Silicon Errata for the CYT6BJ Series Rev. A

Document Revision: 1.2
Date: 2023-10-17

This document describes the errata for CYT6BJ Rev. A Series of devices. Details include trigger conditions, scope of impact, available workarounds, and applicable silicon revisions. Contact your local Infineon Sales representative for further questions.

Part Numbers Affected

Part Number
All CYT6BJ Rev. A parts

CYT6BJ Rev. A Qualification Status

Product Status: Engineering Samples

CYT6BJ Rev. A Errata Summary

This table defines the errata applicable to CYT6BJ Rev. A Series of devices.

Items	Part Number	Silicon Revision	Fix Status
Errata ID 96 CAN FD RX FIFO top pointer feature does not function as expected	CYT6BJ8DDAES CYT6BJBDHAES CYT6BJCDHAES	Rev. A	No silicon fix planned. Use workaround.
Errata ID 97 CAN FD debug message handling state machine not get reset to Idle state when CANFD_CH_CCCR.INIT is set			No silicon fix planned. Use workaround.
Errata ID 206 Hardfault may occur when calling some SROM APIs while executing EraseSector or ProgramRow in non-blocking mode			No silicon fix planned. TRM (002-24401 Rev. H) will be updated.
Errata ID 209 CAN FD sporadic data corruption (payload) in case acceptance filtering is not finished before reception of data R3 (DB7..DB4) is completed			No silicon fix planned. Use workaround.
Errata ID 212 Description for PASS SARx to TCPWMx direct connect triggers one-to-one is incorrect in datasheet			No silicon fix planned. Datasheet (002-33466 Rev. E) will be updated.

96. CAN FD RX FIFO top pointer feature does not function as expected

- **Problem Definition**
  RX FIFO top pointer function calculates the address for received messages in Message RAM by hardware. This address should be re-start back from the start address after reading all messages of RX FIFO n size (n: 0 or 1). However, the address does not re-start back from the start address when RX FIFO n size is set to 1

(CANFD_CH_RXFnC.FnS = 0x01). This results in CPU/DMA to read messages from the wrong address in Message RAM.

- **Parameters Affected**
  N/A
- **Trigger Condition(s)**
  RX FIFO top pointer function is used when RX FIFO n size set to 1 element (CANFD_CH_RXFnC.FnS = 0x01).
- **Scope of Impact**
  Received message cannot be correctly read by using RX FIFO top pointer function, when RX FIFO n size set to 1 element.
- **Workaround**
  Any of the following.
  1) Set RX FIFO n size to 2 or more when using RX FIFO top pointer function.
  2) Do not use RX FIFO top pointer function when RX FIFO n size set to 1 element. Instead of RX FIFO top pointer, read received messages from the Message RAM directly.
- **Fix Status**
  No silicon fix planned. Use workaround.

97. CAN FD debug message handling state machine not get reset to Idle state when CANFD_CH_CCCR.INIT is set

- **Problem Definition**
  If either CANFD_CH_CCCR.INIT bit is set by the Host or when the M_TTCAN module enters BusOff state, the debug message handling state machine stays in its current state instead of being reset to Idle state. Configuring the bit CANFD_CH_CCCR.CCE does not change CANFD_CH_RXF1S.DMS.
- **Parameters Affected**
  N/A
- **Trigger Condition(s)**
  Either CANFD_CH_CCCR.INIT bit is set by the Host or when the M_TTCAN module enters BusOff state.
- **Scope of Impact**
  The errata is limited to the use case when the Debug on CAN functionality is active. Normal operation of CAN module is not affected, in which case the debug message handling state machine always remains in Idle state. In the described use case, the debug message handling state machine is stopped and remains in the current state signaled by the bit CANFD_CH_RXF1S.DMS. In case CANFD_CH_RXF1S.DMS is set to 0b11, DMA request remains active.
  Bosch classifies this as non-critical error with low severity, there is no fix for the IP, Bosch recommends the workaround listed also here.
- **Workaround**
  In case the debug message handling state machine has stopped while CANFD_CH_RXF1S.DMS is 0b01 or 0b10, it can be reset to Idle state by hardware reset or by reception of debug messages after CANFD_CH_CCCR.INIT is reset to zero.
- **Fix Status**
  No silicon fix planned. Use workaround.

206. Hardfault may occur when calling some SROM APIs while executing EraseSector or ProgramRow in non-blocking mode

- **Problem Definition**
  The following SROM APIs read data from bank#0 (or bank#1 if dual bank mode with mapping B is used) in SFlash. While doing that the check for active non-blocking erase or program of bank#0 (or bank#1 if dual bank mode with mapping B is used) is not performed. Therefore, reading bank#0 (or bank#1 if dual bank mode with mapping B is used) while there is an active erase/program operation will trigger a bus error which can result in a hardfault occurrence based on FLASHC_FLASH_CTL register settings.
  Affected SROM APIs:

- ReadSWPU
- WriteSWPU
- GenerateHash
- Checksum*
- ComputeBasicHash*
- CheckFactoryHash
- ProgramWorkFlash**
- SwitchOverRegulators
- LoadRegulatorsTrims

*: Note that it should not be called if you programming/erasing bank that you are going to calculate.
**: Note that it is not possible to use it during non-blocking operation.

