- 100-MHz (max) 32-bit Arm® Cortex® M0+ CPU with
  - Single-cycle multiply
  - Memory protection unit
- Inter-processor communication in hardware
- Three DMA controllers
  - Peripheral DMA controller #0 (P-DMA0) with 92 channels
  - Peripheral DMA controller #1 (P-DMA1) with 44 channels
  - Memory DMA controller #0 (M-DMA0) with 4 channels

- 4160 KB of code-flash with an additional 128 KB of work-flash
  - Read-While-Write (RWW) allows updating the code-flash/work-flash while executing code from it
  - Single- and dual-bank modes (specifically for Firmware update Over The Air [FOTA])
  - Flash programming through SWD/JTAG interface
- 512 KB of SRAM with selectable retention granularity

- [1]
  - Supports enhanced Secure Hardware Extension (eSHE) and Hardware Security Module (HSM)
  - Secure boot and authentication
    - Using digital signature verification
    - Using fast secure boot
  - AES: 128-bit blocks, 128-/192-/256-bit keys
  - 3DES[2]: 64-bit blocks, 64-bit key
  - Vector unit[2] supporting asymmetric key cryptography such as Rivest-Shamir-Adleman (RSA) and Elliptic Curve (ECC)

-
  - Memory protection unit (MPU)
  - Shared memory protection unit (SMPU)
  - Peripheral protection unit (PPU)
  - Watchdog timer (WDT)
  - Multi-counter watchdog timer (MCWDT)
  - Low-voltage detector (LVD)
  - Brown-out detector (BOD)
  - Overvoltage detection (OVD)
  - Clock supervisor (CSV)
  - Hardware error correction (SECDED ECC) on all safety-critical memories (SRAM, flash)
-
  - Low-power Active, Sleep, Low-power Sleep, DeepSleep, and Hibernate modes for fine-grained power management
  - Configurable options for robust BOD
    - Two threshold levels (2.7 V and 3.0 V) for BOD on V_{DDD} and V_{DDA}
    - One threshold level (1.1 V) for BOD on V_{CCD}
-
  - Up to two pins to wakeup from Hibernate mode
  - Up to 152 GPIO pins to wakeup from Sleep modes
  - Event Generator, SCB, Watchdog Timer, RTC alarms to wake from DeepSleep modes
-
  - Internal main oscillator (IMO)
  - Internal low-speed oscillator (ILO)
  - External crystal oscillator (ECO)
  - Watch crystal oscillator (WCO)
  - Phase-locked loop (PLL)
  - Frequency-locked loop (FLL)
-
  - Up to eight CAN FD channels
    - Increased data rate (up to 8 Mbps) compared to classic CAN, limited by physical layer topology and transceivers
    - Compliant to ISO 11898-1:2015
    - Supports all the requirements of Bosch CAN FD Specification V1.0 for non-ISO CAN FD
    - ISO 16845:2015 certificate available
  - Up to eight runtime-reconfigurable SCB (serial communication block) channels, each configurable as I^2C, SPI, or UART
  - Up to 12 independent LIN channels
    - LIN protocol compliant with ISO 17987
  - Up to four CXPI channels with data rate up to 20 kbps
-
  - Up to 75 16-bit and eight 32-bit timer/counter pulse-width modulator (TCPWM) blocks
    - Up to 12 16-bit counters for motor control
    - Up to 63 16-bit counters and eight 32-bit counters for regular operations
    - Supports timer, capture, quadrature decoding, pulse-width modulation (PWM), PWM with dead time (PWM_DT), pseudo-random PWM (PWM_PR), and shift-register (SR) modes
  - Up to 11 Event Generation (EVTGEN) timers supporting cyclic wakeup from DeepSleep
    - Events trigger a specific device operation (such as execution of an interrupt handler, a SAR ADC conversion, and so on)

Table of contents

29.1 Part number nomenclature....146

31.1 Bootloading or end-of-line programming....156
31.2 External IP revisions....157

33.1 Revision history change log....170

Core	32-bit Arm® Cortex®-M4F CPU and 32-bit Arm® Cortex® M0+ CPU
Functional safety	ASIL-B
Operating voltage	2.7 V to 5.5 V

Motor control input	3 ch (synchronous sampling of one channel on each of the 3 ADCs)				
Flash security (program/work read protection)	Supported				
Flash Chip erase enable	Configurable				
eSHE/HSM	By separate firmware[3]				
DMA controller	P-DMA0 with 92 channels (16 general purpose), P-DMA1 with 44 channels (8 general purpose), and M-DMA0 with 4 channels				
Internal main oscillator	8 MHz				
Internal low-speed oscillator	32.768 kHz (nominal)				
PLL	Input frequency: 3.988 to 33.34 MHz, PLL output frequency: up to 160 MHz				
FLL	Input frequency: 0.25 to 80 MHz, FLL output frequency: up to 100 MHz				
Watchdog timer and multi-counter Watchdog timer	Supported (WDT + 2× MCWDT) MCWDT#0 tied to CM0+, MCWDT#1 to CM4				
Clock supervisor	Supported				
Cyclic wakeup	Supported				
GPIO_STD	45	59	74	118	148
GPIO_ENH			4		
Smart I/O (Blocks)	3 blocks, 9 I/Os	3 blocks, 14 I/Os	4 blocks, 20 I/Os	5 blocks, 29 I/Os	5 blocks, 36 I/Os
Low-voltage detect	Two, 26 selectable levels				
Maximum ambient temperature					

The shows the CYT2BL architecture, giving a simplified view of the interconnection between subsystems and blocks. CYT2BL has four major subsystems: CPU, system resources, peripherals, and I/O[4, 5]. The color-coding shows the lowest power mode where the particular block is still functional.

CYT2BL provides extensive support for programming, testing, debugging, and tracing of both hardware and firmware.

Debug-on-chip functionality enables in-system debugging using the production device. It does not require special interfaces, debugging pods, simulators, or emulators.

The JTAG interface is fully compatible with industry-standard third-party probes such as I-jet, J-Link, and GHS.

The debug circuits are enabled by default.

CYT2BL provides a high level of security with robust fl

The power system ensures that the supply voltage levels meet the requirements of each power mode, and provides a full-system reset when these levels are not valid. Internal power-on reset (POR) guarantees full-chip reset during the initial power ramp.

Three Brown-out detection (BOD) circuits monitor the external supply voltages (V_{DDD}, V_{DDA}, V_{CCD}). The BOD on V_{DDD} and V_{CCD} are initially enabled and cannot be disabled. The BOD on V_{DDA} is initially disabled and can be enabled by the user. For the external supplies V_{DDD} and V_{DDA}, BOD circuits are software configurable with two settings; a 2.7-V minimum voltage that is robust for all internal signaling, and a 3.0-V minimum voltage, which is also robust for all I/O specifications (which are guaranteed at 2.7 V). The BOD on V_{CCD} is provided as a safety measure and is not a robust detector.

Three overvoltage detection (OVD) circuits are provided for monitoring external supplies (V_{DDD}, V_{DDA}, V_{CCD}), and overcurrent detection (OCD) circuits for monitoring internal and external regulators. OVD thresholds on V_{DDD} and V_{DDA} are configurable with two settings; a 5.0-V and 5.5-V maximum voltage.

Two voltage detection circuits are provided to monitor the external supply voltage (V_{DDD}) for falling and rising levels, each configurable for one of the 26 selectable levels.

All BOD, OVD, and OCD circuits on V_{DDD} and V_{CCD} generate a reset, because these protect the CPUs and fault logic. The BOD and OVD circuits on V_{DDA} can be configured to generate either a reset, or a fault.

CYT2BL contains two regulators that provide power to the low-voltage core transistors: DeepSleep and core internal. These regulators accept a 2.7–5.5-V V_{DDD} supply and provide a low-noise 1.1-V supply to various parts of the device. These regulators are automatically enabled and disabled by hardware and firmware when switching between power modes. The core internal regulators operate in active mode, and provide power to the CPU subsystem and associated peripherals.

The DeepSleep regulator is used to maintain power to a small number of blocks when in DeepSleep mode. These blocks include the ILO and WDT timers, BOD detector, SCB0, SRAM memories, Smart I/O, and other configuration memories. The DeepSleep regulator is enabled when in DeepSleep mode, and the core internal regulator is disabled. It is disabled when XRES_L is asserted (LOW) and when the core internal regulator is disabled.

The core internal regulator supports load currents up to 150 mA, and is operational during device startup (boot process), and in Active/Sleep modes.

The CYT2BL clock system provides clocks to all subsystems that require them, and glitch-free switching between different clock sources. In addition, the clock system ensures that no metastable conditions occur.

The clock system for CYT2BL consists of the 8-MHz IMO, two ILOs, three watchdog timers, a PLL, an FLL, five clock supervisors (CSV), a 3.988- to 33.34-MHz ECO, and a 32.768-kHz WCO.

The clock system supports two main clock domains: CLK_HF and CLK_LF.

- CLK_HFx are the active domain clocks. Each can use any of the high-frequency clock sources including IMO, EXT_CLK, ECO, FLL, or PLL.
- CLK_LF is a DeepSleep domain clock and provides source for MCWDT or RTC modules. The reference clock for

The IMO is the frequency reference in CYT2BL when no external reference is available or enabled. The IMO operates at a frequency of around 8 MHz.

An ILO is a low-power oscillator, nominally 32.768 kHz, which generates clocks for a watchdog timer when in DeepSleep mode. There are two ILOs to ensure clock supervisor (CSV) capability in DeepSleep mode. ILO-driven co n r(c)27.6an nevccr(ur)e5.4(a)t1.6(y)301.1(o O

CYT2BL contains three 1-Msps SAR ADCs. These ADCs can be clocked at up to 26.67 MHz and provide a 12-bit result in 26 clock cycles.

The references for all three SAR ADCs comes from a dedicated pair of inputs: VREFH and VREFL[7].

CYT2BL devices support up to 85 logical ADC channels, and external inputs from up to 67 I/Os. Each ADC also supports six internal connections for diagnostic and monitoring purposes. The number of ADC channels (per ADC and package type) are listed in .

Each ADC has a sequencer, which autonomously cycles through the configured channels (sequencer scan) with zero-switching overhead (that is, the aggregate sampling bandwidth, when clocked at 26.67 MHz, is equal to 1 Msps whether it is for a single channel or distributed over several channels). The sequencer switching is controlled through a state machine or firmware. The sequencer prioritizes trigger requests, enables the appropriate analog channel, controls ADC sampling, initiates ADC data conversion, manages results, and initiates subsequent conversions for repetitive or group conversions without CPU intervention.

Each SAR ADC has an analog multiplexer used to connect the signals to be measured to the ADC. It has 32 GPIO_STD inputs, one special GPIO_STD input for motor-sense, and six additional inputs to measure internal signals such as a band-gap reference, a temperature sensor, and power supplies. The device supports synchronous sampling of one motor-sense channel on each of the three ADCs.

CYT2BL has one temperature sensor that is shared by all three ADCs. The temperature sensor must only be sampled by one ADC at a time. Software post processing is required to convert the temperature sensor reading into kelvin or Celsius values.

