, LQFP80 and LQFP128 packages



Contents

1 Product evolution

This table summarizes the fix plan status for the next silicon revision (● = fix).

Table 2. Product evolution summary

Section	Limitation	Fixed in Rev D	Fixed in STR91xFA Series
Section 2.1	*VIC interrupt controller wrong vector fetch*	●	●
Section 2.2	*Motor control emergency pin*	● 1 and 2; workaround for 3	●
Section 2.3	*USB CRC computation*	●	●
Section 2.4	*BSPI transmit DMA request*	●	●
Section 2.5	*I2C BUSY bit not cleared by bus error*	●	●
Section 2.6	*UART error handling*	●	●
Section 2.7	*PFQBC/LDMA load multiple (performance related)*	●	●
Section 2.8	*RTC periodic interrupt*	workaround	workaround
Section 2.9	*PLL register default value for 48 MHz clock out of spec.*	workaround	●
Section 2.10	*Flash memory read configuration register*	●	●
Section 2.11	*ISP programming speed*	●	●
Section 2.12	*Flash memory device security*	●	●
Section 2.13	*Flash memory cannot be uploaded by JTAG tool*	●	●
Section 2.14	*Clock control unit clock switching*	workaround	●
Section 2.15	*SRAM discharge delay after tamper event*	●	●
Section 2.16	*Flash memory status register bit 7*	workaround	workaround
Section 2.17	*Sleep mode / Idle Mode*	workaround	●
Section 2.18	*Flash memory sector protection*	workaround	workaround
Section 2.19	*System reset at 96 MHz*	workaround	●
Section 2.20	*Waking up from Sleep Mode*	workaround	workaround
Section 2.21	*Flash memory erase and programming*	workaround	●
Section 2.22	*Sleep Mode Current (I_{SLEEP})*	no solution	●
Section 2.23	*Exit from Sleep and Idle Mode*	workaround	●
Section 2.24	*Sleep and Idle Mode Timing Requirements*	workaround	workaround
Section 2.25	*Wake Up Event Configuration*	workaround	workaround
Section 2.26	*ETM (Embedded Trace Module) Configuration*	workaround	workaround
Section 2.27	*Disabling the Watchdog*	no solution	●

Table 2. Product evolution summary

Section	Limitation	Fixed in Rev D	Fixed in STR91xFA Series
Section 2.28	*LVD (Low Voltage Detect) and VDD fall time requirement*	workaround	●
Section 2.29	*LVD Logic may hold reset active inappropriately*	workaround	●
Section 2.30	*ADC interrupt generation*	workaround	workaround
Section 2.31	*Motor Control output polarity*	no solution	●
Section 2.32	*16-bit pre-scaled clock source for general purpose timers*	no solution	workaround
Section 2.33	*ADC input range*	no solution	spec changed

2 Silicon limitations

2.1 VIC interrupt controller wrong vector fetch

Description of limitation in Rev B

The CPU may read an incorrect interrupt vector if an asynchronous interrupt happens on VIC0 while there is a pending vectored interrupt on VIC1

Fixed in Rev D. Swapped asynchronous USB Resume interrupt input channel with synchronous CCU interrupt.

New channel assignment in rev D:

USB Resume Interrupt is moved to VIC1 channel 30 (was on VIC0, channel 10),

Clock Control Unit CCU Interrupt is moved to VIC0 channel 10 from VIC1 channel 30

2.2 Motor control emergency pin

Description of limitation in Rev B

1. Emergency Stop (EST) pin (P6.7) is always connected as an input. Emergency Stop can be activated even if the pin is configured for a different IO function.
2. The Emergency Stop input signal is active-high (a typical emergency stop input is active-low, open drain).
3. The PWM outputs can be resumed by writing 0x4321 in the MC_ESC register and the motor will restart, even if the emergency pin is still active.

1 and 2 fixed in Rev D.

Added bit 6 (EST_Dis) to MC_PCR1 register. When set, the bit blocks EST input from GPIO.

The EST input polarity is changed from active-high to active-low.

Workaround for 3 using Rev D

- Read EST pin until it is de-asserted
- Mask FIQ/IRQ to prevent the next two write operations to the MC_ESC register from being interrupted
- Write 4321h to MC_ESC register to restart the PWM
- Write 0000h to MC_ESC register to re-arm the EST
- Unmask FIQ/RIQ

Item 3 is fixed in STR91xFA series devices.