- **Parameters Affected**
  N/A
- **Trigger Condition(s)**
  Calling the affected SROM APIs while executing EraseSector or ProgramRow in non-blocking mode on bank#0 (or bank#1 if dual bank mode with mapping B is used).
- **Scope of Impact**
  The affected SROM APIs cannot be used while executing EraseSector or ProgramRow in non-blocking mode on bank#0 (or bank#1 if dual bank mode with mapping B is used).
- **Workaround**
  Do not use the affected SROM APIs while executing EraseSector or ProgramRow in non-blocking mode on bank#0 (or bank#1 if dual bank mode with mapping B is used).
- **Fix Status**
  No silicon fix planned. Architecture TRM (002-24401 Rev. H) will be updated.
- **Impact on Infineon Software**
  Impact: Limitation
  Related modules: HSM-Perf-Lib
  Comment: While executing EraseSector or ProgramRow in non-blocking mode on bank#0 (or bank#1 if dual bank mode with mapping B is used), users must not do anything of following:
  a) call CySldProt_GetSwpuFlashStructCfg
  b) call CySldProt_VerifySecureDomainFlashWriteProtection if CySldProt_SwpuFlashStructGroupConfigurations is non-empty.

209. CAN FD sporadic data corruption (payload) in case acceptance filtering is not finished before reception of data R3 (DB7..DB4) is completed

- **Problem Definition**
  During frame reception the Rx Handler accesses the external Message RAM for acceptance filtering (read accesses) and for storing of the accepted messages (write accesses).

  The time needed for acceptance filtering and for storing of a received message depends on
  · the Host clock frequency
  · the worst-case latency of the read and write accesses to the external Message RAM
  · the number of configured filter elements
  · the workload of the transmit message (Tx) handler in parallel to the receive message (Rx) handler

  Received data bytes (DB0..DBm) from the CAN Core are buffered in the cache of the Rx Handler before they are written to the Message RAM (in words of 4 byte). Data words inside the Message RAM are numbered from R2 to Rn (n ≤ 17).

	31			24	23				16	15		8	7		0
R0	ESI	XTD	RTR	ID[28:0]											
R1A	ANMF	FIDX[6:0]			res	FDF	BRS	DLC[3:0]		RXTS[15:0]					
R1B	ANMF	FIDX[6:0]			res	FDF	BRS	DLC[3:0]		res			TSC	RXTSP [3:0]	
R2	DB3[7:0]				DB2[7:0]					DB1[7:0]			DB0[7:0]		
R3	DB7[7:0]				DB6[7:0]					DB5[7:0]			DB4[7:0]		
...	...				...					...			...		
Rn	DBm[7:0]				DBm-1[7:0]					DBm-2[7:0]			DBm-3[7:0]		

Figure 1. Rx Buffer and FIFO Element

Under the following conditions a received message will have corrupted data while the received message is signaled as valid to the host.
1) The data length code (DLC) of the received Message is greater than 4 (DLC > 4)
2) The storage of Ri of a received message into the Message RAM (after acceptance filtering is done) has not completed before R(i+1) is transferred from the CAN Core into the cache of the Rx Handler (where $2 \le i \le 5$).
3) While condition 1) and 2) apply, a concurrent read of data word Ri from the cache and write of data word R(i+1) into the cache of the Rx handler happens.

The data will be corrupted in a way, that in the Message RAM R(i+1) has the same content as Ri.

Despite the corrupted data, the M_TTCAN signals the storage of a valid frame in the Message RAM:
- Rx FIFO: FIFO put index RXFnS.FnPI is updated.
- Dedicated Rx Buffer: New Data flag NDATn.NDxx is set.
- Interrupt flag IR.MRAF is not set.

The issue may occur in FD Frame Format as well as in Classic Frame Format.

Figure 2 shows how the available time for acceptance filtering and storage is reduced.