To accommodate signals with varying source impedances and frequencies, it is possible to have different sample times programmed for each channel. Each ADC also supports range comparison, which allows fast detection of

CYT2BL contains up to 12 LIN channels. Each channel supports transmission/reception of data following the LIN protocol according to ISO standard 17987. Each LIN channel connects to an external transceiver through a 3-pin interface (including an enable function) and supports master and slave functionality. Each channel also supports classic and enhanced checksum, along with break detection during message reception and wake-up signaling. Break detection, sync field, checksum calculations, and error interrupts are handled in hardware.

CYT2BL contains up to four CXPI channels compliant with JASO D015 and ISO standard 20794 including the controller specification.

Each channel supports:

- Master and slave functionality
- Polling and event trigger method for both normal and long frames
- Non-return to zero (NRZ) and PWM signaling modes
- Collision resolution and carries sense multiple access
- Wakeup pulse generation and detection
- CRC8 and CRC16 for both normal and long frames
- Error detection
- Dedicated FIFO (16 B) for transmit and receive

The I/O port power source mapping is listed in . The associated supply determines the V_{OH}, V_{OL}, V_{IH}, and V_{IL} levels when configured for CMOS and automotive thresholds.

Px.y describes a particular bit “y” available within an I/O port “x.”

For example, P4.2 reads “port 4, bit 2”.

Each I/O implements the following:

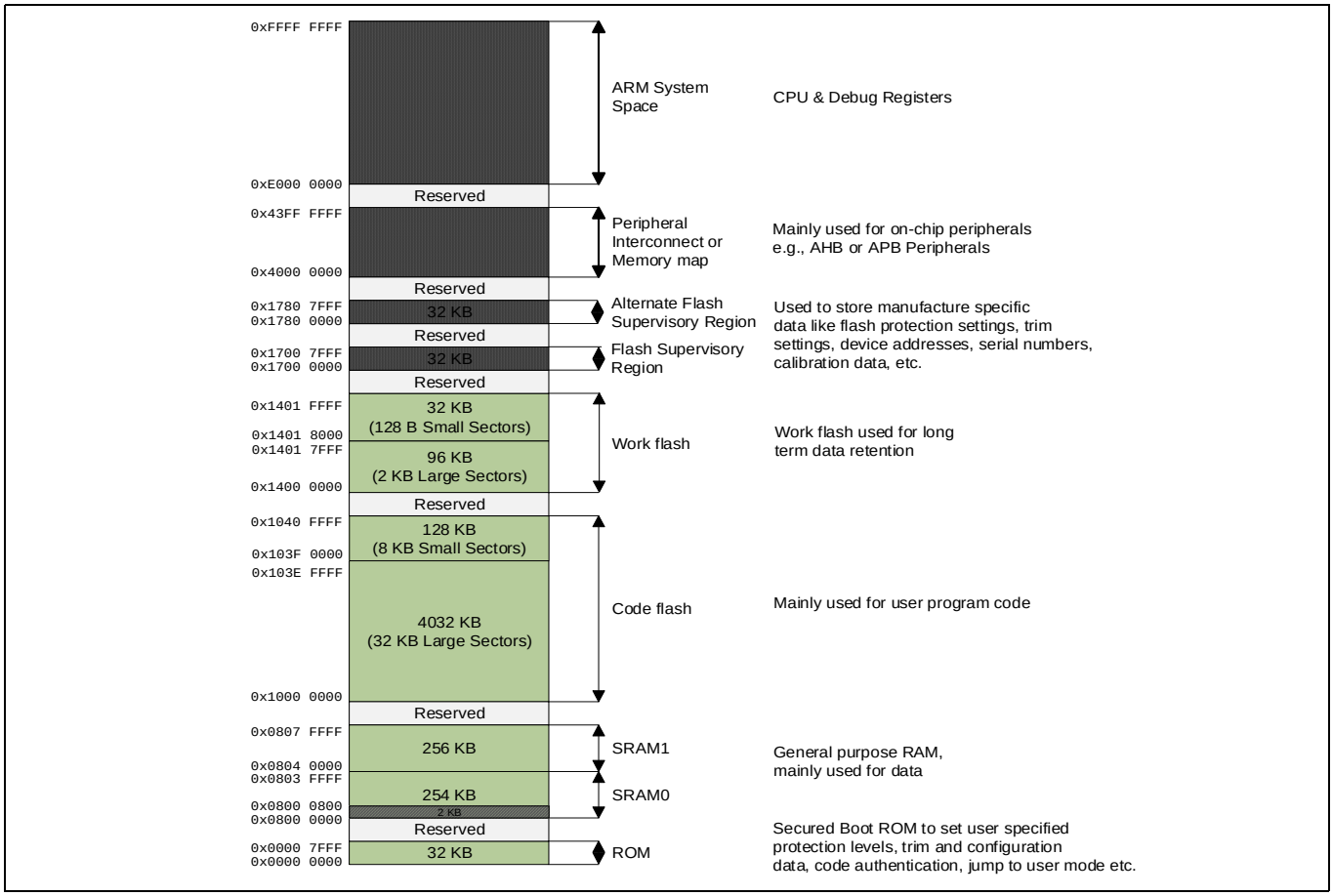
- Programmable drive mode
  - High impedance
  - Resistive pull-up
  - Resistive pull-down
  - Open drain with strong pull-down
  - Open drain with strong pull-up
  - Strong pull-up or pull-down
  - Weak pull-up or pull-down

CYT2BL has two types of programmable I/Os: GPIO Standard and GPIO Enhanced.

34ta

The CYT2BL microcontroller supports the memory spaces shown in :

- 4160 KB (4032 KB + 128 KB) of code-flash, used in the single- or dual-bank mode based on the associated bit in the flash control register
  - Single-bank mode - 4160 KB
  - Dual-bank mode - 2080 KB per bank
- 128 KB (96 KB + 32 KB) of work-flash, used in the single- or dual-bank mode based on the associated bit in the flash control register
  - Single-bank mode - 128 KB
  - Dual-bank mode - 64 KB per bank
- 32 KB of secure ROM
- 512 KB of SRAM (First 2 KB is reserved for internal usage)



[14, 15]

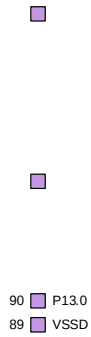
14. The size representation is not up to scale.
15. The first 2KB of SRAM is reserved and not available for users. User must keep the power of the first 32-KB block of SRAM0 in enabled or retained in all Active, LP Active, Sleep, LP Sleep, DeepSleep modes.

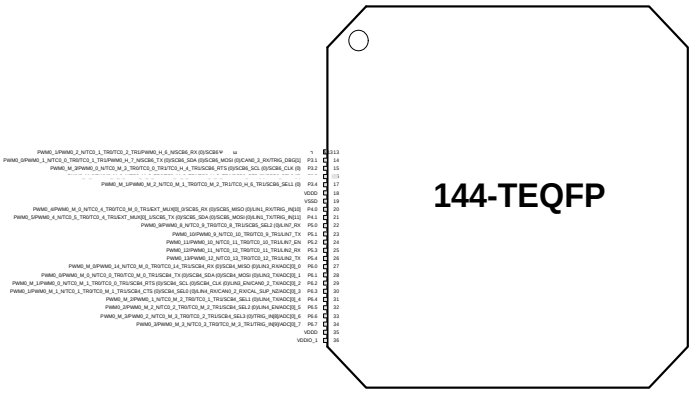
PERI	Peripheral interconnect	0x4000 0000			0	0
	Peripheral group (0, 1, 2, 3, 5, 6, 9)	0x4000 4000	7	0x20		
	Peripheral trigger group	0x4000 8000	11	0x400		
	Peripheral 1:1 trigger group	0x4000 C000	11	0x400		
PERI_MS	Peripheral interconnect, master interface	0x4001 0000			0	1
	PERI Programmable PPU	0x4001 0000	6[16]	0x40		
	PERI Fixed PPU	0x4001 0800	487	0x40		
Crypto	Cryptography component	0x4010 0000			1	0
CPUSS	CPU subsystem (CPUSS)	0x4020 0000			2	0
FAULT	Fault structure subsystem	0x4021 0000			2	1
	Fault structures					

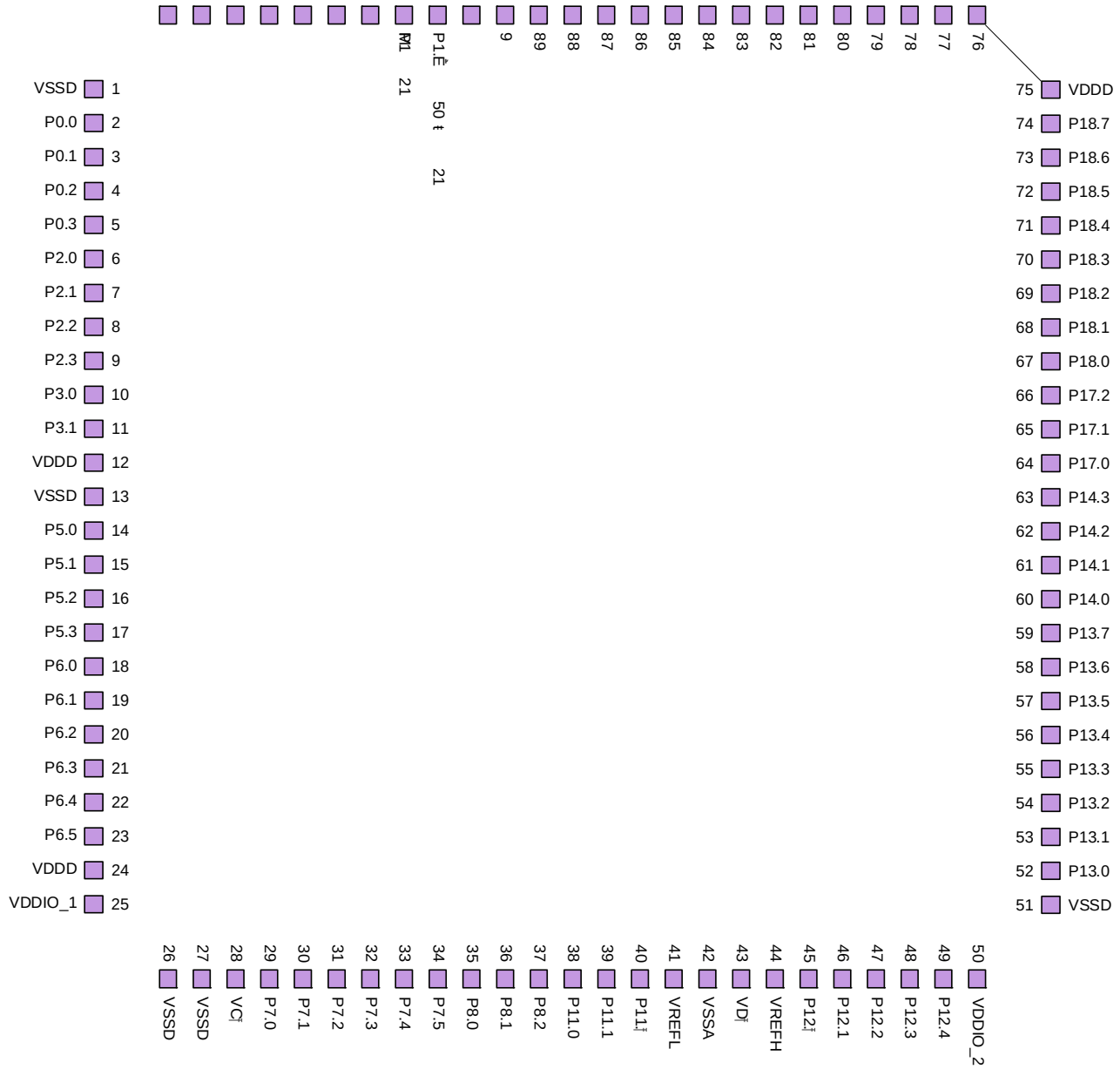
0x4010 99.fT-Tj-36.47t, m l7(, 2

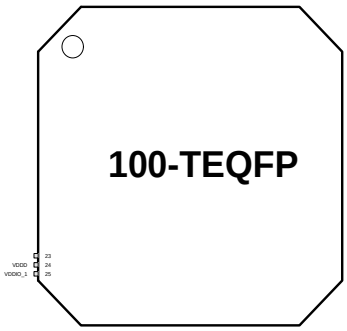
The following steps describe the start-up sequence:

