E-GOLDradio

GSM/GPRS Single Chip Solution

PMB 7870

From V2.1F to V2.2x

Secure Mobile Solutions

Never stop thinking.

Edition 2005-07-19

Published by Infineon Technologies AG,
St.-Martin-Strasse 53,
81669 München, Germany

Attention please!

The information herein is given to describe certain components and shall not be considered as a guarantee of characteristics.

Terms of delivery and rights to technical change reserved.

We hereby disclaim any and all warranties, including but not limited to warranties of non-infringement, regarding circuits, descriptions and charts stated herein.

Information

For further information on technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies Office (**www.infineon.com**).

Warnings

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies Office.

Infineon Technologies Components may only be used in life-support devices or systems with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Trademarks and Registered Terms

BlueMoon®, M-GOLD®, SIEGET®, SMARTI®, UCP®, and S-GOLD® are registered trademarks of Infineon Technologies AG.

BlueNix™, SCT™, S-GOLDlite™, S-GOLD2™, and S-GOLD3™, are trademarks of Infineon Technologies AG.

ARM®, and PrimeCell® are registered trademarks of ARM Limited.

ARM926EJ-S™, and ADS™ are trademarks of ARM Limited.

OakDSPCore® and TEAKLite® DSP Core are registered trademark of ParthusCeva, Inc.

E-GOLDradio

GSM/GPRS Single Chip Solution

PMB 7870

From V2.1F to V2.2x

Secure Mobile Solutions

Never stop thinking.

We Listen to Your Comments

Any information within this document that you feel is wrong, unclear or missing at all?
Your feedback will help us to continuously improve the quality of this document.
Please send your proposal (including a reference to this document) to:

smsdocu.comments@infineon.com

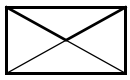

Table of Contents

1 E-GOLDradio Hardware Issues

E-GOLDradio CONFIDENTIAL Revision History:	**2005-07-19**	Rev. 1.02
Previous Version:	1.01, 2005-07-15	
Page	Subjects (major changes since last revision)	
Changes for Rev. 1.01		
	Removed UTP Issue: WS00005882: **I2S: Timing for Master Mode Not Correct**	
	Added UTP Issues: **WS00005901**, **WS00006687**, **WS00007440**, **WS00007736**, **WS00008364**, **WS00008373**, **WS00008476**, **WS00009022**, and **WS00009028**	
Changes for Rev. 1.02		
	UTP issues solved in E-GOLDradio from V2.1F to V2.2x: **WS00006687**, **WS00007440**, **WS00007600**, **WS00007724**, **WS00007736**, **WS00008085**, **WS00008097**, **WS00008142**, **WS00008373**, **WS00008476**	
	Added UTP Issue: **WS00008988**	

1.1 Overview of Hardware Issues

Table 1-1 Overview of HW Issues

Type	Title	ID
DSP	**BBRX: The Adjacent Channel Power Estimation Is Biased By Transients**	WS00005865
DSP	**AFE: Noise Signals At Earpiece Buffer Output During Buffer Power Down**	WS00005868
DSP	**I2S: WA0/1 Signal Is Running While TX/RX Is Not Started**	WS00005876
DSP	**I2S2: RX Path Buggy (polarity bit = 0, Normal Master Mode in 6-Pin Config)**	WS00005877
DSP	**I2S: RX PCM Does Not Work After Restart**	WS00005878
DSP	**I2S Doesn't Always Work in Burst Mode**	WS00005880
MCU	**SIM: SIM_IN IRQ Illegally Occurs After Reset**	WS00005889
MCU	**SIM: 'Ghost Character' after Retransmission at Parity Error**	WS00005893
DSP	**SSC DSP: Interrupt Bug during Transparent Mode**	WS00005897
MCU	**Corruption of PEC src/dst Segment Address When Overridden by Other PEC/INT**	WS00005899
MCU	**Dummy RTC_INT/RTC_T14 Interrupts After Reset Phase**	WS00005900
MCU	**RTC: Power Consumption and Current Peak**	WS00005901
MCU	**PM_INT: the Associated Interrupt Is Missing**	WS00005903
MCU	**EBU: Not Enough GPIO Can Be Used in VDD_EBU Domain (Mainly for WP of Flash)**	WS00005905
Top Level	**I2S: Burst Mode Not Capable Of Quasi-Continuous Transmission**	WS00005909
MCU	**SIM: Auto Powerdown Not SIM Standard Compliant**	WS00005911
DSP	**RTC: Alarm Interrupt Generation**	WS00005913
MCU	**CR110651: Delayed ILLOPA Trap Execution If Caused by Double-Indirect MOV Instruction**	WS00005918
MCU	**CR110652: Delayed ILLBUS Trap Execution When Writing External Bus**	WS00005919
MCU	**CR110653: Visible Mode Limitation for Read Accesses on X-Bus**	WS00005920
MCU	**CR110654: DIP Shows Incorrect Value If Breakpoint Is Inside Atomic/External Sequence**	WS00005921

Table 1-1 Overview of HW Issues

Type	Title	ID
MCU	CR110655: Break-after-Make Does Not Work Correctly in Atomic Scenarios	WS00005922
MCU	CR110656: Read Address Triggers Cannot Be Generated for All Instruction Reads	WS00005923
MCU	CR110657: ILLBUS Exception Inside ILLBUS ISR Problem	WS00005924
MCU	CR110764: External Break Events (BREAK Pins) Can Be Lost	WS00005925
MCU	CR111192: Bit-Manipulating Instructions on EXICON/EXISEL Write Wrong Values	WS00005926
MCU	CR112452: Wrong SP Used with SCXT Reg, Mem, and Mem Pointing to SP	WS00005927
Backend	RTC: Internal Oscillator Instability	WS00006183
DSP	TEAKlite: Loop (rep) Ends Unexpectedly After Interrupt in PROM Page 1	WS00006189
Top Level	DSP: Consecutive Write Operation to Some Registers Fails	WS00006272
MCU	Boot Code: Access in External Flash during the BOOTROM	WS00007413
Physical Implementation	EBU: Improve Timings for EBU Interface	WS00007486
MCU	GSM Timer Unit: TFSKIP.SKIPC Bit Has No Effect	WS00007537
MCU	SCCU X_BUS Clock at 78 MHz	WS00007621
Evaluation Board	SSC: Phase Error When High Baudrate Is Set	WS00008268
Verification	Cannot Make Write Access to Internal Memory Immediately After Resetting MST_CLK_CTRL.CPUH	WS00008364
Top Level	CS2 Usage Restriction	WS00008988
Verification	C166S: Jump-Bit Executed Wrongly after Byte Manipulation	WS00009022
Verification	CGU: Phaseshifter Startup Problems Under Certain Circumstances	WS00009028

1.2 Details of Hardware Issues

1.2.1 CS2 Usage Restriction

ID

WS00008988

Type

Top Level

Description

The external CS2 signal is an alternate function of the pad OE.