2.3 USB CRC computation

Description of limitation in Rev B

When a packet is received with one specific wrong CRC (very rare), the packet is acknowledged whereas no acknowledgement should be sent.

This issue occurs when the CRC computation result is completed prematurely, just before the last bit of the packet, and the packet is accepted.

Fixed in Rev D. The CRC computation result is compared after the reception of the last bit, eliminating the possibility of a premature CRC calculation.

2.4 BSPI transmit DMA request

Description of limitation in Rev B

The DMA Requests signals are active at power-up and cleared only when the SPI block is enabled. This can cause spurious DMA requests if the BSPI DMA channels are configured before enabling the BSPI block.

Fixed in Rev D. The DMA Transmit Requests from BSPI are cleared (de-activated) at reset.

2.5 I2C BUSY bit not cleared by bus error

Description of limitation in Rev B

The BUSY bit is set by a Start condition and cleared by a Stop condition. But if a Bus Error occurs, the BUSY bit is cleared even if there is still activity on the bus.

Fixed in Rev D. The BUSY bit is not cleared inappropriately by a Bus Error.

2.6 UART error handling

Description of limitation in Rev B

When there are 2 consecutive frame errors or break errors, the error flags in the Status register are not set by the second one.

Fixed in Rev D. A second consecutive frame or break error will set the appropriate flags in UART status register.

2.7 PFQBC/LDMA load multiple (performance related)

Description of limitation in Rev B

When the LDMA instruction is executed, multiple CPU registers are loaded with data from the Flash program memory. This instruction fetches data from the Flash memory by issuing a Read bus cycle for each of the data words.

Fixed in Rev D. The LDMA instruction fetches the data from the Flash memory in burst mode, one clock per word instead of one bus cycle per word. This greatly reduces the instruction execution time.

2.8 RTC periodic interrupt

Description of limitation for all silicon revisions

When reading the Status register for the Periodic Interrupt flag, the CPU may occasionally miss the flag's set status.

Workaround

Do not poll the RTC interrupt flag. Always use the RTC interrupt signal to the VIC to cause a standard interrupt. This will not be fixed in future silicon revisions.

2.9 PLL register default value for 48 MHz clock out of spec.

Description of limitation in Rev B and Rev D

The default reset value of the SCU_PLL Configuration register should result in a PLL frequency of 48 MHz. However, the default values are incorrect.

Workaround using Rev B and Rev D

For a 48 MHz PLL clock, set the dividers P=3, N=C0h and M=19h in the SCU_PLL configuration register.

This is fixed in STR91xFA series devices.

2.10 Flash memory read configuration register

Description of limitation in Rev B

The Read Configuration register resides in Bank 1 (32KB) of the Flash memory. To write to the register, Bank 1 must be configured as 64KB in the bank size register.

Fixed in Rev D. Bit 13 of the Read Configuration register is redefined as a reserved bit, so it is permitted to configure Bank 1 as 32KB.

2.11 ISP programming speed

Description of limitation in Rev B

STR91xF devices contain 3 JTAG TAP controllers, which are daisy-chained. In this configuration, the JTAG clock may never exceed 1/8 of the frequency of the CPU clock. This severely limits the speed of programming the Flash memory during JTAG In-System-Programming.

Fixed in Rev D. The JTAG TAP for the ARM CPU may now be bypassed and the JTAG clock frequency is no longer limited to 1/8 CPU clock frequency. This is called “Turbo Mode” and JTAG clock can now equal CPU clock during Flash programming

See STR910 Flash Memory Programming Reference Manual.

2.12 Flash memory device security

Description of limitation in Rev B

The JTAG ISC-Erase instruction cannot erase the Flash memory once the security bit is programmed through the JTAG interface.

Fixed in Rev D. The JTAG ISC-Erase instruction has been modified. However, if the circuit board has other devices connected by the JTAG bus, the STR91xF device must be the first device in the JTAG chain.

See STR910 Flash Memory Programming Reference Manual.

2.13 Flash memory cannot be uploaded by JTAG tool

Description of limitation in Rev B

The JTAG ISC-Read instruction always sets bit 2 and 3 in the field of a Flash page to "1". The uploaded data from Flash is incorrect.

Fixed in Rev D. The JTAG ISC-Read instruction has been modified. However, if the circuit board has other devices connected by the JTAG bus, the STR91xF device must be the first device in the JTAG chain.

See STR910 Flash Memory Programming Reference Manual.