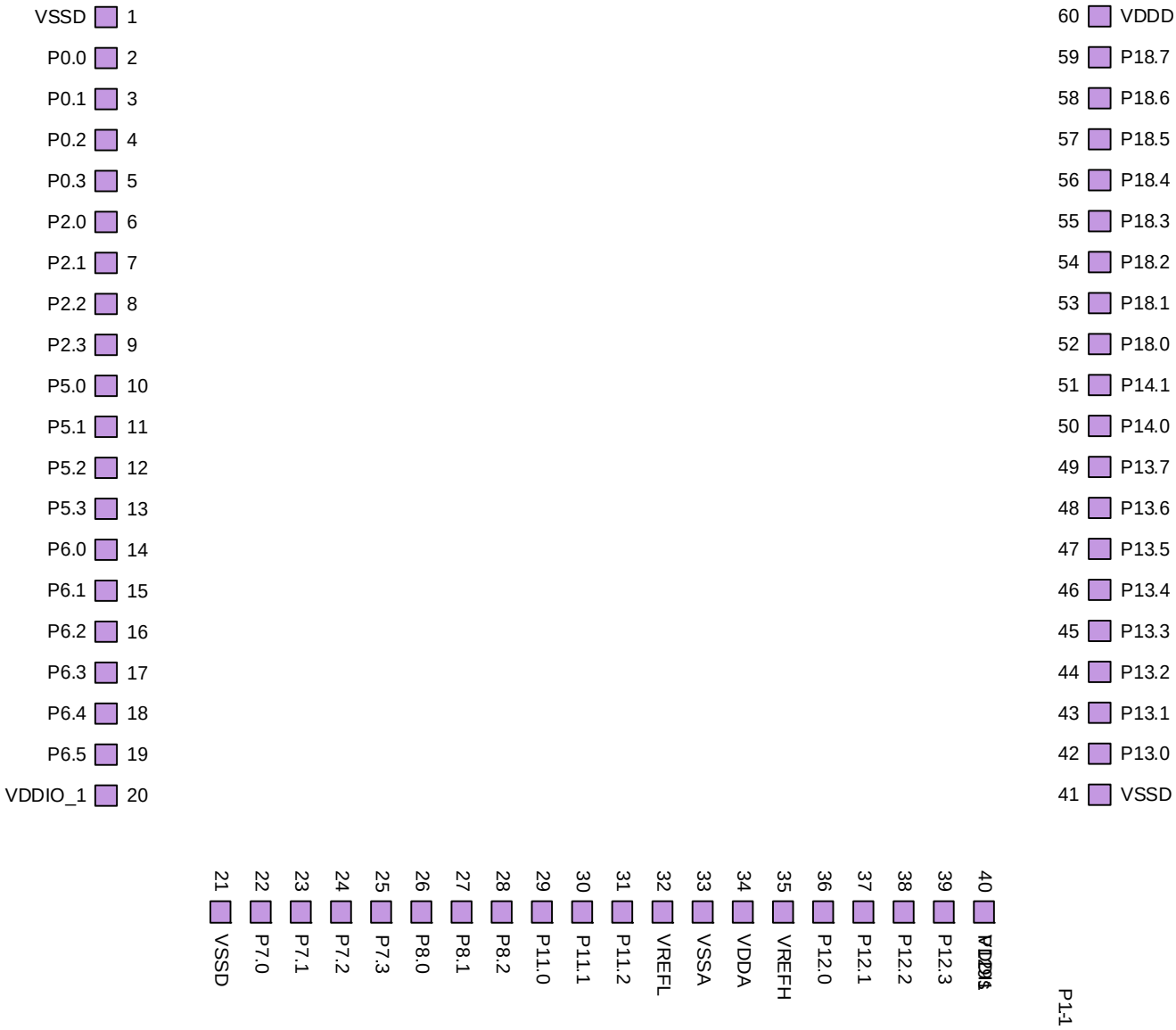
1. System Reset (@0x0000 0000)
2. CM0+ executes ROM boot (@0x0000 0004)
   i. Applies trims
   ii. Applies Debug Access port (DAP) access restrictions and system protection from eFuse and supervisory flash
   iii. Authenticates flash boot (only in SECURE life-cycle stage) and transfers control to it
3. CM0+ executes flash boot (from Supervisory flash @0x1700 2000)
   i. Debug pins are configured as per the SWD/JTAG spec[17]
   ii. Sets CM0+ vector offset register (CM0_VTOR part of the Arm® system space) to the beginning of flash (@0x1000 0000)
   iii. CM0+ branches to its Reset handler
4. CM0+ starts execution

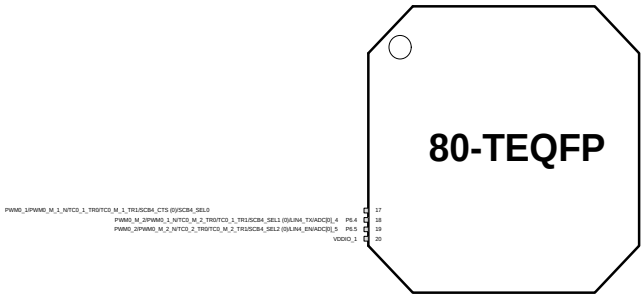

90 P13.0
89 VSSD

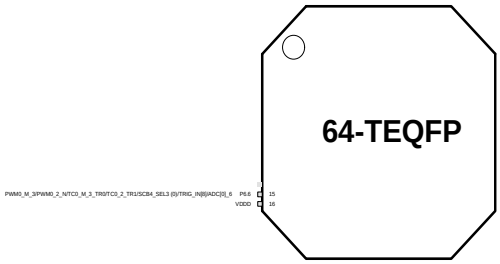

144-TEQFP


VSSD 1
P0.0 2
P0.1 3
P0.2 4
P0.3 5
P2.0 6
P2.1 7
P2.2 8
P2.3 9
P3.0 10
P3.1 11
VDDD 12
VSSD 13
P5.0 14
P5.1 15
P5.2 16
P5.3 17
P6.0 18
P6.1 19
P6.2 20
P6.3 21
P6.4 22
P6.5 23
VDDD 24
VDDIO_1 25
75 VDDD
74 P18.7
73 P18.6
72 P18.5
71 P18.4
70 P18.3
69 P18.2
68 P18.1
67 P18.0
66 P17.2
65 P17.1
64 P17.0
63 P14.3
62 P14.2
61 P14.1
60 P14.0
59 P13.7
58 P13.6
57 P13.5
56 P13.4
55 P13.3
54 P13.2
53 P13.1
52 P13.0
51 VSSD
26 VSSD
27 VSSD
28 VC
29 P7.0
30 P7.1
31 P7.2
32 P7.3
33 P7.4
34 P7.5
35 P8.0
36 P8.1
37 P8.2
38 P11.0
39 P11.1
40 P11
41 VREFL
42 VSSA
43 VD
44 VREFH
45 P12
46 P12.1
47 P12.2
48 P12.3
49 P12.4
50 VDDIO_2


100-TEQFP


VSSD 1
P0.0 2
P0.1 3
P0.2 4
P0.3 5
P2.0 6
P2.1 7
P2.2 8
P2.3 9
P5.0 10
P5.1 11
P5.2 12
P5.3 13
P6.0 14
P6.1 15
P6.2 16
P6.3 17
P6.4 18
P6.5 19
VDDIO_1 20
60 VDDD
59 P18.7
58 P18.6
57 P18.5
56 P18.4
55 P18.3
54 P18.2
53 P18.1
52 P18.0
51 P14.1
50 P14.0
49 P13.7
48 P13.6
47 P13.5
46 P13.4
45 P13.3
44 P13.2
43 P13.1
42 P13.0
41 VSSD
21 VSSD
22 P7.0
23 P7.1
24 P7.2
25 P7.3
26 P8.0
27 P8.1
28 P8.2
29 P11.0
30 P11.1
31 P11.2
32 VREFL
33 VSSA
34 VDDA
35 VREFH
36 P12.0
37 P12.1
38 P12.2
39 P12.3
40
P1-1


80-TEQFP
PWM0_1/PWM0_M_1_N/TC0_1_TR0/TC0_M_1_TR1/SCB4_CTS (0)/SCB4_SEL0 17
PWM0_M_2/PWM0_1_N/TC0_M_2_TR0/TC0_1_TR1/SCB4_SEL1 (0)/LIN4_TX/ADC[0]_4 P6.4 18
PWM0_2/PWM0_M_2_N/TC0_2_TR0/TC0_M_2_TR1/SCB4_SEL2 (0)/LIN4_EN/ADC[0]_5 P6.5 19
VDDIO_1 20


64-TEQFP
P6.6
VDDD
15
16

Most pins have alternate functionality, as specified in .
Port 11 has the following additional features:

- Ability to pass full-level analog signals to the SAR without clipping to V_{DDD} in cases where $V_{DDD} < V_{DDA}$
- Ability to simultaneously capture all three ADC signals with highest priority (ADC[0:2]_M)
- Lower noise for the most sensitive sensors

	[18]						[20]				
							[19]				
P0.0	GPIO_ENH	2	2	2	2	1			SCB0_MISO (0)		
P0.1	GPIO_ENH	3	3	3	3	2			SCB0_MOSI (0)		
P0.2	GPIO_ENH	4	4	4	4	3	SCB0_SCL (0)		SCB0_CLK (0)		
P0.3	GPIO_ENH	5	5	5	5	4	SCB0_SDA (0)		SCB0_SEL0 (0)		
P1.0	GPIO_STD	6	6	NA	NA	NA	SCB0_SCL (1)		SCB0_MISO (1)		
P1.1	GPIO_STD	7	7	NA	NA	NA	SCB0_SDA (1)		SCB0_MOSI (1)		
P1.2	GPIO_STD	8	NA	NA	NA	NA			SCB0_CLK (1)		
P1.3	GPIO_STD	9	NA	NA	NA	NA			SCB0_SEL0 (1)		
P2.0	GPIO_STD	10	8	6	6	5		SWJ_TRSTN	SCB0_SEL1 (0)		
P2.1	GPIO_STD	11	9	7	7	6			SCB0_SEL2 (0)		
P2.2	GPIO_STD	12	10	8	8	NA			SCB0_SEL3 (0)		
P2.3	GPIO_STD	13	11	9	9	NA					
P2.4	GPIO_STD	14	12	NA	NA	NA					
P2.5	GPIO_STD	15	NA	NA	NA	NA					
P3.0	GPIO_STD	16	13	10	NA	NA					
P3.1	GPIO_STD	17	14	11	NA	NA					
P3.2	GPIO_STD	18	15	NA	NA	NA					

18.High Speed I/O matrix connection (HCon) reference as per .
19.DeepSleep ordering (DS#0, DS#1, DS#2) does not have any impact on choosing any alternate functions; the HSIOM module handles the individual alternate function assignment.
20.All port pin functions available in DeepSleep mode are also available in Active mode.
21.Refer to for more information on pin multiplexer abbreviations used.
22.For any tion marked with an iden tifier (n), the AC timing is only guaranteed within the respective group "n".