After reset this pad has the behavior of the OE function, that means if a device is connected on this pad for CS2, there may be a conflict on the data bus if another device (Flash) is connected on CS0. During the boot, when an access to CS0 is done to check if a code is in the external Flash, the OE pin (CS2) is also active and the CS2 device can drive the data bus at the same time as the CS0 flash.

Because the page mode switch is not supported by E-GOLDradio V2.1F to V2.2x , the OE function is not needed and the RD pad can be connected to the OE Flash input for CS0.

Solution

Do not use CS2.

1.2.2 CGU: Phaseshifter Startup Problems Under Certain Circumstances

ID

WS00009028

Type

Verification

Description

It has been observed that a phaseshifter output clock may be missing immediately after switching on and removing the phaseshifter bypass for some variable time.

The problem has been seen with higher probability for a small voltage band (of around 40 mV) of the PLL + Core power supply at a given temperature.

If the missing clock is driving an active subsystem, the subsystem stops until the phaseshifter clock is started.

On E-GOLDradio, the problem has been observed only with the phaseshifter1. For phaseshifter2, Phs2_X only uses 0 and 2, so the problem has not been observed here.

System impact is low (when DSP clock is set to 78, 113.5 or 124.8 MHz).

Solution

1. Do not use values of Phs_X = 4 and Phs_X = 5 for phaseshifter2.
2. If a phaseshifter frequencies with Phs_X = 4 (or 5) is necessary in the system, the phaseshifter must be started using the following procedure (“switching on the fly”):
   a) Set PHX_CTRL.x to 3
   b) Power on the phaseshifter
   c) Wait for 1 µs
   d) Switch on-the-fly towards the needed settings (for example, PHX_CTRL.x = 4)
   e) Wait for 1 µs
   f) Activate phaseshifter clock by removing bypass.

1.2.3 C166S: Jump-Bit Executed Wrongly after Byte Manipulation

ID

WS00009022

Type

Verification

Description

AI00025116: Jump-bit wrongly executed after Byte manipulation instructions

When a JB/JNB/JBC/JNBS on a GPR follows an instruction that performs a byte write operation (MOVB, ADDB/ADDBC, ANDB, XORB, ORB, NEGB, CPLB, SUBB/SUBCB) on the same GPR but on the byte that is not been used by the JB/JNB/JBC/JNBS (that is, the byte where the bit used by the Jump is not located), the program flow gets corrupted. That means, the Jump may be wrong (or not done).

A side effect of this bug is that further program branches may also lead to illegal instruction fetches (fetches are performed from wrong addresses).

- Example 1

```
; Assume Rx.0 is 0 (x any GPR 0..7)
MOVB RxH, any_value   ; Any Byte write instruction on RxH
JB Rx.0, jump_address  ; WILL BE INCORRECTLY TAKEN
```

- Example 2

```
; Assume Rx.y is 1 (x any GPR 0..7; y any bit except bit0)
MOVB RxL, any_value    ; Any Byte write instruction on RxL for y= 8..15 or RxH for y= 1..7
JNB Rx.y, jump_address ; WILL BE INCORRECTLY TAKEN
```

- Example 3

```
; Assume Rx.0 is 0(x any GPR 0..7)
MOVB RxH, any_value    ; Any Byte write instruction on R0H
JNB Rx.0, jump_address  ; WILL BE INCORRECTLY TAKEN
```

- Example 4

```
; Assume Rx.y is 1(x any GPR 0..7; y any bit except bit0)
MOVB RxL, any_value    ; Any Byte write instruction on RxL for y= 8..15 or RxH for y= 1..7
JB Rx.y, jump_address    ; WILL BE INCORRECTLY TAKEN
```

Note: The bug is only visible when the bit used for the Jump evaluation is set (‘1’) for all the bits Rx.15 to Rx.1, or not-set (‘0’) for Rx.0.

- Example 5 (BUG NOT VISIBLE)

```
; Assume Rx.0 is 1 (x any GPR 0..7)
MOVB RxH, any_value   ; Any Byte write instruction on RxH
JB Rx.0, jump_address   ; WILL BE CORRECTY EXECUTED (TAKEN)
```

Note: The bug exist also when the Byte write operation writes into the GPR using indirect addressing mode or memory addressing mode. However, this kind of addressing modes for accessing GPRs are not really expected to be generated by a compiler. This situation includes also the use of PECB/DPECB which destination pointer points into a GPR (that is, data write is performed on a GPR), but this use of PECs is also not expected.

Solution

Workaround:

Include a NOP between any Byte write instruction on a GPR and a Jump-bit instruction on the same GPR but on a bit belonging to a different byte.

Note: Some more general and easier to implement workarounds can be also defined (however, with unnecessary increase of the code size).

For example:

- Workaround Alt 1:

Include a NOP between any Byte write instruction and a Jump-bit instruction (on a GPR).

This will cover the case when the Byte write instruction uses indirect addressing mode or memory addressing for accessing GPRs.

- Workaround Alt 2:

Before any jump-bit instruction on a GPR add an "Atomic#1" instruction.

This will cover the case when PECB/DPECB write into the GPR.

Workaround Alt 2 is then the most general, covering all possible cases when the bug can occur, however it is the most expensive in terms of code size and, therefore, is not recommended unless strictly necessary. A possible compromise is to implement "workaround" (or workaround alt 1) in the compiler and in addition to add an assembler checker for workaround alt 2.That is, with each JB/JNB/JBC/JNBS on a GPR warns about possible program flow corruption in case PECB/DPECB are used which destination pointer points into the same GPR used by the Jump-bit instruction (unless they are preceded by an atomic instruction).

1.2.4 Cannot Make Write Access to Internal Memory Immediately After Resetting MST_CLK_CTRL.CPUH

ID

WS00008364

Type

Verification

Description

When the CPU is at 78 MHz and codes are fetched from Internal RAM, to set clock back to 26 MHz: reset **MST_CLK_CTRL.CPUH** and make three write accesses to address 1 $BF00_H$ (Internal RAM), then verify the written result in this address.

The problem is that the value in address 1 $BF00_H$ is not the one expected. It seems that we cannot immediately after resetting **MST_CLK_CTRL.CPUH**, write accesses cannot be made to the IRAM.