2.14 Clock control unit clock switching

Description of limitation in Rev B and Rev D

When switching the main clock (f_{MSTR}) from PLL clock source to OSC clock source, NOP instructions must be inserted after the OSC is selected and before the disabling of the PLL. Without the NOPs, the CPU will lose its clock and a warm, external reset will not re-boot the processor. The number of NOPs needed depends on the PLL frequency. The total NOP execution time must be greater than 10 OSC clock periods (one PLL clock is needed to execute one NOP).

Workaround using Rev B and Rev D

1. To prevent the CPU from losing its clock, do not disable the PLL while it is the selected main clock source.
2. When configuring the PLL for a different frequency, it is recommended to:
   a) Select the OSC as clock source
   b) After inserting the required number of NOPs, disable the PLL
   c) Enter new P, M and N values in the SCU-PLL Configuration register
   d) Enable PLL
   e) Select PLL as clock source
3. If you have programmed the device with the incorrect clock switching codes and the processor has no clock source to run on, you can still reprogram the Flash with a JTAG

tool. The JTAG tool will directly control the board reset line to program the device. The programming steps are:

a) Run JTAG programming application software
b) Go to "Program Device"
c) Power-off the board
d) Click "Assert Reset" button in JTAG software tool
e) Power-on the board
f) Perform any programming operation
g) When done, click "De-Assert Reset" button

This is fixed in STR91xFA series devices.

2.15 SRAM discharge delay after tamper event

Description of limitation in Rev B

SRAM discharge could take a few msec after a tamper event has been detected on TAMPER-IN pin.

Fixed in Rev D. New design results in quick discharge of internal SRAM power when power is turned off by a Tamper event.

2.16 Flash memory status register bit 7

Description of limitation in Rev B and Rev D

Status Register Bit 7 (ready bit) does not reflect the correct status when it is read immediately after the CPU issues a Flash memory Program or Erase command.

Workaround using Rev B or Rev D

Set bit 18 (write order bit) in the Configuration Control register of the ARM966 core. After this, it is ensured that the write command to the Flash memory occurs before the read command of the status bit.

This will not be fixed in future silicon revisions.

2.17 Sleep mode / Idle Mode

Description of limitation in Rev B and Rev D devices

The RTC crystal must be connected to the device in order to enter Sleep mode or Idle mode.

Entry to Idle Mode fixed in STR91xFA series devices.

2.18 Flash memory sector protection

Description of limitation in all silicon revisions

At power up, the Flash Protection Level 1 register is reset to 0FFFh (all flash sectors are protected). A warm reset does not reset the register.

Workaround

Firmware must change the values if desired.

This will not be fixed in future silicon revisions.

2.19 System reset at 96 MHz

Description of limitation in Rev B and Rev D (limitation does not apply to Rev D device with date code 618 and later)

When the CPU is fetching instruction from the Flash memory at 96 MHz FMI clock, the flash memory must be configured to operate at 2 wait states.

When a system reset occurs, the CPU clock control registers remain unchanged and the FMI clock keeps operating at 96 MHz. However, the Flash Configuration Register is reset by the system reset from 2 wait states to 1 wait state and the Flash memory is too slow for the CPU. The source of the system reset is either the external reset or the Watchdog reset.

Workaround using Rev B and Rev D with date code less than 618

The CPU hangs after the system reset occurs. A Power Off is the only way to re-start the system. Work around includes:

1. CPU can run at 96 MHz but the FMI clock must be configured to run at 48 MHz in the SCU_CLKCNTR register.
2. Do not reset the STR91xF while the FMI clock frequency is above 66 MHz.

This is fixed in Rev D devices with date code 618 and later. It is also fixed in STR91xFA series devices.

2.20 Waking up from Sleep Mode

Description of limitation in all silicon revisions

After the CPU enters Sleep Mode, it can be woken up by:

1. External interrupt
2. RTC/USB interrupt
3. External reset

When an oscillator chip is used as the clock source for the STR91xF, the CPU wakes up from Sleep Mode following any of the above three input events. If a crystal is used as the clock source, the crystal is disabled in Sleep Mode to save power consumption. When a wake-up event occurs, the crystal will not recover fast enough and the CPU hangs.