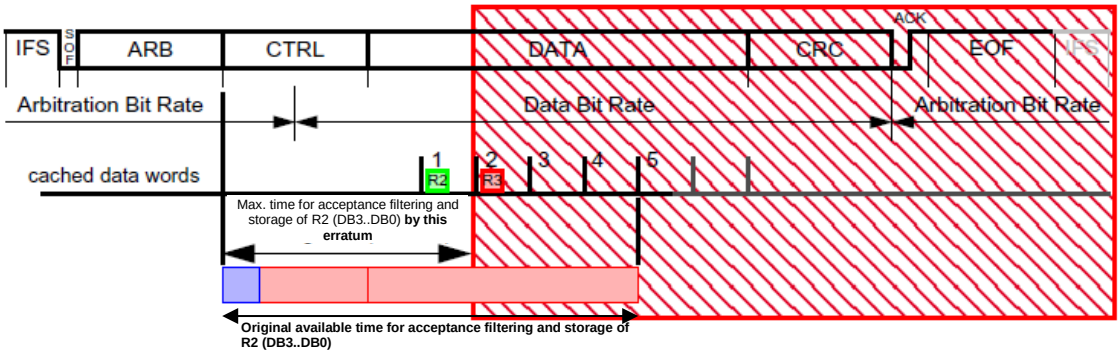

Figure 2. CAN Frame with DLC>4

Table 1. TRAVEO T2G: Minimum host clock frequency for CAN FD when DLC = 5

Number of configured active filter element 11-bit IDs / 29-bit IDs [1,2]	Number of active CAN channels in an instance	Arbitration bit rate = 0.5 Mbps				Arbitration bit rate = 1 Mbps			
		Data bit rate = 0.5 Mbps	Data bit rate = 1 Mbps	Data bit rate = 2 Mbps	Data bit rate = 4 Mbps	Data bit rate = 1 Mbps	Data bit rate = 2 Mbps	Data bit rate = 4 Mbps	Data bit rate = 5 Mbps
32 / 16	2	3.9 MHz	7.1 MHz	13.1 MHz	22.8 MHz	7.7 MHz	14.1 MHz	26.1 MHz	31.5 MHz
	3	5.4 MHz	9.9 MHz	18.3 MHz	31.8 MHz	10.7 MHz	19.7 MHz	36.5 MHz	44.0 MHz
	4	6.9 MHz	12.7 MHz	23.5 MHz	40.8 MHz	13.8 MHz	25.3 MHz	46.9 MHz	56.5 MHz
	5	8.4 MHz	15.5 MHz	28.6 MHz	49.9 MHz	16.8 MHz	30.9 MHz	57.2 MHz	69.0 MHz
64 / 32	2	7.4 MHz	13.5 MHz	24.9 MHz	43.4 MHz	14.7 MHz	26.9 MHz	49.8 MHz	60.0 MHz
	3	10.3 MHz	18.8 MHz	34.9 MHz	60.7 MHz	20.5 MHz	37.6 MHz	69.7 MHz	84.0 MHz
	4	13.2 MHz	24.2 MHz	44.8 MHz	78.0 MHz	26.3 MHz	48.4 MHz	89.5 MHz	107.9 MHz[3]
	5	16.1 MHz	29.6 MHz	54.7 MHz	95.3 MHz	32.1 MHz	59.1 MHz	109.4 MHz[3]	131.8 MHz[3]
96 / 48	2	10. 8 MHz	19.9 MHz	36.8 MHz	64.0 MHz	21.6 MHz	39.7 MHz	73.5 MHz	88.6 MHz
	3	15.1 MHz	27.8 MHz	51.5 MHz	89.6 MHz	30.2 MHz	55.6 MHz	102.9 MHz[3]	124.0 MHz[3]
	4	19.4 MHz	35.7 MHz	66.1 MHz	115.1 MHz[3]	38.8 MHz	71.4 MHz	132.2 MHz[3]	159.3 MHz[3]
	5	23.7 MHz	43.6 MHz	80.8 MHz	140.7 MHz[3]	47.4 MHz	87.2 MHz	161.5 MHz[3]	194.7 MHz[3]
128 / 64	2	14.3 MHz	26.3 MHz	48.6 MHz	84.7 MHz	28.4 MHz	52.5 MHz	97.2 MHz	117.2 MHz[3]
	3	20.0 MHz	36.8 MHz	68.0 MHz	118.5 MHz[3]	40.0 MHz	73.5 MHz	136.0 MHz[3]	164.0 MHz[3]
	4	25.7 MHz	47.2 MHz	87.5 MHz	152.3 MHz[3]	51.4 MHz	94.4 MHz	174.9 MHz[3]	210.8 MHz[3]
	5	31.4 MHz	57.7 MHz	106.9 MHz[3]	186.1 MHz[3]	62.7 MHz	115.4 MHz[3]	213.7 MHz[3]	257.5 MHz[3]

1. M_TTCAN starts always at filter element #0 and proceeds through the filter list to find a matching element. Acceptance filtering stops at the first matching element and the following filter elements are not evaluated for this message. Therefore, the sequence of configured filter elements has a significant impact on the performance of the filtering process.
2. Acceptance filtering search for 11-bit IDs and 29-bit IDs filter element is running separately, only one configured filter setting should be considered. Searching for one 29-bit filter element requires approximately double cycles for one 11-bit filter element.
3. Frequency is not reachable since the maximum host clock frequency for M_TTCAN in TRAVEO™ T2G is 100 MHz.