Codes:

```
PUSH   R12
PUSH   R13
MOV    R12,#0FFFEh
EXTP   #03h,#1
AND    02D04h,R12
EXTP   #06h,#1
MOV    R12,03F00h
BFLDL  R12,#0Fh,#05h
EXTP   #06h,#1
MOV    03F00h,R12
EXTP   #06h,#1
MOV    R12,03F00h
BFLDL  R12,#0F0h,#0A0h
EXTP   #06h,#1
MOV    03F00h,R12
EXTP   #06h,#1
MOV    R12,03F00h
BFLDH  R12,#0Fh,#05h
EXTP   #06h,#1
MOV    03F00h,R12
EXTP   #06h,#1
MOV    R12,03F00h
BFLDH  R12,#0F0h,#0A0h
EXTP   #06h,#1
MOV    03F00h,R12
POP    R13
POP    R12
```

After these codes the value in address 1 $BF00_H$ should be $A5A5_H$, but it is $A505_H$.

Solution

Workaround:

- Add 20 NOPs after switching to 52 MHz
- Add 15 NOPs after switching to 78 MHz.

1.2.5 RTC: Power Consumption and Current Peak

ID

WS00005901

Type

MCU

Description

When the RTC is not in the Isolation mode, the module has a high power consumption and high current peak. Here are the values measured on the Evaluation Board:

Current measurement on E-GOLDradio:

RTC In Isolation	10.6uA
RTC with Aynchronous Access @ 26 MHz	86.6uA
RTC with Aynchronous Access @ 52 Mhz	241uA
RTC with Aynchronous Access @ 78 Mhz	351uA
RTC with Synchronous Access @ 26 MHz	337uA
RTC with Synchronous Access @ 52 MHz	651uA
RTC with Synchronous Access @ 78 MHz	939uA

Power consumption is too high on VRTC when RTC is in Synchronous mode. Current are above requirements for E-POWERlite.

The impact on overall consumption is limited.

Solution

To reduce power consumption, the RTC register interface has to be:

1. Enabled before the register access.
2. Disabled after the register access.

1.2.6 BBRX: The Adjacent Channel Power Estimation Is Biased By Transients

ID

WS00005865

Type

DSP

Firmware

Description

The bandpass filter (10th order IIR type), adaptively used during phases of high interference by adjacent channels, by its nature has a long settling time, which causes that the transient reaches into the (temporal) estimation window of the adjacent channel power (P_adj).

This leads to an erroneously too high P_adj value estimation.

Solution

Workaround:

The wrong estimated adjacent power value P_adj_HW is approximately compensated by a firmware patch according to following formula:

P_adj_FW = P_adj_HW + P_adj_correction

For a delay between BB_EQON and BB_RXON of 130us the parameter P_adj_correction is different depending on used RF. (e.g.: -200 (SMARTiDC+) and -800 (SMARTiSD)).

Solved in G14_ER_STARTUP_V11.

1.2.7 AFE: Noise Signals At Earpiece Buffer Output During Buffer Power Down

ID

WS00005868

Type

DSP

Description

When earpiece devices in VBRX are powered down (VEPPA = 0), a noise signal in band can be observed.

This disappears, if a resistive load of < 600 Ohms is connected between the outputs. This solves the problem with headphones etc., but there may be a problem with the capacitive coupling of the car kit.

This is a inherent phenomenon of the circuit, because in power-down the circuit output is high ohmic and, therefore, on- and off-chip noise signals can couple in path.

Solution

Connect resistance (less than 600 Ohms) to earpiece output.

1.2.8 I2S: WA0/1 Signal Is Running While TX/RX Is Not Started

ID

WS00005876

Type

DSP

Description

The problem occurs in Normal/Master mode if both the Tx and Rx paths derive their CLK from the same fractional divider (0/1).

Even if one path (Tx or Rx) is not started (TxStart or RxStart = '0'), the corresponding signals (CLK and WA, Tx) are generated and available on the respective pins.

Solution

The FW has to ignore the interrupts from the unused transfer direction.

1.2.9 I2S2: RX Path Buggy (polarity bit = 0, Normal Master Mode in 6-Pin Config)

ID

WS00005877

Type

DSP

Description

The I2S2 RX path does not receive the right values when it's configured as a master in the Normal mode in the 6 pins configuration.

It's due to the polarity bit (POL) in the RXCONF register. If the polarity bit is set to 1, the receive path is OK. If it's set to 0, the first 4 values are wrong in the RX buffer.

Solution

Infineon FW does not use the 6 pin configuration.

If the 6 pin configuration is used, the FW must delete first four data words.

1.2.10 I2S: RX PCM Does Not Work After Restart

ID

WS00005878

Type

DSP

Description

The I2Sx does not receive any data in PCM mode after the following sequence:

1. Set all configuration registers and start data transfer
2. Wait for the RX interrupt, data transfer stops because of PCM mode
3. Reset I2SON bit in I2SX.CTRL register
4. Set I2SON bit and start data transfer again.

The first transfer works correctly, but after switching off and on the I2S interface hangs.

Solution

SW workaround:

The internal signal can be reset if the I2SControl register is cleared completely when disabling the I2S interface. Use the following sequence:

1. Set all configuration registers and start data transfer
2. Wait for the RX interrupt, data transfer stops because of PCM mode
3. Reset I2SControl register instead of I2SON bit
4. Set I2SControl register and start data transfer again.

1.2.11 I2S Doesn’t Always Work in Burst Mode

ID

WS00005880

Type

DSP

Description

The delay (CLK/WA) placed on the Evaluation Board for Normal Mode should also be valid for Burst Mode, but this is not quite correct. Running the Burst Mode correctly implies that the register TxCONF and RxCONF should be set correctly - yet according to the spec these two registers are not relevant for the configuration of the Burst Mode.

Solution

-

1.2.12 SIM: SIM_IN IRQ Illegally Occurs After Reset

ID

WS00005889

Type

MCU

Description

Because of synchronization stages are included for Pad CC_IN signal the inactive value ('1') produces SIMIN_INT IRQ two cycles later when reset has already ended. This means that an IRQ is always generated.

This fact results in the following behavior at SIM configuration:

After the first configuration of pad SMC_IN(=CC_IN) as input:

A. A SIM_IN interrupt is generated if no SIM card is in the simcard holder.

B. A SIM_IN interrupt is not generated if there is a SIM card in place.

Solution

After reset, the SIMIN_INT.IR bit has to be cleared.

No fix planned in hardware.

Information

After reset IRQ-Bit has to be erased.

No fix planned in hardware.