P8.1	GPIO_STD	57	49	36	27	22	ADC[0]_16

(continued)

P13.4	GPIO_STD	94	78	56	46	NA	ADC[1]_16	SMARTIO13_4
P13.5	GPIO_STD	95	79	57	47	NA	ADC[1]_17	SMARTIO13_5
P13.6	GPIO_STD	96	80	58	48	NA	ADC[1]_18	SMARTIO13_6
P13.7	GPIO_STD	97	81	59	49	NA	ADC[1]_19	SMARTIO13_7
P14.0	GPIO_STD	98	82	60	50	38	ADC[1]_20	SMARTIO14_0
P14.1	GPIO_STD	99	83	61	51	39	ADC[1]_21	SMARTIO14_1
P14.2	GPIO_STD	100	84	62	NA	40	ADC[1]_22	SMARTIO14_2
P14.3	GPIO_STD	101	85	63	NA	NA	ADC[1]_23	SMARTIO14_3
P14.4	GPIO_STD	102	86	NA	NA	NA	ADC[1]_24	SMARTIO14_4
P14.5	GPIO_STD	103	87	NA	NA	NA	ADC[1]_25	SMARTIO14_5
P14.6	GPIO_STD	104	NA	NA	NA	NA	ADC[1]_26	SMARTIO14_6
P14.7	GPIO_STD	105	NA	NA	NA	NA	ADC[1]_27	SMARTIO14_7
P15.0	GPIO_STD	106	88	NA	NA	NA	ADC[1]_28	SMARTIO15_0
P15.1	GPIO_STD	107	89	NA	NA	NA	ADC[1]_29	SMARTIO15_1
P15.2	GPIO_STD	108	90	NA	NA	NA	ADC[1]_30	SMARTIO15_2
P15.3	GPIO_STD	109	91	NA	NA	NA	ADC[1]_31	SMARTIO15_3
P16.0	GPIO_STD	112	92	NA	NA	NA		
P16.1	GPIO_STD	113	93	NA	NA	NA		
P16.2	GPIO_STD	114	94	NA	NA	NA		
P16.3	GPIO_STD	115	NA	NA	NA	NA		
P17.0	GPIO_STD	116	95	64	NA	NA		SMARTIO17_0
P17.1	GPIO_STD	117	96	65	NA	NA		SMARTIO17_1
P17.2	GPIO_STD	118	97	66	NA	NA		SMARTIO17_2
P17.3	GPIO_STD	119	98	NA	NA	NA		SMARTIO17_3
P17.4	GPIO_STD	120	99	NA	NA	NA		SMARTIO17_4
P17.5	GPIO_STD	121	NA	NA	NA	NA		SMARTIO17_5
P17.6	GPIO_STD	122	NA	NA	NA	NA		SMARTIO17_6
P17.7	GPIO_STD	123	NA	NA	NA	NA		SMARTIO17_7
P18.0	GPIO_STD	124	100	67	52	41	ADC[2]_0	
P18.1	GPIO_STD	125	101	68	53	42	ADC[2]_1	
P18.2	GPIO_STD	126	102	69	54	NA	ADC[2]_2	

P18.3	GPIO_STD	127	103	70	55	43		ADC[2]_3
P18.4	GPIO_STD	128	104	71	56	44		ADC[2]_4
P18.5	GPIO_STD	129	105	72	57	45		ADC[2]_5
P18.6	GPIO_STD	130	106	73	58	46		ADC[2]_6
P18.7	GPIO_STD	131	107	74	59	47		ADC[2]_7
P19.0	GPIO_STD	134	110	77	62	NA		
P19.1	GPIO_STD	135	111	78	63	NA		
P19.2	GPIO_STD	136	112	79	NA	NA		
P19.3	GPIO_STD	137	113	80	NA	NA		
P19.4	GPIO_STD	138	114	NA	NA	NA		
P20.0	GPIO_STD	139	115	NA	NA	NA		
P20.1	GPIO_STD	140	116	NA	NA	NA		
P20.2	GPIO_STD	141	117	NA	NA	NA		
P20.3	GPIO_STD	142	118	NA	NA	NA		
P20.4	GPIO_STD	143	NA	NA	NA	NA		
P20.5	GPIO_STD	144	NA	NA	NA	NA		
P20.6	GPIO_STD	145	NA	NA	NA	NA		
P20.7	GPIO_STD	146	NA	NA	NA	NA		
P21.0	GPIO_STD	147	119	81	64	50		WCO_IN[23]
P21.1	GPIO_STD	148	120	82	65	51		WCO_OUT[23]
P21.2	GPIO_STD	149	121	83	66	52		ECO_IN[23]
P21.3	GPIO_STD	150	122	84	67	53		ECO_OUT[23]
P21.4[24]	GPIO_STD	151	NA	NA	NA	NA		HIBERNATE_WAKEUP[0]
P21.5	GPIO_STD	157	128	90	NA	NA		
P21.6	GPIO_STD	158	129	NA	NA	NA		
P21.7	GPIO_STD	159	NA	NA	NA	NA	RTC_CAL	
P22.0	GPIO_STD	160	130	91	73	59		
P22.1	GPIO_STD	161	131	92	74	NA		
P22.2	GPIO_STD	162	132	93	NA	NA		
P22.3	GPIO_STD	163	133	94	NA	NA		
P22.4	GPIO_STD	164	134	NA	NA	NA		

P22.5	GPIO_STD	165	135	NA	NA	NA
P22.6	GPIO_STD	166	136	NA		

VDDD	55, 48, 16	80, 69, 60	100, 86, 75, 24, 12	144, 124, 108, 35, 18	176, 153, 132, 110, 43, 22	Main digital supply
VSSD	57, 56, 49, 33, 17	71, 70, 61, 41, 21, 1	88, 87, 76, 51, 27, 26, 13, 1	126, 125, 109, 73, 38, 37, 19, 1	155, 154, 133, 111, 89, 46, 45, 23, 1	Main digital ground
VDDIO_1	NA	20	25	36	44	I/O supply for group 1
VDDIO_2	32	40	50	72	88	I/O supply for group 2
VCCD[25]	58	72	89, 28	127, 39	156, 47	Main regulated supply. Driven by LDO regulator
VREFH	29	35	44	65	79	High reference voltage for SAR
VREFL	26	32	41	62	76	Low reference voltage for SAR
VDDA	28	34	43	64	78	Main analog supply (for PASS SAR)
VSSA	27	33	42	63	77	Main analog ground

25.The V_{CCD} pins must be connected together to ensure a low-impedance connection. (see the requirement in).

P0.0	PWM0_18	PWM0_22_N	TC0_18_TR0	TC0_22_TR1		SCB0_RX (0)	SCB7_SDA (2)		LIN1_RX			
P0.1	PWM0_17	PWM0_18_N	TC0_17_TR0	TC0_18_TR1		SCB0_TX (0)	SCB7_SCL (2)		LIN1_TX			
P0.2	PWM0_14	PWM0_17_N	TC0_14_TR0	TC0_17_TR1		SCB0_RTS (0)			LIN1_EN	CAN0_1_TX		
P0.3	PWM0_13	PWM0_14_N	TC0_13_TR0	TC0_14_TR1		SCB0_CTS (0)				CAN0_1_RX		
P1.0	PWM0_12	PWM0_13_N	TC0_12_TR0	TC0_13_TR1	PWM0_H_4							
P1.1	PWM0_11	PWM0_12_N	TC0_11_TR0	TC0_12_TR1	PWM0_H_5							
P1.2	PWM0_10	PWM0_11_N	TC0_10_TR0	TC0_11_TR1	PWM0_H_6						TRIG_IN[0]	
P1.3	PWM0_8	PWM0_10_N	TC0_8_TR0	TC0_10_TR1	PWM0_H_7						TRIG_IN[1]	
P2.0	PWM0_7	PWM0_8_N	TC0_7_TR0	TC0_8_TR1	TC0_H_4_TR0	SCB7_RX (0)		SCB7_MISO (0)	LIN0_RX	CAN0_0_TX	TRIG_IN[2]	
P2.1	PWM0_6	PWM0_7_N	TC0_6_TR0	TC0_7_TR1	TC0_H_5_TR0	SCB7_TX (0)	SCB7_SDA (0)	SCB7_MOSI (0)	LIN0_TX	CAN0_0_RX	TRIG_IN[3]	
P2.2	PWM0_5	PWM0_6_N	TC0_5_TR0	TC0_6_TR1	TC0_H_6_TR0	SCB7_RTS (0)	SCB7_SCL (0)	SCB7_CLK (0)I-4.84(N)3(0)CN		TG-3.9(N_I-4.84(N[4])]TJ-45.5948 -2.0762 TD.00078Tc.0081 Tw[(A2.23-1683.2(P)-2.2(WM)-5(0_64-3536		