Workaround

Workaround solutions include:

1. Use the 32 kHz RTC clock as the clock source for Sleep Mode:
   a) Select the RTC clock as the CPU clock source prior entering Sleep Mode
   b) The CPU wakes up following any of the 3 wake up events and waits for the crystal to start oscillation. A crystal start up time is about 1.5 ms typical.
   c) After the crystal wakes up, the CPU waits for another Twait time before the first code is fetched from flash memory. The software can then change the CPU clock source back to the OSC or PLL clock. The length of Twait depends on the crystal frequency. Twait is 50us at 25 MHz and is 312 us at 4 MHz.
2. Instead of a crystal, use an oscillator as the STR91xF clock source.

This limitation will not be fixed in future silicon revisions.

2.21 Flash memory erase and programming

Description of limitation in Rev B and Rev D

The Dual Bank Flash architecture in the STR91X supports the modification (erase or programming) of one Flash bank while the CPU is fetching codes from the other bank.

This feature in the STR91x is functional only when the Flash bus clock (FMI Clock) frequency is 25 MHz or lower. At higher clock frequency, the CPU may read incorrect code or status from the Flash banks. The Flash modification problem affects both the dual banks and the OTP sector.

Note: This limitation has no impact on the Flash erase or programming through the JTAG port.

Workarounds using Rev B and Rev D

1. Re-configure the STR91X clock when Flash erase or programming is needed. The CPU clock (RCLK) must be configured to be 50 MHz or less, the RCLK is then divided by 2 to provide a 25 MHz FMI clock. The CPU can return to full speed at 96 MHz after the erase or programming operation is completed.
2. Another workaround is to copy the erase/programming routine from the Flash memory to the SRAM; the CPU then branches to SRAM and executes the routine. The advantage in this workaround is the CPU clock and the FMI clock can remain at 96 MHz during the Flash erase or programming.

This limitation is fixed in STR91xFA series devices.

2.22 Sleep Mode Current (I_{SLEEP})

Description of limitation in Rev B and D

When the STR91x enters Sleep Mode, the device consumes very little current (I_{SLEEP}). The typical I_{SLEEP} current is specified to be 55 µA (LVD On) and 50 µA (LVD Off). However, the Rev B, D and E devices consume more than the specified values. The typical I_{SLEEP} is 700 µA and the maximum at 1000 µA, for both LVD On or Off.

The Sleep Mode current consumption will be fixed in the STR91xFA series device.

2.23 Exit from Sleep and Idle Mode

Description of limitation in Rev B and Rev D

After the STR91X enters into Idle or Sleep Mode, the CPU can be woken up by:

1. Interrupts (internal or external)
2. Resets (internal or external)

The current device may fetch incorrect instruction from the Flash memory after waking up by interrupt. The exit from Sleep or Idle mode is therefore limited to wake up event from Reset; wake up from interrupt is not functional. When the CPU is woken up by a Reset event, the CPU fetches code from location zero in the Flash memory.

Workarounds using Rev B and Rev D

If waking up from Interrupt is a requirement, the workaround is as follows:

1. Before going into Idle or Sleep mode, copy the power mode switching routine from the flash memory to the SRAM.
2. Branch to SRAM and execute the switching routine.
3. After waking up, the CPU resumes execution from the SRAM.
4. If the wake up event is an interrupt, the CPU will branch to the Interrupt Service Routine which resides in the Flash memory.

This problem is fixed in STR91xFA series devices.

Note: When performing Sleep mode switching, the CPU must run on the RTC clock. Refer to Errata Item Section 2.20: Waking up from Sleep Mode.

2.24 Sleep and Idle Mode Timing Requirements

Description of limitation in all silicon revisions

1. Code Execution after Setting the Sleep or Idle Mode Bit:

   Once the Idle or Sleep modes are entered by writing the PWR_MODE[2:0] bits in the SCU_PWRMNG register it takes about 12 f_{OSC} cycles for the device to stop the execution.

Workarounds using Rev B and Rev D

In order to avoid executing any valid instructions after the Idle or Sleep bit setting and before entering the mode, it is mandatory to execute a certain number of dummy instructions after the SCU_PWRMNG register setting. The number of dummy instructions to be executed is:

No_dummy_Instr = (f_{CPUCLK}/f_{OSC})*12

Where f_{CPUCLK} is the CPU core clock frequency and f_{OSC} is the oscillator frequency. The worst case is represented by the core working out of the PLL maximum frequency (96Mhz) with an 4MHz crystal or oscillator on the X1_CPU input. In that case 288 dummy instructions would be needed.

Note: If (f_{CPUCLK}/f_{OSC}) is less than 1, the number of dummy instruction is always 3.