1.2.13 SIM: ‘Ghost Character‘ after Retransmission at Parity Error

ID

WS00005893

Type

MCU

Description

If the SIM block is transmitting and a parity error is detected, after the retransmission of the character a 'ghost character' is observed. This is valid for both the T=0 and Character Modes.

Solution

To avoid the observed behavior, the SIM Receive Spacing register SIMTXSPC has to be set to 0003_H.

1.2.14 SSC DSP: Interrupt Bug during Transparent Mode

ID

WS00005897

Type

DSP

Description

During the Transparent Mode, the SSC does not generate transmit interrupts when writing in the TXB register. Interrupts are generated when a value was transmitted (equal to standard mode).

Solution

In the Transparent Mode interrupts are generated as in the standard mode. This means that there are no transmit interrupts generated when writing in the SSCDSP_TXB register in the Transparent Mode.

1.2.15 Corruption of PEC src/dst Segment Address When Overridden by Other PEC/INT

ID

WS00005899

Type

MCU

Description

The issue occurs on PEC reception transfers from the SIM card. The missed character caused a SIM overrun error that could not be recovered. This issue has also been observed with the camera and display interfaces and is likely to be the source of occasional trace corruption (on ASC1).

When PECs/INTs are requested from the CPU too close to each other, in some special situations the PEC Segment Address of the source/destination pointer can be corrupted. That is, the Segment portion of the address takes the value from the following PEC/INT.

The conditions for this problem are:

1. The CPU is about to process an PEC (first event) but can't do it immediately (due to a data dependency problem). So it delays its processing until the data conflict has been solved (by cancelling the PEC instruction in the Decode stage and re-injected it again later).
   The PEC is however immediately acknowledged by the interrupt controller so that the IC can start arbitrating other events.
2. By the time the first PEC should be re-injected, another PEC or INT is also requested from the CPU (second event). In this case the CPU Finnish first the first PEC correctly, however this is not consistently implemented, because the PEC Segment Pointer (PECSNx) that is used is the one corresponding to the second PEC/INT, that is, the address is wrong.

Example:

```
mov 0FCE4h, R9     ; any instruction modifying SRCPx (in this example SRCP1-0FCE4-) with direct or indirect
                   ; addressing mode
mov [R0+], [R1]    ; injected PEC (PEC1 in this example) PEC1 will be canceled and reinjected due to the
                   ; instruction above modifying its SRC pointer. This PEC takes as Segment pointer the one
                   ; corresponding to PEC8 ->WRONG
mov [R0+], [R8]    ; injected PEC (PEC8 in this example)
```

There are 3 situations where a PEC cannot be immediately be processed and is cancelled and reinject again (condition necessary for the bug to happen):

1. The PEC is injected right after an instruction that modifies the PEC Source Pointer (see Example)

2. The PEC Source Pointer points to the PSW register and is injected right after any instruction modifying the PSW flags (almost all instructions modify the PSW flags)

3. The PEC is injected right after an instruction that modifies the CP explicitly (through a SCXT, MOV, etc.).

Note:

When PEC Segment Pointers are 0, i.e. only 64 Kbytes regions are addressed by PECs, the problem is not visible.

Also, when the PEC Segment Pointers of all the PEC channels (or at least the ones that can happen simultaneously) have the same value, the problem is not visible (in this case, the PECSNx for nodes that trigger an INT instead of a PEC must also have PECSNx programmed to the same value as the one used for the PECs).

Additional Note:

Situation 3 was the only one actually seen in the field.

The workaround is implemented on the protocol stack using a Perl script processing the RTOS (OSE) generated assembly code.

It should be implemented in both the RTOS kernel and the C166 C compiler.

Solution

Workarounds

For situation 1:

Disable all interrupts while modifying src/dst PEC pointers or include these instructions in atomic sequences with an extra instruction after them (a NOP or any instruction not modifying these registers).

Example:

```
atomic #2          ; WORKAROUND
mov 0FCE4h, R9     ; any instruction modifying SRCPx with direct or indirect addressing mode
NOP                ; WORKAROUND (any instruction not modifying SRCPx's)
```

For situation 2:

No real workaround, just do not use PECs with a source pointer to PSW.

For situation 3:

Include any explicit modification of the CP within an atomic-3 sequence (this workaround may be selectively implemented on code sequences where the bug can occur, i.e., at least PECs can occur, different segment pointers are used).

Example:

```
atomic #3          ; WORKAROUND
scxt cp,#new_cp    ; any instruction modifying CP explicitly
...                ; any instr not modifying CP explicitly
...                ; any instr not modifying CP explicitly
```

Notes:

1. *A patch (build 234.1.10) is available from Tasking for the A166 assembler and C166 Compiler, version 7.5r5. The new assembler and compiler options make sure the PEC problem does not show up.*
2. *Compiler Version 8.5 includes the CHECKPECCP switch, which avoids the PEC problem. There is no patch or switch available for Tasking Version 8.0.*
3. *The effect of the bug is that the SRC and DST pointers can be corrupted. The workaround avoids this situation. It does not need to be applied to modifications of DSTPx pointers (only the SRCPx pointers).*

1.2.16 Dummy RTC_INT/RTC_T14 Interrupts After Reset Phase

ID

WS00005900

Type

MCU

Description

Due to a incorrect reset implementation some dummy interrupts on RTC_INT and RTC_T14 may be generated as soon as the reset is inactive.

Just after the HW reset phase, the RTC block is still in an unstable state because this block is reset afterward by a SW event: so RTC_INT and RTC_T14 maybe '0' or '1'.

If RTC_INT is set to '0', the reset phase is left without detecting any interrupts.

If RTC_INT is set to '1', the reset phase is left and issues a dummy interrupt to the C166S is issued.

This also applies to RTC_T14.

Solution

At startup (before enabling the interrupts):

Clear interrupt bits T14IR, RTC0IR,RTC1IR, RTC2IR and RTC3IR in RTCISNC register.

Set CLR_RTCINT bit in RTC_CTRL register.

1.2.17 PM_INT: the Associated Interrupt Is Missing

ID

WS00005903

Type

MCU

Description

There is no interrupt vector for the pad PM_INT.

Solution

Workaround: Use S0ASIC as PM_INT interrupt. The S0ASIC interrupt is generated as soon as wakeup signal is asserted.

If "wakeup on external interrupt" in the SCCU_HWWAKEUP register is enabled, each time the pad PM_INT is asserted an interrupt is generated on S0ASIC independent of the frequency of the CPU (32 kHz or 26 MHz).

1.2.18 EBU: Not Enough GPIO Can Be Used in VDD_EBU Domain (Mainly for WP of Flash)

ID

WS00005905

Type

MCU

Description

In a standard configuration of E-GOLDradio, only the A23 pin, configured as GPIO, can be used as the write protect pin of a Flash.