P5.0	PWM0_9	PWM0_8_N	TC0_9_TR0	TC0_8_TR1				SCB5_SEL2 (0)	LIN7_RX				
P5.1	PWM0_10	PWM0_9_N	TC0_10_TR0	TC0_9_TR1					LIN7_TX				
P5.2	PWM0_11	PWM0_10_N	TC0_11_TR0	TC0_10_TR1					LIN7_EN				
P5.3	PWM0_12	PWM0_11_N	TC0_12_TR0	TC0_11_TR1					LIN2_RX				
P5.4	PWM0_13	PWM0_12_N	TC0_13_TR0	TC0_12_TR1					LIN2_TX				
P5.5	PWM0_14	PWM0_13_N	TC0_14_TR0	TC0_13_TR1					LIN2_EN				
P6.0	PWM0_M_0	PWM0_14_N	TC0_M_0_TR0	TC0_14_TR1		SCB4_RX (0)		SCB4_MISO (0)	LIN3_RX				
P6.1	PWM0_0	PWM0_M_0_N	TC0_0_TR0	TC0_M_0_TR1		SCB4_TX (0)	SCB4_SDA (0)	SCB4_MOSI (0)	LIN3_TX				
P6.2	PWM0_M_1	PWM0_0_N	TC0_M_1_TR0	TC0_0_TR1		SCB4_RTS (0)	SCB4_SCL (0)	SCB4_CLK (0)	LIN3_EN	CAN0_2_TX			
P6.3	PWM0_1	PWM0_M_1_N	TC0_1_TR0	TC0_M_1_TR1		SCB4_CTS (0)		SCB4_SEL0 (0)	LIN4_RX	CAN0_2_RX			CAL_SUP_NZ
P6.4	PWM0_M_2	PWM0_1_N	TC0_M_2_TR0	TC0_1_TR1				SCB4_SEL1 (0)	LIN4_TX				
P6.5	PWM0_2	PWM0_M_2_N	TC0_2_TR0	TC0_M_2_TR1				SCB4_SEL2 (0)	LIN4_EN				
P6.6	PWM0_M_3	PWM0_2_N	TC0_M_3_TR0	TC0_2_TR1				SCB4_SEL3 (0)				TRIG_IN[8]	
P6.7	PWM0_3	PWM0_M_3_N	TC0_3_TR0	TC0_M_3_TR1								TRIG_IN[9]	
P7.0	PWM0_M_4	PWM0_3_N	TC0_M_4_TR0	TC0_3_TR1		SCB5_RX (1)		SCB5_MISO (1)	LIN4_RX		CXPI0_RX		
P7.1	PWM0_15	PWM0_M_4_N	TC0_15_TR0	TC0_M_4_TR1		SCB5_TX (1)	SCB5_SDA (1)	SCB5_MOSI (1)	LIN4_TX		CXPI0_TX		
P7.2	PWM0_M_5	PWM0_15_N	TC0_M_5_TR0	TC0_15_TR1		SCB5_RTS (1)	SCB5_SCL (1)	SCB5_CLK (1)	LIN4_EN		CXPI0_EN		
P7.3	PWM0_16	PWM0_M_5_N	TC0_16_TR0	TC0_M_5_TR1		SCB5_CTS (1)		SCB5_SEL0 (1)					
P7.4	PWM0_M_6	PWM0_16_N	TC0_M_6_TR0	TC0_16_TR1				SCB5_SEL1 (1)					
P7.5	PWM0_17	PWM0_M_6_N	TC0_17_TR0	TC0_M_6_TR1				SCB5_SEL2 (1)	LIN10_RX				
P7.6	PWM0_M_7	PWM0_17_N	TC0_M_7_TR0	TC0_17_TR1					LIN10_TX			TRIG_IN[16]	
P7.7	PWM0_18	PWM0_M_7_N	TC0_18_TR0	TC0_M_7_TR1					LIN10_EN			TRIG_IN[17]	
P8.0	PWM0_19	PWM0_18_N	TC0_19_TR0	TC0_18_TR1					LIN2_RX	CAN0_0_TX			
P8.1	PWM0_20	PWM0_19_N	TC0_20_TR0	TC0_19_TR1					LIN2_TX	CAN0_0_RX		TRIG_IN[14]	
P8.2	PWM0_21	PWM0_20_N	TC0_21_TR0	TC0_20_TR1					LIN2_EN			TRIG_IN[15]	
P8.3	PWM0_22	PWM0_21_N	TC0_22_TR0	TC0_21_TR1									TRIG_DBG[0]
P8.4	PWM0_23	PWM0_22_N	TC0_23_TR0	TC0_22_TR1									TRIG_DBG[1]
P9.0	PWM0_24	PWM0_23_N	TC0_24_TR0	TC0_23_TR1									
P9.1	PWM0_25	PWM0_24_N	TC0_25_TR0	TC0_24_TR1									
P9.2	PWM0_26	PWM0_25_N	TC0_26_TR0	TC0_25_TR1									
P9.3	PWM0_27	PWM0_26_N	TC0_27_TR0	TC0_26_TR1									

P10.0 PWM0_28 PWM0_27_N TC0_28_TR0 TC0_27_TR

P10.1 PWM0_29 PWM0_28_N TC0_29_TR0 TC0_28_TRSCB4_

P10.2 PWM0_30 PWM0_29_N TC0_30_TR0 TC0_29_TRS 4_RTS (1)CB4_SCL (1) SCB4_CLK (1) LIN8_RX

P10.3 PWM0_31 PWM0_30_N TC0_31_TR0 TC0_30_TR0_284_CTS (1) S4_SEL0 (1) LIN8_TX

P10.4 PWM0_32 PWM0_31_N TC0_32_TR0 TC0_31_TR0_226380.9(S)-7.7(CB4)7(_)-7.2(SEL1 (1))-1824.9(L)4.6(IN8_)-7.2(EN)]TJ-27.181 -2.0776 TD.0019 Tc-.0036 Tw[(P10)7.5(.5)-1190.8(PWM0_33)-3035.6(P)-1.1(WM0_32_N)-2739(TC0_33_TR0)-2871.8

P10.6 PWM0_34 PWM0_33_N TC0_34_TR0 TC0_33_TR CXPI0_TX

P10.7 PWM0_35 PWM0_34_N TC0_35_TR0 TC0_34_TR CXPI0_EN

P11.0

P11.0_]TJT*[(P11)7.8(.2)]TJ0 -2.0776 TD.0021 Tc[(P12)7.7(.0)-1190.6(PWM0_36)-3035.4(P)-.9(WM0_35_N)-2738.8(T)20.9(C0_36_TR0)-2871.6(T)29.5(C0_35_TR)-6.5M0_]TJ71.6121 0 TD.0012 Tc[(CA)-6.5MN0_2_T

LIN6_EN CAN0_2_RX TRIG_IN[21]

_MUX[1]_EN LIN6_RX

_MUX[1]_0 LIN6_TX

(0)MUX[1]_IN3]TJT*[(P12)7.6(.5)-1190.7(PWM0_41)-3035.5(P)-1(WM0_40_N)-2738.9(T)20.8(C0_41_TR0)-2871.7(T)29.4(C0_40_TR)-6.6(1)-3052.7(E)3.1(X)15(T)3.5(_MU)9.1(X)-2.2([1]_2)]TJ0 -2.0776 TD[(P12)7.7(.6)-1190.6(PWM0_42)-3035.4(P)-.9(WM0_41_N)-2738.8(T)20.9(C0_42_TR0)-2871.6(T)29.5

(0) LIN3_TX CXPI1_TX

S LIN3_EN CXPI1_EN

.6(IN8_)-7.2(R)10.1(X)]TJ-28.2586 -2.0862 TD.0024 Tc0 Tw[(P13)8(.5)-1190.3(PWM0_46)-3035.1(P)-.6(WM0_M)5.3(_)-6.2(00)8(_)-6.2(N)-1505.8(T)21.2MC0_46_TR0 TC0_M_10_TR_]TJ32.6035 0 TD.0016 Tc-.0033 TI7WM0_28820.4(P)-1.4(WM0_H_4_N)-17437.6(S)-7.5(CB3)7.2(_)-7MSEL2 (0)

(SEL3 (0))-1824.8(L)4.7(IN8_)-7.1(EN)-8489.4(CXPI)5.9(2_T)20.3(X)-1623r45TRIG_IN[22]

P14.4	PWM0_52	PWM0_51_N	TC0_52_TR0	TC0_51_TR1	TC0_H_4_TR0	SCB2_SEL1 (0)	LIN6_EN			
P14.5	PWM0_53	PWM0_52_N	TC0_53_TR0	TC0_52_TR1	TC0_H_4_TR1	SCB2_SEL2 (0)		CXPI2_RX		
P14.6	PWM0_54	PWM0_53_N	TC0_54_TR0	TC0_53_TR1	TC0_H_5_TR0			CXPI2_TX	TRIG_IN[24]	
P14.7	PWM0_55	PWM0_54_N	TC0_55_TR0	TC0_54_TR1	TC0_H_5_TR1			CXPI2_EN	TRIG_IN[25]	

P19.3	PWM0_28	PWM0_27_N	TC0_28_TR0	TC0_27_TR1	TC0_H_1_TR1	SCB2_CTS (1)		SCB2_SEL0 (1)			CXPI3_EN	TRIG_IN[29]	
P19.4	PWM0_29	PWM0_28_N	TC0_29_TR0	TC0_28_TR1	TC0_H_2_TR0			SCB2_SEL1 (1)					
P20.0	PWM0_30	PWM0_29_N	TC0_30_TR0	TC0_29_TR1	TC0_H_2_TR1			SCB2_SEL2 (1)	LIN5_RX				
P20.1	PWM0_49	PWM0_30_N	TC0_49_TR0	TC0_30_TR1	TC0_H_3_TR0				LIN5_TX				
P20.2	PWM0_48	PWM0_49_N	TC0_48_TR0	TC0_49_TR1	TC0_H_3_TR1				LIN5_EN				
P20.3	PWM0_47	PWM0_48_N	TC0_47_TR0	TC0_48_TR1		SCB1_RX (1)		SCB1_MISO (1)		CAN1_2_TX			
P20.4	PWM0_46	PWM0_47_N	TC0_46_TR0	TC0_47_TR1		SCB1_TX (1)	SCB1_SDA (1)	SCB1_MOSI (1)		CAN1_2_RX			
P20.5	PWM0_45	PWM0_46_N	TC0_45_TR0	TC0_46_TR1		SCB1_RTS (1)	SCB1_SCL (1)	SCB1_CLK (1)			CXPI3_RX		
P20.6	PWM0_44	PWM0_45_N	TC0_44_TR0	TC0_45_TR1		SCB1_CTS (1)		SCB1_SEL0 (1)			CXPI3_TX		
P20.7	PWM0_43	PWM0_44_N	TC0_43_TR0	TC0_44_TR1				SCB1_SEL1 (1)			CXPI3_EN		
P21.0	PWM0_42	PWM0_43_N	TC0_42_TR0	TC0_43_TR1				SCB1_SEL2 (1)					
P21.1	PWM0_41	PWM0_42_N	TC0_41_TR0	TC0_42_TR1									
P21.2	PWM0_40	PWM0_41_N	TC0_40_TR0	TC0_41_TR1								EXT_CLK	TRIG_DBG[1]
P21.3	PWM0_39	PWM0_40_N	TC0_39_TR0	TC0_40_TR1									
P21.4	PWM0_38	PWM0_39_N	TC0_38_TR0	TC0_39_TR1									
P21.5	PWM0_37	PWM0_38_N	TC0_37_TR0	TC0_38_TR1					LIN0_RX				
P21.6	PWM0_36	PWM0_38.8(T)20.9(C0_28_TR0)-TR0P98_TR1LIN0_RX											

P23.5	PWM0_24	PWM0_25_N	TC0_24_TR0	TC0_25_TR1		SCB7_SEL2 (1)	LIN9_RX

1	PWMx_y	TCPWM	TCPWM 16-bit PWM (no motor control), PWM_DT and PWM_PR line out, x-TCPWM block, y-counter number
2	PWMx_y_N	TCPWM	TCPWM 16-bit PWM (no motor control), PWM_DT and PWM_PR complementary line out (N), x-TCPWM block, y-counter number
3	PWMx_M_y	TCPWM	TCPWM 16-bit PWM with motor control line out, x-TCPWM block, y-counter number
4	PWMx_M_y_N	TCPWM	TCPWM 16-bit PWM with motor control complementary line out (N), x-TCPWM block, y-counter number
5	PWMx_H_y	TCPWM	TCPWM 32-bit PWM, PWM_DT and PWM_PR line out, x-TCPWM block, y-counter number
6	PWMx_H_y_N	TCPWM	TCPWM 32-bit PWM, PWM_DT and PWM_PR complementary line out (N), x-TCPWM block, y-counter number
7	TCx_y_TRz	TCPWM	TCPWM 16-bit dedicated counter input triggers, x-TCPWM block, y-counter number, z-trigger number
8	TCx_M_y_TRz	TCPWM	TCPWM 16-bit dedicated counter input triggers with motor

(continued)