This limitation will not be fixed in future silicon revisions.

2. Time Required for Sleep Mode Entry:

After the mode bit is set in the SCU_PWRMNG register, the Power Management Unit requires a period of time (Tsleep) to switch off all the CPU and peripheral clocks safely before entering Sleep Mode. If a peripheral is running on a very slow clock, that would result in a longer switch off time.

The Tsleep time required to enter Sleep mode depends on the frequency of the oscillator, the CPU clock and the slowest peripheral clock. If a wake-up event occurs within this Tsleep time, it will be ignored and the STR91x will not exit from Sleep mode. The Tsleep time can be expressed in term of the T period of these clocks:

Tsleep = 17*(T_OSC) + 14*(T_Slowest_IP_CLK) + 6*(T_CPUCLK)

Example to calculate Tsleep (CPU is running on RTC clock before entering Sleep mode, f_{CPUCLK} = 32 kHz)

T_OSC=40ns (f_{OSC}=25 MHz)

T_CPUCLK = T_RTC = 31,250ns (f_{CPUCLK}=32 kHz)

T_slowest_IP_CLK = 2*31,250ns

Assume all clock dividers are at the default state of 1 except APB clock divided by 2. The slowest peripheral clock is then = $f_{CPUCLK}/2$

Tsleep = 17*40ns + 14*(2*31,250ns) + 6*31,250ns ~ 1.06 ms

Workarounds for Rev B and Rev D

Take account of the maximum time required to enter Sleep mode (Tsleep) in your specific application. This limitation will not be fixed in future silicon revisions.

2.25 Wake Up Event Configuration

Description of limitation in all silicon revisions

In the Wake Up Unit, the 30 External GPIO Interrupt inputs, the RTC interrupt, and the USB Resume interrupt are all logically ORed together to form an output which can be configured by the WIU_CTRL register to be one of the following:

1. A WIU interrupt input to the Vectored Interrupt Controller (VIC1, channel 9) by setting the INT_EN bit in the WIU_CTRL Register
2. An event input to the Wake Up Unit to wake up the CPU from Idle or Sleep mode by setting the WKUP_INT bit in the WIU_CTRL Register
3. As both an interrupt and a wake up event

Silicon Limitation: The WKUP_INT bit is not functional; only the INT_EN bit is gating the output of the 32 ORed interrupts. When the INT_EN bit is set, the output is routed to the VIC1 channel 9 as well as to the Wake Up Unit. There is no distinction between a Wake Up event and a Wake Up interrupt. When a wake up event occurred, the CPU wakes up from Idle or Sleep mode and generates an interrupt as well.

Workarounds

If it is not desired to serve an interrupt after exiting from low power mode, especially in Sleep Mode, the user can disable the WIU interrupt channel 9 on VIC1 before entering the low power mode. The channel can be enabled again after the wake up.

This limitation will not be fixed in future silicon revisions.

2.26 ETM (Embedded Trace Module) Configuration

Description of limitation in silicon

By default, the ETM9 interface in the STR91X core is an 8-bit medium size model as defined by ARM Ltd. When an Emulator tool boots up and performs an automatic configuration on the ETM (ETM sniffers), it will receive from the Configuration register that it is an 8-bit model.

However, in order to reduce the number of I/O pins required for debugging, the ETM trace data port is implemented as a-4 bit port. When polled, the 8-bit status provided by the Configuration register is incorrect.

Workarounds

When booting up an Emulator, do not select the automatic configuration option. Instead, configure the ETM to be a 4-bit port manually.

This limitation will not be fixed in future silicon revisions.

2.27 Disabling the Watchdog

Description of limitation in Rev D

Watchdog Mode is enabled by setting the WE bit in the WDG_CR register. Once enabled the Watchdog cannot be disabled by software by clearing the WE bit.

In the System Control Unit, the following two registers also control the clocking and reset operation of the peripherals, including the Watchdog:

1. SCU_PCGR1 register bit 12 (WDG) - by writing a "0" to this bit, the Watchdog clock is disabled.
2. SCU_PRR1 register bit 12 (RST_WDG) - by writing a "0" to this bit, the Watchdog is put to reset state.

By setting any of the above two register bits, the Watchdog can be disabled and become not functional. This is in contradiction to the Watchdog specification that once it is enabled, it cannot be disabled by software.

This problem will be fixed in STR91xFA series devices by removing the above two control bits.