If A23 is used as address line, one line from the VDD_EBU power supply domain is missing, to use the write protect pin of a Flash.

Pads on VDD_EBU domain are (in a standard configuration) used as their main function and are not available as GPIO.

Solution

1. Another GPIOs can be used, when the voltages of VDD_EBU and VDD_DIG are almost in the same range. Further investigations have to be done during board design.
2. Use a Level Shifter.

1.2.19 I2S: Burst Mode Not Capable Of Quasi-Continuous Transmission

ID

WS00005909

Type

Top Level

Description

The I2S interface is not capable of quasi-continuous transmission in the burst mode. This is of special importance for the 8 kHz IFX Bluetooth Interface.

The FW needs a certain reaction time (>50µs) to reconfigure the RX/TX Start bit. Therefore, a quasi-continuous burst transmission with a 8 kHz (125 µs) spacing of the WA (sync) signal is not possible.

The problem arises also for higher frequencies.

Clarification of the problem:

In PCM mode (also known as "burst mode"), the start bits are cleared with the TX/RX interrupt; they have to be set again by software to continue the transfer.

The TX interrupt occurs at the rising edge of WA for the last transmitted sample. The RX interrupt occurs when the last bit of the last received sample is latched (in the middle of the last bit period).

In order to achieve a continuous transmission this way, the start bits need to be set again in the time between the TX/RX interrupt and the next WA position. This may be possible for the TX side but is very difficult for the RX side. (Example timing for 8 kHz sample rate: ca. 125 µs for TX, ca 3 µs for RX).

Note: The "last sample" in this context is the sample in the position before TXINTADDR/RXINTADDR.

Solution

A FW patch for the slave mode and 8 kSpls/s is available.

Do not use master mode or higher frequencies.

HW change not planned.

Workaround:

1. Higher bit clock frequency (slave mode only)

It is possible to gain time for the RX side with a configuration that has some delay between the last bit period and the next WA signal. This is achieved by using a higher bit clock frequency that results in some spare clock periods between the last bit of sample (n) and the beginning of sample (n+1). Such a configuration cannot be set up when the S-GOLDLite is master, but it can be used when the S-GOLDLite is slave.

The BlueMoon Single Cellular PMB8761 as master uses a bit clock of 500 kHz for a sampling rate of 8 kHz, the resulting reaction time after the RX interrupt is about 90 µs.

Generally it is preferable to let S-GOLDLite be the master on the I2S interfaces. (even small clock difference can cause sample under/overflow; slave mode needs more pins on I2S1)

2. Avoid interrupt situation (master or slave mode)

A continuous transfer can also be achieved if the interrupt situation is avoided: while the interface is running, the INTADDR is modified such that the interface logic never reaches the "last sample".

– Use TX interrupt but avoid RX interrupt
  The interface could be used with TX interrupts only: at startup and whenever a TX interrupt occurs, the INTADDR registers are set up so that the TX interrupt occurs some samples before the RX interrupt. The sample buffer can be maintained based on RWADDR.
– Avoid all interrupts
  With a synchronous DSP loop, e.g. the voiceband scheduler running at 8 kHz, it is also possible to avoid interrupts for both TX and RX by continuously keeping the INTADDR behind the current RWADDR position. Again, the sample buffer can be maintained based on RWADDR.

Note: The workaround technique “Avoid all interrupts*” is used in the current Infineon firmware.*

1.2.20 SIM: Auto Powerdown Not SIM Standard Compliant

ID

WS00005911

Type

MCU

Description

The hardware controlled powerdown function of the SIM interface does not always do a ISO/IEC 7816-3 standard compliant powerdown of the SIM interface.

The hardware controlled powerdown is activated by:

1) Writing a 1 to the SIM_CTRL.SIMPDWN bit

or

2) Writing a 1 to the SIM_CTRL.APDWN bit so that any change on input signal SMC_IN will trigger a hardware controlled powerdown.

The legal powerdown of a SIM has the following sequence:

a) CC_RST is driven from logic 1 to logic 0

b) CC_CLK is deactivated

c) CC_IO is driven to logic 0

d) CC_VZ deactivates the power to the SIM.

The error is that the CC_CLK is sometimes:

i) Deactivated before the CC_RST

or

ii) Deactivated after CC_CLK, but with a reduced duty cycle (shortened high time).

Solution

Workaround:

Do a software controlled powerdown of the SIM. This means that software deactivates each SIM signal individually. (See SIM specification.)

1.2.21 RTC: Alarm Interrupt Generation

ID

WS00005913

Type

DSP

Description

The spec says:

"One RTC_CNT cycle after the value in register RTC_CNT equals the value in RTC_ALARM, an interrupt request is generated."

But the interrupt is not generated one RTC_CNT cycle later (i.e., after the value in RTC_CNT has incremented). It is generated immediately on (or possibly one RTC_T14CNT cycle after) this condition.

As a consequence, the alarm stays active and cannot be cleared until the next RTC_CNT cycle, because the alarm condition is still valid.

Solution

Workaround:

Change RTC_ALARM to another value and clear the Alarm interrupt.

If the old value is to be kept, wait until the next T14 interrupt (RTC_CNT increased) and then set the RTC_ALARM register back to the old value.

1.2.22 CR110651: Delayed ILLOPA Trap Execution If Caused by Double-Indirect MOV Instruction

ID

WS00005918

Type

MCU

Description

The illegal word operand access trap (ILLOPA) should be processed whenever a word operand read or write of an odd byte address is started. The class B trap routine expects the stack to contain the instruction following the instruction that caused the trap.

The immediate execution of the ILLOPA trap following the instruction with an illegal odd byte address cannot be guaranteed in case of the double-indirect MOVes (only word MOVes are of interest):

MOV [Rwn+], [Rwm] (opcode D8)

MOV [Rwn], [Rwm+] (opcode E8)

MOV [Rwn], [Rwm] (opcode C8)

Depending on the fill-state of the CPU internal instruction queue, an additional instruction may be executed before the ILLOPA trap is executed. This can be the linear successor of the MOV or any instruction injected at that time, such as an interrupt TRAP.

Depending on the type of instruction that slips in between, the state of the stack may be different from that expected by the class B trap routine.

In any case, the latest stack entry will not contain the IP of the instruction following the instruction that caused the ILLOPA.

Solution

-

1.2.23 CR110652: Delayed ILLBUS Trap Execution When Writing External Bus

ID

WS00005919

Type

MCU

Description

The illegal external bus access trap (ILLBUS) should be processed whenever an external instruction fetch, data read, or data write is started and no external bus configuration has been specified. The class B trap routine expects the stack to contain the instruction following the one that caused the trap.