39	ioss_interrupts_gpio_18_IRQn	DeepSleep	GPIO_STD Port #18 interrupt
40	ioss_interrupts_gpio_19_IRQn	DeepSleep	GPIO_STD Port #19 interrupt
41	ioss_interrupts_gpio_20_IRQn	DeepSleep	GPIO_STD Port #20 interrupt
42	ioss_interrupts_gpio_21_IRQn	DeepSleep	GPIO_STD Port #21 interrupt
43	ioss_interrupts_gpio_22_IRQn	DeepSleep	GPIO_STD Port #22 interrupt
44	ioss_interrupts_gpio_23_IRQn	DeepSleep	GPIO_STD Port #23 interrupt
45	cpuss_interrupt_crypto_IRQn	Active	Crypto accelerator interrupt
46	cpuss_interrupt_fm_IRQn	Active	Flash macro Interrupt
47	cpuss_interrupts_cm4_fp_IRQn	Active	CM4 floating point operation fault
48	cpuss_interrupts_cm0_cti_0_IRQn	Active	CM0+ CTI (Cross trigger interface) #0
49	cpuss_interrupts_cm0_cti_1_IRQn	Active	CM0+ CTI #1
50	cpuss_interrupts_cm4_cti_0_IRQn	Active	CM4 CTI #0
51	cpuss_interrupts_cm4_cti_1_IRQn	Active	CM4 CTI #1
52	evtgen_0_interrupt_IRQn	Active	Event Generator Active domain interrupt
53	canfd_0_interrupt0_IRQn	Active	CAN0, Consolidated interrupt #0 for all three channels
54	canfd_0_interrupt1_IRQn	Active	CAN0, Consolidated interrupt #1 for all three channels
55	canfd_1_interrupt0_IRQn	Active	CAN1, Consolidated interrupt #0 for all three channels
56	canfd_1_interrupt1_IRQn	Active	CAN1, Consolidated interrupt #1 for all three channels
57	canfd_0_interrupts0_0_IRQn	Active	CAN0, Interrupt #0, Channel #0
58	canfd_0_interrupts0_1_IRQn	Active	CAN0, Interrupt #0, Channel #1
59	canfd_0_interrupts0_2_IRQn	Active	CAN0, Interrupt #0, Channel #2
60	canfd_0_interrupts0_3_IRQn	Active	CAN0, Interrupt #0, Channel #3
61	canfd_0_interrupts1_0_IRQn	Active	CAN0, Interrupt #1, Channel #0
62	canfd_0_interrupts1_1_IRQn	Active	CAN0, Interrupt #1, Channel #1
63	canfd_0_interrupts1_2_IRQn	Active	CAN0, Interrupt #1, Channel #2
64	canfd_0_interrupts1_3_IRQn	Active	CAN0, Interrupt #1, Channel #3
65	canfd_1_interrupts0_0_IRQn	Active	CAN1, Interrupt #0, Channel #0
66	canfd_1_interrupts0_1_IRQn	Active	CAN1, Interrupt #0, Channel #1
67	canfd_1_interrupts0_2_IRQn	Active	CAN1, Interrupt #0, Channel #2
68	canfd_1_interrupts0_3_IRQn	Active	CAN1, Interrupt #0, Channel #3
69	canfd_1_interrupts1_0_IRQn	Active	CAN1, Interrupt #1, Channel #0
70	canfd_1_interrupts1_1_IRQn	Active	CAN1, Interrupt #1, Channel #1
71	canfd_1_interrupts1_2_IRQn	Active	CAN1, Interrupt #1, Channel #2
72	canfd_1_interrupts1_3_IRQn	Active	CAN1, Interrupt #1, Channel #3
73	lin_0_interrupts_0_IRQn	Active	LIN0, Channel #0 Interrupt
74	lin_0_interrupts_1_IRQn	Active	LIN0, Channel #1 Interrupt
75	lin_0_interrupts_2_IRQn	Active	LIN0, Channel #2 Interrupt
76	lin_0_interrupts_3_IRQn	Active	LIN0, Channel #3 Interrupt
77	lin_0_interrupts_4_IRQn	Active	LIN0, Channel #4 Interrupt

(continued)

78	lin_0_interrupts_5_IRQn	Active	LIN0, Channel #5 Interrupt
79	lin_0_interrupts_6_IRQn	Active	LIN0, Channel #6 Interrupt
80	lin_0_interrupts_7_IRQn	Active	LIN0, Channel #7 Interrupt
81	lin_0_interrupts_8_IRQn	Active	LIN0, Channel #8 Interrupt
82	lin_0_interrupts_9_IRQn	Active	LIN0, Channel #9 Interrupt
83	lin_0_interrupts_10_IRQn	Active	LIN0, Channel #10 Interrupt
84	lin_0_interrupts_11_IRQn	Active	LIN0, Channel #11 Interrupt
85	cxpi_0_interrupts_0_IRQn	Active	CXPI0 Channel #0 Interrupt
86	cxpi_0_interrupts_1_IRQn	Active	CXPI0 Channel #1 Interrupt
87	cxpi_0_interrupts_2_IRQn	Active	CXPI0 Channel #2 Interrupt
88	cxpi_0_interrupts_3_IRQn	Active	CXPI0 Channel #3 Interrupt
89	scb_1_interrupt_IRQn	Active	SCB1 Interrupt
90	scb_2_interrupt_IRQn	Active	SCB2 Interrupt
91	scb_3_interrupt_IRQn	Active	SCB3 Interrupt
92	scb_4_interrupt_IRQn	Active	SCB4 Interrupt
93	scb_5_interrupt_IRQn	Active	SCB5 Interrupt
94	scb_6_interrupt_IRQn	Active	SCB6 Interrupt
95	scb_7_interrupt_IRQn	Active	SCB7 Interrupt
96	pass_0_interrupts_sar_0_IRQn	Active	SAR0, Logical channel #0 interrupt
97	pass_0_interrupts_sar_1_IRQn	Active	SAR0, Logical channel #1 interrupt
98	pass_0_interrupts_sar_2_IRQn	Active	SAR0, Logical channel #2 interrupt
99	pass_0_interrupts_sar_3_IRQn	Active	SAR0, Logical channel #3 interrupt
100	pass_0_interrupts_sar_4_IRQn	Active	SAR0, Logical channel #4 interrupt
101	pass_0_interrupts_sar_5_IRQn	Active	SAR0, Logical channel #5 interrupt
102	pass_0_interrupts_sar_6_IRQn	Active	SAR0, Logical channel #6 interrupt
103	pass_0_interrupts_sar_7_IRQn	Active	SAR0, Logical channel #7 interrupt
104	pass_0_interrupts_sar_8_IRQn	Active	SAR0, Logical channel #8 interrupt
105	pass_0_interrupts_sar_9_IRQn	Active	SAR0, Logical channel #9 interrupt
106	pass_0_interrupts_sar_10_IRQn	Active	SAR0, Logical channel #10 interrupt
107	pass_0_interrupts_sar_11_IRQn	Active	SAR0, Logical channel #11 interrupt
108	pass_0_interrupts_sar_12_IRQn	Active	SAR0, Logical channel #12 interrupt
109	pass_0_interrupts_sar_13_IRQn	Active	SAR0, Logical channel #13 interrupt
110	pass_0_interrupts_sar_14_IRQn	Active	SAR0, Logical channel #14 interrupt
111	pass_0_interrupts_sar_15_IRQn	Active	SAR0, Logical channel #15 interrupt
112	pass_0_interrupts_sar_16_IRQn	Active	SAR0, Logical channel #16 interrupt
113	pass_0_interrupts_sar_17_IRQn	Active	SAR0, Logical channel #17 interrupt
114	pass_0_interrupts_sar_18_IRQn	Active	SAR0, Logical channel #18 interrupt
115	pass_0_interrupts_sar_19_IRQn	Active	SAR0, Logical channel #19 interrupt
116	pass_0_interrupts_sar_20_IRQn	Active	SAR0, Logical channel #20 interrupt
117	pass_0_interrupts_sar_21_IRQn	Active	SAR0, Logical channel #21 interrupt
118	pass_0_interrupts_sar_22_IRQn	Active	SAR0, Logical channel #22 interrupt

119	pass_0_interrupts_sar_23_IRQn	Active	SAR0, Logical channel #23 interrupt
120	pass_0_interrupts_sar_32_IRQn	Active	SAR1, Logical channel #0 interrupt
121	pass_0_interrupts_sar_33_IRQn	Active	SAR1, Logical channel #1 interrupt
122	pass_0_interrupts_sar_34_IRQn	Active	SAR1, Logical channel #2 interrupt
123	pass_0_interrupts_sar_35_IRQn	Active	SAR1, Logical channel #3 interrupt
124	pass_0_interrupts_sar_36_IRQn	Active	SAR1, Logical channel #4 interrupt
125	pass_0_interrupts_sar_37_IRQn	Active	SAR1, Logical channel #5 interrupt
126	pass_0_interrupts_sar_38_IRQn	Active	SAR1, Logical channel #6 interrupt
127	pass_0_interrupts_sar_39_IRQn	Active	SAR1, Logical channel #7 interrupt
128	pass_0_interrupts_sar_40_IRQn	Active	SAR1, Logical channel #8 interrupt
129	pass_0_interrupts_sar_41_IRQn	Active	SAR1, Logical channel #9 interrupt
130	pass_0_interrupts_sar_42_IRQn	Active	SAR1, Logical channel #10 interrupt
131	pass_0_interrupts_sar_43_IRQn	Active	SAR1, Logical channel #11 interrupt
132	pass_0_interrupts_sar_44_IRQn	Active	SAR1, Logical channel #12 interrupt
133	pass_0_interrupts_sar_45_IRQn	Active	SAR1, Logical channel #13 interrupt
134	pass_0_interrupts_sar_46_IRQn	Active	SAR1, Logical channel #14 interrupt
135	pass_0_interrupts_sar_47_IRQn	Active	SAR1, Logical channel #15 interrupt
136	pass_0_interrupts_sar_48_IRQn	Active	SAR1, Logical channel #16 interrupt
137	pass_0_interrupts_sar_49_IRQn	Active	SAR1, Logical channel #17 interrupt
138	pass_0_interrupts_sar_50_IRQn	Active	SAR1, Logical channel #18 interrupt
139	pass_0_interrupts_sar_51_IRQn	Active	SAR1, Logical channel #19 interrupt
140	pass_0_interrupts_sar_52_IRQn	Active	SAR1, Logical channel #20 interrupt
141	pass_0_interrupts_sar_53_IRQn	Active	SAR1, Logical channel #21 interrupt
142	pass_0_interrupts_sar_54_IRQn	Active	SAR1, Logical channel #22 interrupt
143	pass_0_interrupts_sar_55_IRQn	Active	SAR1, Logical channel #23 interrupt
144	pass_0_interrupts_sar_56_IRQn	Active	SAR1, Logical channel #24 interrupt
145	pass_0_interrupts_sar_57_IRQn	Active	SAR1, Logical channel #25 interrupt
146	pass_0_interrupts_sar_58_IRQn	Active	SAR1, Logical channel #26 interrupt
147	pass_0_interrupts_sar_59_IRQn	Active	SAR1, Logical channel #27 interrupt
148	pass_0_interrupts_sar_60_IRQn	Active	SAR1, Logical channel #28 interrupt
149	pass_0_interrupts_sar_61_IRQn	Active	SAR1, Logical channel #29 interrupt
150	pass_0_interrupts_sar_62_IRQn	Active	SAR1, Logical channel #30 interrupt
151	pass_0_interrupts_sar_63_IRQn	Active	SAR1, Logical channel #31 interrupt
152	pass_0_interrupts_sar_64_IRQn	Active	SAR2, Logical channel #0 interrupt
153	pass_0_interrupts_sar_65_IRQn	Active	SAR2, Logical channel #1 interrupt
154	pass_0_interrupts_sar_66_IRQn	Active	SAR2, Logical channel #2 interrupt
155	pass_0_interrupts_sar_67_IRQn	Active	SAR2, Logical channel #3 interrupt
156	pass_0_interrupts_sar_68_IRQn	Active	SAR2, Logical channel #4 interrupt
157	pass_0_interrupts_sar_69_IRQn	Active	SAR2, Logical channel #5 interrupt
158	pass_0_interrupts_sar_70_IRQn	Active	SAR2, Logical channel #6 interrupt
159	pass_0_interrupts_sar_71_IRQn	Active	SAR2, Logical channel #7 interrupt