2.28 LVD (Low Voltage Detect) and VDD fall time requirement

Description of limitation in Rev D

The STR9 generates an internal reset signal when the VDD power supply drops below the LVD (Low Voltage Detect) threshold level. If the rate of fall of VDD voltage is less than 100us as VDD drops from 1.8 V through the 1.4 V threshold, the Low Voltage Detect logic will not generate the internal reset signal.

Workarounds using Rev D

An external reset supervisor device can be used to drive the RESET_INn input pin on the STR9. The LVD must be switched off in this case. This problem is fixed in STR91xFA series devices.

2.29 LVD Logic may hold reset active inappropriately

Description of limitation in Rev D

The STR9 LVD logic monitors internally the VDD and VDDQ voltage levels independently, and generates an internal reset when either voltage level drops below their respective threshold levels (1.4 V for VDD ; 2.4 V or 2.7 V for VDDQ, depending on LVD configuration specified in CAPS software tool or 3rd party development tools).

Silicon Limitation: The LVD logic is able to detect the first occurrence of VDD or VDDQ voltage drop that goes below the threshold voltage and it correctly generates an internal reset signal to the CPU and system. However, for subsequent drops of VDD or VDDQ voltage below threshold, the LVD logic generates an active internal reset signal, but the signal remains active (is stuck) even after VDD or VDDQ rises above the threshold again. The CPU is hung in a state of reset. A power cycle is needed on VDD to clear the stuck reset and release the CPU.

Figure 4 illustrates the first case, when VDD is steady, but VDDQ falls below its LVD threshold. You'll see that the generated reset signal during the first drop of VDDQ is correct, and the reset is correctly deactivated when VDDQ returns to normal voltage. However, the second, and subsequent drops, of VDDQ will cause an active reset which remains active even after VDDQ returns to normal level. Only after VDD voltage goes through a power cycle will the reset signal deactivate.

Figure 5 illustrates the second case, when VDDQ is steady, but VDD falls below its LVD level, but not below 1.2 V (power-on-reset level). You'll see that the generated reset signal during the first drop of VDD is correct, but the second and subsequent drops of VDD will cause an active reset which remains active even after VDD returns to normal level. Only after VDD voltage goes through a power cycle (drops below 1.2 V) will the reset signal deactivate.

Figure 6 illustrates the third case, when VDDQ is steady, but VDD falls below its LVD level, and falls below 1.2 V. In this case, there is no problem, the generated reset signal will activate and deactivate appropriately (will not get stuck).

Workarounds using Rev D

An external reset supervisor device can be used to drive the RESET_INn input pin on the STR9. The LVD must be switched off in this case. This limitation will be fixed in the STR91xFA series.