The immediate execution of the ILLBUS trap following the instruction performing the external bus access cannot be guaranteed for instructions that write - but don't read - to the external bus. Instructions that also perform a read access to the external bus (double-indirect MOVes) do not show the problem.

Depending on the fill-state of the CPU-internal instruction queue, an additional instruction may be executed before the ILLBUS trap is executed. This can be the linear successor of the instruction accessing the external bus or any instruction injected at that time, for example, an interrupt TRAP.

Depending on the type of instruction that slips in between, the state of the stack will be different from that expected by the class B trap routine. In any case, the latest stack entry will not contain the IP of the instruction following the one that caused the ILLBUS.

Example:

...

MOV [R12], R1 ; this instruction performs the illegal write access

MOVB RH4, T3 ; this instruction slips unbeaten

TRAP #0Ah ; late TRAP, the stack will not contain the IP value of the MOVB as it should

...

Solution

-

1.2.24 CR110653: Visible Mode Limitation for Read Accesses on X-Bus

ID

WS00005920

Type

MCU

Description

Read accesses on the internal X-Bus are not displayed properly on the external bus despite an active visible mode. This happens because the XBC needs an additional cycle to propagate the read data via the write data bus to the external bus.

The data shown in the external bus at the time it would be expected (according to the X-Bus configuration) is an old value and when the right value would be available, it is not really visible on the external pins because the port drivers are already controlled according to the next access.

Solution

Workaround:

Program one or more additional MCTC waitstates than are needed for the data transfer. This gives the XBC the time to propagate the data to the external bus.

At first, the old data is shown on the external bus. Only in the additional cycles are the data correct.

This workaround changes the timing behavior, however this may be acceptable since the Visible Mode is just used for debugging purposes.

1.2.25 CR110654: DIP Shows Incorrect Value If Breakpoint Is Inside Atomic/External Sequence

ID

WS00005921

Type

MCU

Description

The OCDS DIP register should reflect the current program counter value when the CPU goes into the halt mode, for example, due to a hardware breakpoint.

If the hardware breakpoint is inside an atomic sequence, the DIP register will be updated with the IP of the instruction that caused the hardware trigger. However, due to the low priority of the 'CPU halt' respective atomic, all of the protected instructions will be executed before the CPU actually halts.

Example 1:

```
    atomic #4
@ mov   R0, #1111h
    mov   R0, #2222h
    mov   R0, #3333h
    mov   R0, #4444h
```

The breakpoint is set on the IP of the first mov. The CPU halts when all four mov instructions have been executed. The DIP register is updated to the IP of the first mov.

Example 2:

```
    atomic #4
    mov   R0, ONES      ; R_ADR trigger hits -> DIP update
    mov   R0, #3333h
    mov   R0, #2222h
    mov   R0, #1111h
```

The OCDS has been programmed to trigger when address (ONES) is read. Again, the CPU halts when all four mov instructions have been executed. The DIP register is updated to the IP of the first mov.

Solution

-> IP breakpoints: Disable placement of IP breakpoints within atomic/extended sequences.

-> Address/data breakpoints: None

1.2.26 CR110655: Break-after-Make Does Not Work Correctly in Atomic Scenarios

ID

WS00005922

Type

MCU

Description

The C166Sv1 OCDS supports break-before-make (BBM) and break-after-make (BAM) for instruction pointer (IP) breakpoints.

For BBM (bit DTREVT.BAM cleared) code execution will stop before the instruction upon which the breakpoint is set is executed.

For BAM (bit DTREVT.BAM set) code execution will stop after the instruction upon which the breakpoint is set is executed.

There is however 1 scenario where the BAM is not performed as it should be: a BAM breakpoint set before an ATOMIC/extended sequence. In this case, the CPU does not go into the Halt mode until the last protected instruction has been executed.

In this scenario, the user also observes the problem according to CPU_BREAK_DIP ("DIP shows incorrect value if breakpoint is inside atomic/ext. sequence"), for example, the DIP is not updated to indicate the last instruction that was executed, but it shows the IP of the breakpoint instead.

Example:

A BAM breakpoint is set before an ATOMIC/extended sequence. The CPU does not halt until all the NOP's have been executed. DIP is updated to @

```
...
@MULR0, R1; hardware breakpoint has been set to this location (BAM)
EXTR#4
NOP
NOP
NOP
NOP
ADDR0, R1
...
```

Solution

Avoid setting BAM breakpoints on instructions followed by atomic/extended sequences.

1.2.27 CR110656: Read Address Triggers Cannot Be Generated for All Instruction Reads

ID

WS00005923

Type

MCU

Description

The C166Sv1 OCDS can be programmed to generate hardware triggers on the address reads bitfield DTREVT.MUX_R = “11” (R_ADR)).

Any range of addresses can be programmed using registers DCMPG and DCMPL.

Any C166Sv1 instruction performs up to two operand reads from the address space.

Any implicit GPR (or IDX register for MAC instructions) reads during indirect addressing mode are not taken into account.

Several restrictions apply to the generation of read address triggers in the current implementation:

For each instruction, triggers can be generated for only one operand's read address:

1a. There are instructions performing only one operand read for which read address triggers cannot be generated (see below).

1b. For all instructions performing two operand reads, only one of the two operand's address can be triggered on (see below).

2. (MAC users only): Due to special CoREG addressing scheme, no triggers can be generated on MAC instruction CoSTORE.

* Details for 1a.

The following instructions perform only one read in the address space (when certain addressing modes are used). It is not possible to trigger on the related read addresses:

i. ADD(B), ADDC(B), SUB(B), SUBC(B), AND(B), OR(B), XOR(B), CMP(B)

=> applies to addressing modes:

INST Rw/b, #data3

INST reg, #data8/16

ii. CMPD1/2, CMPI1,2

=> applies to addressing modes:

INST Rw, #data4/16

iii. CPL(B), NEG(B)

iv. SCXT reg, #data16

v. MAC instruction CoNOP [IDX*]

Triggers can be generated for all other instructions performing a single operand read.

Note: Though the bit-modifying instructions BSET, BCLR, BFLDH, BFLDL perform implicit register reads, the instructions are considered to be write-only.

Some counter-examples:

- A trigger can be generated on the GPR's address for any "DIVxy Rw" or "EXTxy Rw, #irang"
- MAC: a trigger can be generated for operand address "CoNOP [Rwn*]" (address of unused read)

* Details for 1b.