- **Parameters Affected**
  N/A
- **Trigger Condition(s)**
  Under the following conditions a received message will have corrupted data while the received message is signaled as valid to the host.
  1) The data length code (DLC) of the received Message is greater than 4 (DLC > 4)
  2) The storage of Ri of a received message into the Message RAM (after acceptance filtering is done) has not completed before R(i+1) is transferred from the CAN Core into the cache of the Rx Handler (where 2 ≤ i ≤ 5).
  3) While condition 1) and 2) apply, a concurrent read of data word Ri from the cache and write of data word R(i+1) into the cache of the Rx handler happens.
- **Scope of Impact**
  The erratum is limited to the case when the Host clock frequency used in the actual device is below the limit shown in Table 1.
  Corrupted data is written to the Rx FIFO element respective the dedicated Rx Buffer.
  The received frame is nevertheless signaled as valid.
- **Workaround**
  Check whether the minimum Host clock frequency, that is shown in Table 1, is below the Host clock frequency used in the actual device.
  If yes, there is no problem with the selected configuration.
  If no, use one of the following two workarounds.

  First workaround
  Try different configuration by changing the following parameters until the actual host clock frequency (CLK_GR5) is above the minimum host frequency shown in Table 1.
  · Increase the CLK_GR5 frequency in the actual device
  · Reduce the CAN-FD Data Bit rate
  · Reduce the number of configured filter elements
  · Reduce the number of active CAN channels in an instance
  Also, use DLC>=8 instead of DLCs 5, 6, and 7 in the CAN Environment/System, as they place higher demands on the minimum Host clock frequency (the worst case is DLC=5) or restrict your CAN Environment/System to DLC 4.

  Note: While changing the actual host clock frequency, CLK_GR5 must always be equal or higher than PCLK_CANFD[x]_CLOCK_CAN[y] for all configurations.

  Second workaround
  Due to condition 3) the issue occurs only sporadically. Use an end-to-end (E2E) protection (for example, checksum or CRC covering the data field) and add it to all messages in the CAN system, to detect data corruption in received frames.

- **Fix Status**
  No silicon fix planned. Use workaround.
- **Impact on Infineon Software**
  Impact: Limitation
  Related modules: CAN, MCU
  Comment: The user must evaluate the impact of the erratum for each CAN instance separately. A CAN instance is the entirety of CanControllers with the same CanControllerInstance value.
  1) For the number of active CAN nodes: Use the maximum number of CanController configurations of a CAN instance that can be active (Autosar controller state STARTED or SLEEP) at a time.
  2) For the host clock frequency: In McuPeriGroupSettings locate the setting with McuPeriGroup=MCU_PERI_GROUP5_MMIO5 and take the value from McuPeriGroupClockFrequency.
  4) For the number of configured active filter element 11-bit IDs / 29-bit IDs": Use the corresponding values

from the “Message RAM (…) linking table” in the generated Can_PBcfg.h file. Note that each CanController has its separate table. Take the maximum values.
5) For the Arbitration bit rate: Use the maximum CanControllerBaudRate value of all the CanControllers.
6) For the Data bit rate: Use the maximum CanControllerFdBaudRate value of all the CanControllers if configured. Otherwise use CanControllerBaudRate.

212. Description for PASS SARx to TCPWMx direct connect triggers one-to-one is incorrect in datasheet

- **Problem Definition**
  The existing datasheet shows ‘trig=2’ in the description for PASS SARx to TCPWMx direct connect triggers one-to-one, which is incorrect as TCPWM’s input trigger selection (TR_IN_SEL) value. The correct value is ‘4’ as shown in the architecture TRM chapter 25 descriptions and table 25-2.
- **Parameters Affected**
  N/A
- **Trigger Condition(s)**
  Using the triggers one-to-one for PASS SARx to TCPWMx direct connect
- **Scope of Impact**
  The triggers one-to-one for PASS SARx to TCPWMx direct connect cannot work if TCPWM’s input trigger selection is not correct.
- **Workaround**
  Use ‘4’ as TCPWM’s input trigger selection (TR_IN_SEL) value for PASS SARx to TCPWMx direct connect
- **Fix Status**
  No silicon fix planned. Datasheet (002-33466 Rev. E) will be updated.
- **Impact on Infineon Software**
  Impact: No
  Related modules: PWM
  Comment: MCAL PWM module does not support one-to-one triggers.

Revision history

Document revision	Date of release	Description of Change
1.0	2023-03-29	Initial release
1.1	2023-06-15	Added errata ID 209
1.2	2023-10-17	Added errata ID 206, 212

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2023-10-17

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Go to www.infineon.com/support

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

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

WARNINGS

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

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.