Figure 4. VDD steady, VDDQ falls below its LVD threshold

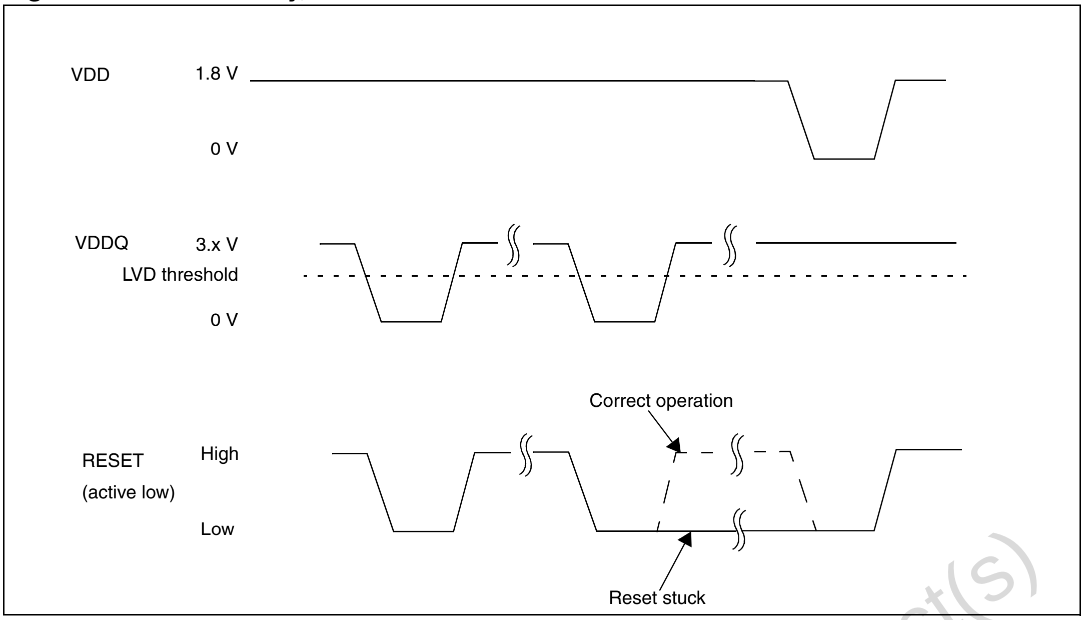


Figure 5. VDDQ steady, VDD falls below its LVD threshold, but not below 1.2 V

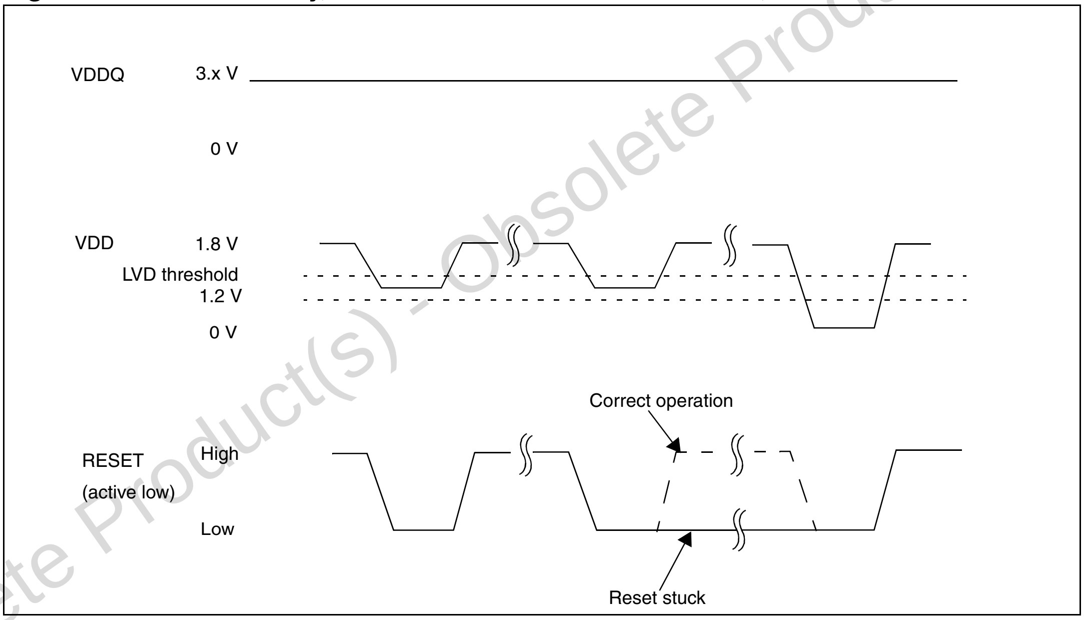
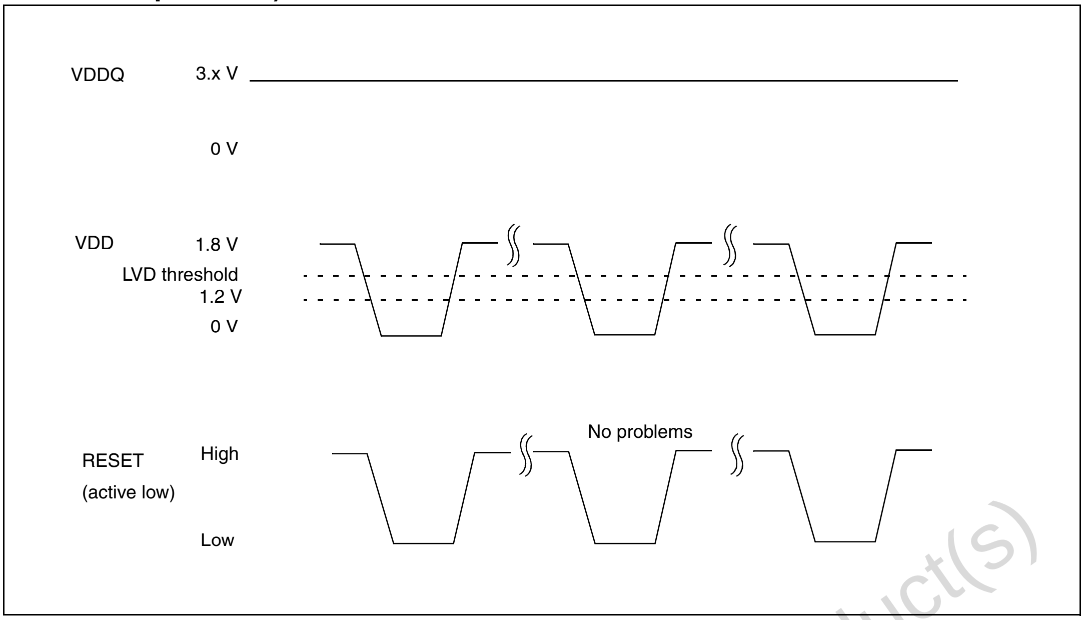