(continued)

160	cpuss_interrupts_dmac_0_IRQn	Active	CPUSS M-DMA0, Channel #0 Interrupt
161	cpuss_interrupts_dmac_1_IRQn	Active	CPUSS M-DMA0, Channel #1 Interrupt
162	cpuss_interrupts_dmac_2_IRQn	Active	CPUSS M-DMA0, Channel #2 Interrupt
163	cpuss_interrupts_dmac_3_IRQn	Active	CPUSS M-DMA0, Channel #3 Interrupt
164	cpuss_interrupts_dw0_0_IRQn	Active	CPUSS P-DMA0, Channel #0 Interrupt
165	cpuss_interrupts_dw0_1_IRQn	Active	CPUSS P-DMA0, Channel #1 Interrupt
166	cpuss_interrupts_dw0_2_IRQn	Active	CPUSS P-DMA0, Channel #2 Interrupt
167	cpuss_interrupts_dw0_3_IRQn	Active	CPUSS P-DMA0, Channel #3 Interrupt
168	cpuss_interrupts_dw0_4_IRQn	Active	CPUSS P-DMA0, Channel #4 Interrupt
169	cpuss_interrupts_dw0_5_IRQn	Active	CPUSS P-DMA0, Channel #5 Interrupt
170	cpuss_interrupts_dw0_6_IRQn	Active	CPUSS P-DMA0, Channel #6 Interrupt
171	cpuss_interrupts_dw0_7_IRQn	Active	CPUSS P-DMA0, Channel #7 Interrupt
172	cpuss_interrupts_dw0_8_IRQn	Active	CPUSS P-DMA0, Channel #8 Interrupt
173	cpuss_interrupts_dw0_9_IRQn	Active	CPUSS P-DMA0, Channel #9 Interrupt
174	cpuss_interrupts_dw0_10_IRQn	Active	CPUSS P-DMA0, Channel #10 Interrupt
175	cpuss_interrupts_dw0_11_IRQn	Active	CPUSS P-DMA0, Channel #11 Interrupt
176	cpuss_interrupts_dw0_12_IRQn	Active	CPUSS P-DMA0, Channel #12 Interrupt
177	cpuss_interrupts_dw0_13_IRQn	Active	CPUSS P-DMA0, Channel #13 Interrupt
178	cpuss_interrupts_dw0_14_IRQn	Active	CPUSS P-DMA0, Channel #14 Interrupt
179	cpuss_interrupts_dw0_15_IRQn	Active	CPUSS P-DMA0, Channel #15 Interrupt
180	cpuss_interrupts_dw0_16_IRQn	Active	CPUSS P-DMA0, Channel #16 Interrupt
181	cpuss_interrupts_dw0_17_IRQn	Active	CPUSS P-DMA0, Channel #17 Interrupt
182	cpuss_interrupts_dw0_18_IRQn	Active	CPUSS P-DMA0, Channel #18 Interrupt
183	cpuss_interrupts_dw0_19_IRQn	Active	CPUSS P-DMA0, Channel #19 Interrupt
184	cpuss_interrupts_dw0_20_IRQn	Active	CPUSS P-DMA0, Channel #20 Interrupt
185	cpuss_interrupts_dw0_21_IRQn	Active	CPUSS P-DMA0, Channel #21 Interrupt
186	cpuss_interrupts_dw0_22_IRQn	Active	CPUSS P-DMA0, Channel #22 Interrupt
187	cpuss_interrupts_dw0_23_IRQn	Active	CPUSS P-DMA0, Channel #23 Interrupt
188	cpuss_interrupts_dw0_24_IRQn	Active	CPUSS P-DMA0, Channel #24 Interrupt
189	cpuss_interrupts_dw0_25_IRQn	Active	CPUSS P-DMA0, Channel #25 Interrupt
190	cpuss_interrupts_dw0_26_IRQn	Active	CPUSS P-DMA0, Channel #26 Interrupt
191	cpuss_interrupts_dw0_27_IRQn	Active	CPUSS P-DMA0, Channel #27 Interrupt
192	cpuss_interrupts_dw0_28_IRQn	Active	CPUSS P-DMA0, Channel #28 Interrupt
193	cpuss_interrupts_dw0_29_IRQn	Active	CPUSS P-DMA0, Channel #29 Interrupt
194	cpuss_interrupts_dw0_30_IRQn	Active	CPUSS P-DMA0, Channel #30 Interrupt
195	cpuss_interrupts_dw0_31_IRQn	Active	CPUSS P-DMA0, Channel #31 Interrupt
196	cpuss_interrupts_dw0_32_IRQn	Active	CPUSS P-DMA0, Channel #32 Interrupt
197	cpuss_interrupts_dw0_33_IRQn	Active	CPUSS P-DMA0, Channel #33 Interrupt
198	cpuss_interrupts_dw0_34_IRQn	Active	CPUSS P-DMA0, Channel #34 Interrupt
199	cpuss_interrupts_dw0_35_IRQn	Active	CPUSS P-DMA0, Channel #35 Interrupt
200	cpuss_interrupts_dw0_36_IRQn	Active	CPUSS P-DMA0, Channel #36 Interrupt

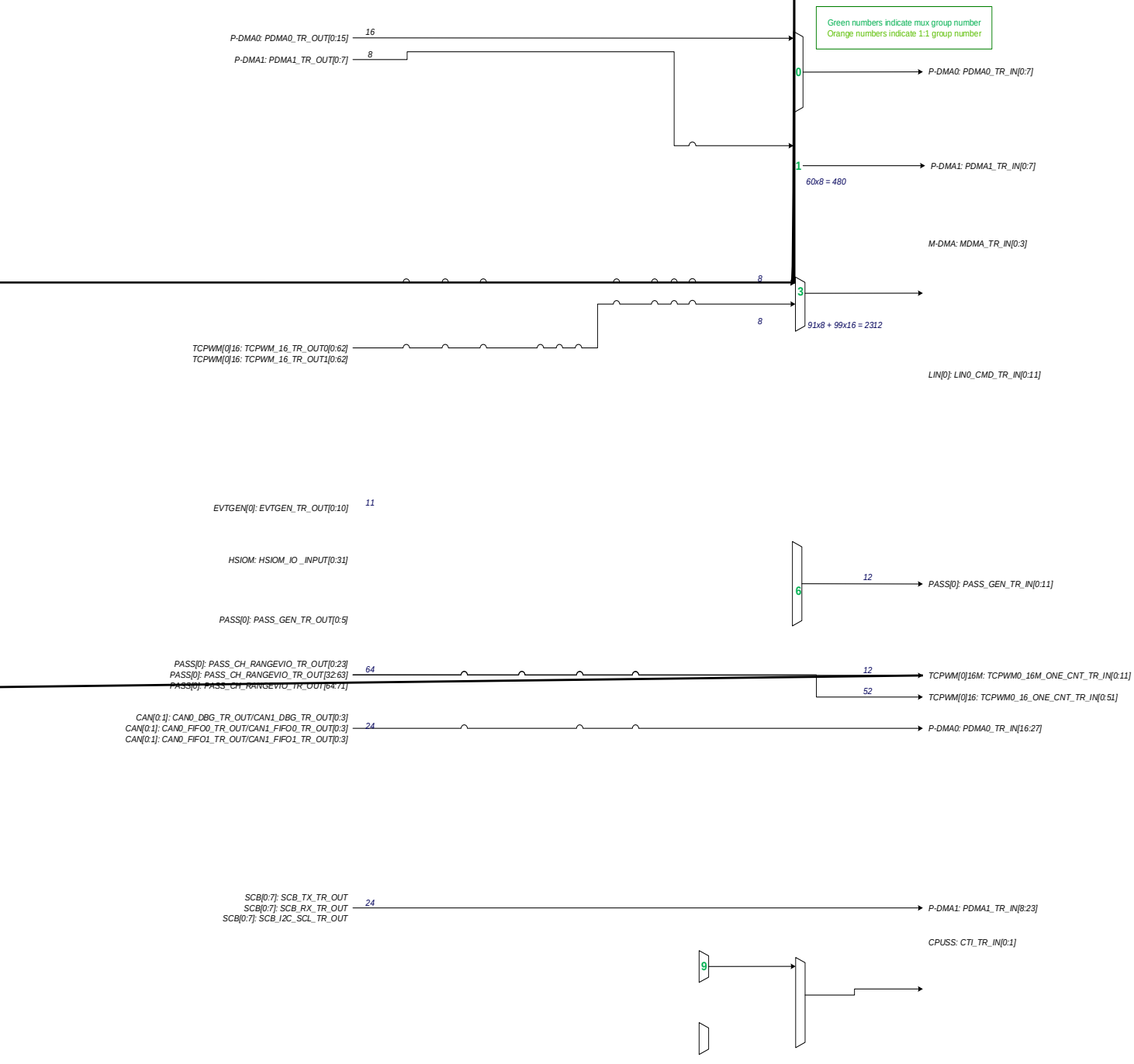
(continued)

365	tcpwm_0_interrupts_258_IRQn	Active	TCPWM0 Group #1, Counter #2 Interrupt
366	tcpwm_0_interrupts_259_IRQn	Active	TCPWM0 Group #1, Counter #3 Interrupt
367	tcpwm_0_interrupts_260_IRQn	Active	TCPWM0 Group #1, Counter #4 Interrupt
368	tcpwm_0_interrupts_261_IRQn	Active	TCPWM0 Group #1, Counter #5 Interrupt
369	tcpwm_0_interrupts_262_IRQn	Active	TCPWM0 Group #1, Counter #6 Interrupt
370	tcpwm_0_interrupts_263_IRQn	Active	TCPWM0 Group #1, Counter #7 Interrupt
371	tcpwm_0_interrupts_264_IRQn	Active	TCPWM0 Group #1, Counter #8 Interrupt
372	tcpwm_0_interrupts_265_IRQn	Active	TCPWM0 Group #1, Counter #9 Interrupt
372	tcpwm_0_interrupts_266_IRQn	Active	TCPWM0 Group #1, Counter #10 Interrupt
374	tcpwm_0_interrupts_267_IRQn	Active	TCPWM0 Group #1, Counter #11 Interrupt
375	tcpwm_0_interrupts_512_IRQn	Active	TCPWM0 Group #2, Counter #0 Interrupt
376	tcpwm_0_interrupts_513_IRQn	Active	TCPWM0 Group #2, Counter #1 Interrupt
377	tcpwm_0_interrupts_514_IRQn	Active	TCPWM0 Group #2, Counter #2 Interrupt
378	tcpwm_0_interrupts_515_IRQn	Active	TCPWM0 Group #2, Counter #3 Interrupt
379	tcpwm_0_interrupts_516_IRQn	Active	TCPWM0 Group #2, Counter #4 Interrupt
380	tcpwm_0_interrupts_517_IRQn	Active	TCPWM0 Group #2, Counter #5 Interrupt
381	tcpwm_0_interrupts_518_IRQn	Active	TCPWM0 Group #2, Counter #6 Interrupt
382	tcpwm_0_interrupts_519_IRQn	Active	TCPWM0 Group #2, Counter #7 Interrupt