All instructions performing two operand reads have a syntax corresponding to "INST op1, op2" ;(op1, op2 != immediate values).

Read address triggers are possible only for the address of op2.

Example:

ADD R1, R2

=> It is possible to trigger only on the address of R2.

(MAC users:) This also applies to all MAC instructions "CoXXX op1, op2" ; (op1, op2 != immediate values, CoREG)

Solution

-

1.2.28 CR110657: ILLBUS Exception Inside ILLBUS ISR Problem

ID

WS00005924

Type

MCU

Description

When in the ISR of an ILLBUS trap and another ILLBUS exception is detected due to a Data Read Operation, the core will not resume correctly but waits indefinitely for the second Data Read (unless a Reset is performed).

This situation is very unlikely in a real application. If the External Bus is disabled (the reason for having an ILLBUS), the ILLBUS-ISR will not perform a read operation on the external bus space unless it first enables the bus. Also EPECs/DPECs (the only events that can interrupt a class-B ISR and can lead to the problem) accessing the external bus should not be requested unless the external bus is enabled.

No problem appears if the ISR enables the External Bus before the Data Read Operation that would cause the second ILLBUS trap is executed. Also, no problem appears when the TFR.ILLBUS flag is reset at the beginning of the ISR (and before the Data Read Operation that would cause the second ILLBUS trap is executed).

Solution

-

1.2.29 CR110764: External Break Events (BREAK Pins) Can Be Lost

ID

WS00005925

Type

MCU

Description

The C166SV1 is a 2-cycle-machine meaning that a machine cycle has at least two clock cycles.

However, under certain circumstances, for example, insertion of waitstates while fetching instructions or accessing data, a machine cycle takes longer than two clock cycles.

As described in the Integration Manual, the brk_async_n_i input must be asserted for at least 2 clock cycles. However, internally this input is sampled/synchronized with machine cycles. So, if there are waitstates, such a short pulse might not be detected.

There is also a similar problem with the brk_sync_n_i input. Assuming that it is synchronous to c166_clk_pos (as defined in the Integration Manual), it is combinational used in the CPU. However, some of these generated signals are sampled with machine cycle dependant signals and so a short pulse on brk_sync_n_i might also be lost.

Solution

Workaround:

Generate pulses (for brk_async_n_i and brk_sync_n_i inputs) of at least one machine cycle.

To assure a reliable behavior of the Break feature, this workaround can be implemented in the hardware, in the logic responsible for the generation/synchronization of these signals (outside of the C166S core). This logic can make use of the Core output trc_sel_o to detect a "machine cycle". One cycle of this signal is identical with one machine cycle. To use the trc_sel_o signal the trace bus has to be enabled (trc_enable_i '1').

It is suggested to connect trc_enable_i to the same signal as ocds_enable_i, or if the OCDS can be also enabled by software, to the core output ocds_enabled_o.

If the brk_async_n_i is used, the brkout_n_o output (Activate External Pin action) can be used to deactivate the input signal. The brk_async_n_i has already been synchronized when this output is set.

This is not possible for the brk_sync_n_i as the brkout_n_o is combinational generated in this case.

1.2.30 CR111192: Bit-Manipulating Instructions on EXICON/EXISEL Write Wrong Values

ID

WS00005926

Type

MCU

Description

Bit-manipulating instructions do not work properly on the EXISEL and EXICON registers.

Instead of updating just the affected byte, the other bytes are overwritten with wrong data.

Example:

MOV R0, #0000h

EXTR #1

MOV EXICON, #2222h

NOP

EXTR #1

BMOV EXICON.1, R0.0 ; expected: EXICON = #2220h, actual (in error): EXICON = #0220h

Affected registers:

EXISEL, EXICON; no other registers are affected by this bug

Affected instructions:

BAND, BCLR, BFLDL, BFLDH, BMOV, BMOVN, BOR, BSET, BXOR, JBC, JNBS

Solution

Don’t use bit-manipulating instructions that write on these two registers. Instead use word operations (e.g., MOV, AND, OR) and treat them like any other non-bit addressable register.

1.2.31 CR112452: Wrong SP Used with SCXT Reg, Mem, and Mem Pointing to SP

ID

WS00005927

Type

MCU

Description

The behavior of the SCXT instruction is as follows:

SCXT reg, mem

(tmp1) = (reg)

(tmp2) = (mem)

(SP) = (SP) - 2

((SP)) = (tmp1)

(reg) = (tmp2)

With this description, in a sequence like

```
MOV SP, #0FC00h
SCXT R1, 0FC12h  ; SP address is 0FE12h
```

R1 should contain the old SP value 0FC00h. However, the new SP value is written to R1 (0FBFE).

Solution

Do not directly access SP in the SCXT instruction, but use an intermediate register.

Example:

```
MOV R15, SP
SCXT R1, R15
```

1.2.32 RTC: Internal Oscillator Instability

ID

WS00006183

Type

Backend

Description

The internal oscillator shows a instable jitter in the running CPU:

- D000 is the Jitter measured with the phone on; Sigma = 203.6 Hz peaked noise
- D001 is the Jitter measured with the phone off; Sigma = 78.8 Hz Gaussian noise.

Solution

SW workaround:

Perform 32 kHz RTC calibration with the CPU:

- Waiting in a code loop in internal RAM (with no access to external bus)
- Running at 26 MHz pre-scaled by 2^7.

The calibration is run only if more than 16 frames of possible sleep were foreseen and a calibration time out is required. Also, if possible, an equalizer timing offset based calibration algorithm is preferably used when mobile is camped good.

The root cause is due to:

- The very high impedance of 32 kHz oscillator, needed by low power requirement
- Capacitance crosstalk with the address/data/control EBU lines.

1.2.33 TEAKlite: Loop (rep) Ends Unexpectedly After Interrupt in PROM Page 1

ID

WS00006189

Type

DSP

Description

After returning from IR service routine, the first interrupted REP on the extended PROM page (page 1 for E-GOLDradio) does not continue correctly. The error only occurs one time after reset. All the following REPs cannot be interrupted

Solution

During the boot procedure, interrupt the first REP. Because the second REP is not interruptible, the system becomes deterministic.

Solved in G14_ER_STARTUP_V11.

1.2.34 DSP: Consecutive Write Operation to Some Registers Fails

ID

WS00006272

Type

Top Level

Description

When the TEAKlite clock is slower than clk_ms/2 (52 MHz) than consecutive write operations to registers clocked by clk_ms (104 MHz) can fail.

Solution

Workaround:

Insert NOPs between consecutive writes to registers within a clock domain.