Figure 6. VDDQ steady, VDD falls below its LVD threshold, and below 1.2 V (no problems)



2.30 ADC interrupt generation

Description of limitation in Rev D

The ADC generates end of conversion (EVC) or analog watchdog (AWD) interrupt when enabled. Before returning from serving the interrupt, the ISR typically clears the interrupt by setting the corresponding EVC or AWD flag bit in the ADC_CR register to "0".

The ADC clock is used to clear the interrupt flags. The time it takes to clear the flags is longer when the ADC runs on a slow clock. There are situations where the CPU returns from ISR and the interrupt flag has not been cleared yet. Since the Interrupt Controller input is level sensitive, the CPU will see immediately another interrupt.

Workarounds using Rev D

Instead of clearing the ADC interrupt flag at the end of the ISR, clear the flag when ISR is entered.

This limitation will not be fixed in future silicon revisions.

2.31 Motor Control output polarity

Description of limitation in Rev D

The Motor Control (MC) drives 3-Phase PWM signals and complementary signals (UH, UL, VH, VL, WH, WL). There are two situations where the MC outputs remain a constant:

1. When the Compare register (MC_CMPx) value is greater than the Compare 0 Register (MC_CMP0), the corresponding PWM output signal is held at '1'.
2. When the Compare register (MC_CMPx) value is 0, the corresponding PWM output signal is held at '0'.

The output is a constant at "1" and "0" for the above two configurations when the polarity bits in the MC_PSR register are set to "0".

Silicon Limitation: The MC outputs (UH and UL, for example) are complementary to each other. But for the above two configurations the outputs are not complementary and both signals stay at the same level.

This problem will be fixed in the STR91xFA series.

2.32 16-bit pre-scaled clock source for general purpose timers

Description of limitation in Rev B and Rev D

The typical clock source for each of the four standard 16-bit timers is the APB clock (PCLK) through an 8-bit prescaler. Optionally, these timers can be clocked in pairs (TIM0/TIM1 and TIM2/TIM3) from a clock source coming directly from an external pin, or from the main system clock (f_{MSTR}) through a 16-bit prescaler.

Silicon Limitation: The clock source from f_{MSTR} through the 16-bit prescaler does not operate correctly and cannot be used. The 16-bit prescaler does not output the expected clock pattern.

The other two clock sources (APB through 8-bit prescaler and external pin) operate correctly.

Workaround using Rev B and Rev D

Instead of using the clock source from f_{MSTR} through 16-bit prescaler for the general purpose timers, use one of the other two clock sources.

This limitation will not be fixed in future silicon revisions.

2.33 ADC input range

Description of limitation

AV_{REF} must be 2.65 V or higher, and $AV_{DDQ} \geq A_{VDD} \geq AV_{REF}$. ADC accuracy is downgraded if AV_{REF} goes below these limits.

This limitation will not be fixed in future silicon revisions.

3 Revision history

Table 3. Document revision history

Date	Revision	Changes
12-Apr-2006	1	Initial release.
26-Jun-2006	2	Changed *Section 2.17 on page 11* Changed *Section 2.19 on page 12* Added *Section 2.21 on page 13* and *Section 2.22 on page 14*
04-Sep-2006	3	Changed *Section 2.18 on page 12* Workaround modified in *Section 2.20 on page 13* Added *Section 2.23 on page 14* - *Section 2.26 on page 16*
05-Feb-2007	4	Updated *Silicon identification on page 1* Added sections *Section 2.28 on page 17* - *Section 2.31 on page 19*
09-May-2007	5	Added *Section 2.32 on page 20* and *Section 2.33 on page 20*

Please Read Carefully:

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com