1.2.35 Boot Code: Access in External Flash during the BOOTROM

ID

WS00007413

Type

MCU

Description

The internal boot is proceed from the internal ROM with a 26 MHz frequency, a latched chip select and normal ALE.

This configuration leads to timing violation with some Flashes, when a check is done to know if the external flash contains valid code to be executed or not.

The timing which is violated is coming from the Flash, which needs "ADV/ low when chip select is active (min 10ns)" (according to the Flash specification).
When the boot is directly from the external flash (no bootrom execution), the ALE is automatically configured as extended and there is no timing violation.

Solution

No solution.

1.2.36 GSM Timer Unit: TFSKIP.SKIPC Bit Has No Effect

ID

WS00007537

Type

MCU

Description

The TFSKIP.SKIPC bit has no effect of the reset behavior to skip the reset of the current CTDMA counter.

Specified behavior:

If TFSKIP.SKIPC =1 the next time a reset occurs the CTDMA counter value is not set to 0 and continues to be increased over its normal max value.

The TFSKIP.SKIPC is automatically reset internally after skipping the reset of CTDMA counter.

Actual behavior:

The TFSKIP.SKIPC bit has no effect and the reset of the CTDMA counter is processed.

The TFSKIP.SKIPC is automatically reset internally after the reset of CTDMA counter.

Solution

Do not use TFSCKIP.SKIPC bit.

Skip functionality have to be handled by SW

1.2.37 SCCU X_BUS Clock at 78 MHz

ID

WS00007621

Type

MCU

Description

The SCCU must run with 13 MHz to ensure GSM timer synchronization.

At 78 MHz this cannot be achieved due to RST_CTRL_STA.SCCUCL bits only allows divisions of the master clock by factor 1; 2; 4, or 8.

The division by 6, which is needed for 78 MHz Mode to setup 13 MHz, is not supported.

This means, the Stand-By Mode cannot be entered in the 78 MHz mode.

Solution

Use 26 MHz or 5 MHz before entering the Stand-By Mode.

1.2.38 RTC: Internal Oscillator Instability

ID

WS00006183

Type

Backend

Description

The internal Oscillator shows a instable Jitter behavior, by running CPU.

Solution

-

1.2.39 GSM Timer Unit: TFSKIP.SKIPC Bit Has No Effect

ID

WS00007537

Type

DSP

Description

The TFSKIP.SKIPC bit has no effect of the reset behavior to skip the reset of the current CTDMA counter.

Specified behavior:

If TFSKIP.SKIPC = 1, the next time a reset occurs the CTDMA counter value is not set to 0 and continues to be increased over its normal max value.

The TFSKIP.SKIPC is automatically reset internally after skipping the reset of CTDMA counter.

Actual behavior:

The TFSKIP.SKIPC bit has no effect and the reset of the CTDMA counter is processed.

The TFSKIP.SKIPC is automatically reset internally after the reset of CTDMA counter.

Solution

Do not use TFSCKIP.SKIPC bit, the Skip functionality has to be handled by SW.

1.2.40 SSC: Phase Error When High Baudrate Is Set

ID

WS00008268

Type

Evaluation Board

Description

The SSCs (PD Bus SSC and DSP SSC) support a phase error detection, as defined in the spec:

A phase error is detected if the phase error detection is enabled and the receive data change is within 1 module clock cycle before and 2 module clock cycles after the receive edge of the shift clock SSCx_SCLK.

If the shift clock SSCx_SCLK is generated by a fractional divider, a phase error may occur due to the fractional divider jitter.

The fractional divider generates clock low/high phases of different lengths. In case of a short phase, the difference between the receive edge and the transmit edge of the shift clock may be too small as compared to the module clock period and thus violates the condition given above.

This may happen, for example, if an SSC is master and receives data. It always happens in this case if the shift clock frequency is between 1/4 and 1/2 of the module clock frequency.

This may also happen if the shift clock frequency is between 1/8 and 1/4 of the module clock frequency.

In the latter case, the critical clock and date edges sometimes occur logically at the same time and it depends on pad and board timings whether a phase error is detected or not.

No problem occurs if the shift clock frequency is exactly 1/2, 1/4 or 1/8 or below 1/8 of the module clock frequency.

Also no problem occurs if the master SSC does not generate a fractional divider jitter or if the receiving SSC does not support/not enable phase error detection.

If a receiving SSC outside of the E-GOLDradio supports phase error detection and it is enabled, the detection may follow other rules as compared to the E-GOLDradio SSCs. In this case, other frequencies may be critical as in the test case given above.

Solution

-

1.2.41 EBU: Improve Timings for EBU Interface

ID

WS00007486

Type

Physical Implementation

Description

EBU Timings need to be improved to fulfil specified target values.

Solution

To be done in a future version of E-GOLDradio.

Total Quality Management

Qualität hat für uns eine umfassende Bedeutung. Wir wollen allen Ihren Ansprüchen in der bestmöglichen Weise gerecht werden. Es geht uns also nicht nur um die Produktqualität – unsere Anstrengungen gelten gleichermaßen der Lieferqualität und Logistik, dem Service und Support sowie allen sonstigen Beratungs- und Betreuungsleistungen.

Dazu gehört eine bestimmte Geisteshaltung unserer Mitarbeiter. Total Quality im Denken und Handeln gegenüber Kollegen, Lieferanten und Ihnen, unserem Kunden. Unsere Leitlinie ist jede Aufgabe mit „Null Fehlern“ zu lösen – in offener Sichtweise auch über den eigenen Arbeitsplatz hinaus – und uns ständig zu verbessern.

Unternehmensweit orientieren wir uns dabei auch an „top“ (Time Optimized Processes), um Ihnen durch größere Schnelligkeit den entscheidenden Wettbewerbsvorsprung zu verschaffen.

Geben Sie uns die Chance, hohe Leistung durch umfassende Qualität zu beweisen.

Wir werden Sie überzeugen.

Quality takes on an allencompassing significance at Semiconductor Group. For us it means living up to each and every one of your demands in the best possible way. So we are not only concerned with product quality. We direct our efforts equally at quality of supply and logistics, service and support, as well as all the other ways in which we advise and attend to you.

Part of this is the very special attitude of our staff. Total Quality in thought and deed, towards co-workers, suppliers and you, our customer. Our guideline is “do everything with zero defects”, in an open manner that is demonstrated beyond your immediate workplace, and to constantly improve.

Throughout the corporation we also think in terms of Time Optimized Processes (top), greater speed on our part to give you that decisive competitive edge.

Give us the chance to prove the best of performance through the best of quality – you will be convinced.

w w w . i n f i n e o n . c o m

Published by Infineon Technologies AG