0	CPUIntIdx0_IRQn[30]	DeepSleep	CPU user interrupt #0
1	CPUIntIdx1_IRQn[30]	DeepSleep	CPU user interrupt #1
2	CPUIntIdx2_IRQn	DeepSleep	CPU user interrupt #2
3	CPUIntIdx3_IRQn	DeepSleep	CPU user interrupt #3
4	CPUIntIdx4_IRQn	DeepSleep	CPU user interrupt #4
5	CPUIntIdx5_IRQn	DeepSleep	CPU user interrupt #5
6	CPUIntIdx6_IRQn	DeepSleep	CPU user interrupt #6
7	CPUIntIdx7_IRQn	DeepSleep	CPU user interrupt #7
8	Internal0_IRQn	Active	Internal software interrupt #0
9	Internal1_IRQn	Active	Internal software interrupt #1
10	Internal2_IRQn	Active	Internal software interrupt #2
11	Internal3_IRQn	Active	Internal software interrupt #3
12	Internal4_IRQn	Active	Internal software interrupt #4
13	Internal5_IRQn	Active	Internal software interrupt #5
14	Internal6_IRQn	Active	Internal software interrupt #6
15	Internal7_IRQn	Active	Internal software interrupt #7

30.User interrupt cannot be used for CM0+ application, as it is used internally by system calls. Note, this does not impact CM4 application.

Trigger multiplexer



[31]

	: PDMA0_TR (P-DMA0 trigger multiplexer)	
1:16[32]	PDMA0_TR_OUT[0:15]	Allow P-DMA0 to chain to itself, useful for triggering once per row for 2D transfer
17:24	PDMA1_TR_OUT[0:7]	Cross connections from P-DMA1 to P-DMA0, Channels 0-7 are used
25:28	MDMA_TR_OUT[0:3]	Cross connections from M-DMA0 to P-DMA0
29:32	FAULT_TR_OUT[0:3]	Allow faults to initiate data transfer for debug purposes
33:34	CTI_TR_OUT[0:1]	Trace events
35:38	EVTGEN_TR_OUT[3:6]	EVTGEN triggers
39:54	HSIOM_IO_INPUT[0:15]	I/O inputs
	: PDMA1_TR (P-DMA1 trigger multiplexer)	
1:16	PDMA0_TR_OUT[0:15]	Allow P-DMA0 to trigger P-DMA1
17:24	PDMA1_TR_OUT[0:7]	Allow P-DMA1 to chain to itself, useful for triggering once per row for 2D transfer
25:28	MDMA_TR_OUT[0:3]	Allow M-DMA0 to trigger P-DMA0
29:32	FAULT_TR_OUT[0:3]	Allow faults to initiate data transfer for debug purposes
33:34	CTI_TR_OUT[0:1]	Trace events
35:38	EVTGEN_TR_OUT[7:10]	EVTGEN triggers
39:54	HSIOM_IO_INPUT[16:31]	I/O inputs
55:60	PASS_GEN_TR_OUT[0:5]	PASS SAR events
	: MDMA (M-DMA0 trigger multiplexer)	
1:4	MDMA_TR_OUT[0:3]	Allow M-DMA0 to trigger itself
	: TCPWM_TO_PDMA0 (TCPWM0 to P-DMA0 trigger multiplexer)	
1:8	TCPWM_32_TR_OUT0[0:7]	32-bit TCPWM0 counters
9:20	TCPWM_16M_TR_OUT0[0:11]	16-bit motor enhanced TCPWM0 counters
21:83	TCPWM_16_TR_OUT0[0:62]	16-bit TCPWM0 counters
84:87		

(continued)

: TCPWM_IN (TCPWM0 Trigger Multiplexer)		
1:16	PDMA0_TR_OUT[0:15]	General-purpose P-DMA0 triggers
17:24	PDMA1_TR_OUT[0:7]	General-purpose P-DMA1 triggers
25:28	MDMA_TR_OUT[0:3]	M-DMA0 triggers
29:30	CTI_TR_OUT[0:1]	Trace events
31:34	FAULT_TR_OUT[0:3]	Fault events
35:40	PASS_GEN_TR_OUT[0:5]	PASS SAR events
41:72	HSIOM_IO_INPUT[0:31]	I/O inputs
73	SCB_TX_TR_OUT[0]	SCB0 TX trigger
74	SCB_RX_TR_OUT[0]	SCB0 RX trigger
75	SCB_I2C_SCL_TR_OUT[0]	SCB0 I^2C trigger
76	SCB_TX_TR_OUT[1]	SCB1 TX trigger
77	SCB_RX_TR_OUT[1]	SCB1 RX trigger
78	SCB_I2C_SCL_TR_OUT[1]	SCB1 I^2C trigger
79	SCB_TX_TR_OUT[2]	SCB2 TX trigger
80	SCB_RX_TR_OUT[2]	SCB2 RX trigger
81	SCB_I2C_SCL_TR_OUT[2]	SCB2 I^2C trigger
82	SCB_TX_TR_OUT[3]	SCB3 TX trigger
83	SCB_RX_TR_OUT[3]	SCB3 RX trigger
84	SCB_I2C_SCL_TR_OUT[3]	SCB3 I^2C trigger
85	SCB_TX_TR_OUT[4]	SCB4 TX trigger
86	SCB_RX_TR_OUT[4]	SCB4 RX trigger
87	SCB_I2C_SCL_TR_OUT[4]	SCB4 I^2C trigger
88	SCB_TX_TR_OUT[5]	SCB5 TX trigger
89	SCB_RX_TR_OUT[5]	SCB5 RX trigger
90	SCB_I2C_SCL_TR_OUT[5]	SCB5 I^2C trigger
91	SCB_TX_TR_OUT[6]	SCB6 TX trigger
92	SCB_RX_TR_OUT[6]	SCB6 RX trigger
93	SCB_I2C_SCL_TR_OUT[6]	SCB6 I^2C trigger
94	SCB_TX_TR_OUT[7]	SCB7 TX trigger
95	SCB_RX_TR_OUT[7]	SCB7 RX trigger
96	SCB_I2C_SCL_TR_OUT[7]	SCB7 I^2C trigger
97:100	CAN0_DBG_TR_OUT[0:3]	CAN0 M-DMA0 events
101:104	CAN0_FIFO0_TR_OUT[0:3]	CAN0 FIFO0 events
105:108	CAN0_FIFO1_TR_OUT[0:3]	CAN0 FIFO1 events
109:112	CAN1_DBG_TR_OUT[0:3]	CAN1 M-DMA0 events
113:116	CAN1_FIFO0_TR_OUT[0:3]	CAN1 FIFO0 events
117:120	CAN1_FIFO1_TR_OUT[0:3]	CAN1 FIFO1 events
121:124	CXPI_TX_TR_OUT[0:3]	CXPI transmit events
125:128	CXPI_RX_TR_OUT[0:3]	CXPI receive events

(continued)

129:136	EVTGEN_TR_OUT[3:10]	EVTGEN triggers
: PASS (PASS SAR trigger multiplexer)		
1:16	PDMA0_TR_OUT[0:15]	General purpose P-DMA0 triggers
17:18	CTI_TR_OUT[0:1]	Trace events
19:22	FAULT_TR_OUT[0:3]	Fault events
23:25	EVTGEN_TR_OUT[0:2]	EVTGEN triggers
26:31	PASS_GEN_TR_OUT[0:5]	PASS SAR done signals
32:63	HSIOM_IO_INPUT[0:31]	I/O inputs
64:71	TCPWM_32_TR_OUT1[0:7]	32-bit TCPWM0 counters
72:83	TCPWM_16M_TR_OUT1[0:11]	16-bit Motor enhanced TCPWM0 counters
: CAN TT sync triggers		
1:4	CAN0_TT_TR_OUT[0:3]	CAN0 TT sync outputs
5:8	CAN1_TT_TR_OUT[0:3]	CAN1 TT sync outputs
: DebugMain (Debug Multiplexer)		
1:5	TR_GROUP9_OUTPUT[0:4]	Output from debug reduction multiplexer #1
6:10	TR_GROUP10_OUTPUT[0:4]	Output from debug reduction multiplexer #2
: DebugReduction1 (Debug Reduction #1)		
1:92	PDMA0_TR_OUT[0:91]	P-DMA0 triggers
93:100	SCB_TX_TR_OUT[0:7]	SCB TTCAN tx triggers
101:108	SCB_RX_TR_OUT[0:7]	SCB TTCAN rx triggers
109:116	SCB_I2C_SCL_TR_OUT[0:7]	SCB I^2C triggers
117:120	CAN0_DBG_TR_OUT[0:3]	CAN0 P-DMA
121:124	CAN0_FIFO0_TR_OUT[0:3]	CAN0 FIFO0
125:128	CAN0_FIFO1_TR_OUT[0:3]	CAN0 FIFO1
129:132	CAN0_TT_TR_OUT[0:3]	CAN TT sync outputs
133:136	CAN1_DBG_TR_OUT[0:3]	CAN1 P-DMA
137:140	CAN1_FIFO0_TR_OUT[0:3]	CAN1 FIFO0
141:144	CAN1_FIFO1_TR_OUT[0:3]	CAN1 FIFO1
145:148	CAN1_TT_TR_OUT[0:3]	CAN TT sync outputs
149:150	CTI_TR_OUT[0:1]	Trace events
151:154	FAULT_TR_OUT[0:3]	Fault events
155:162	TCPWM_32_TR_OUT0[0:7]	32-bit TCPWM0 counters
163:174	TCPWM_16M_TR_OUT0[0:11]	16-bit Motor enhanced TCPWM0 counters
175:237	TCPWM_16_TR_OUT0[0:62]	16-bit TCPWM0 counters

(continued)

49:56	TCPWM_32_TR_OUT1[0:7]	32-bit TCPWM0 counters
57:68	TCPWM_16M_TR_OUT1[0:11]	16-bit Motor enhanced TCPWM0 counters
69:131	TCPWM_16_TR_OUT1[0:62]	16-bit TCPWM0 counters
132:137	PASS_GEN_TR_OUT[0:5]	PASS SAR conversion complete events
138:148	EVTGEN_TR_OUT[0:10]	EVTGEN Triggers
149:152	CXPI_TX_TR_OUT[0:3]	CXPI transmit events
153:156	CXPI_RX_TR_OUT[0:3]	CXPI receive events
157:188	HSIOM_IO_INPUT[0:31]	I/O inputs

	: PDMA0_TR (P-DMA0 trigger multiplexer)	
0:7	PDMA0_TR_IN[0:7]	Triggers to P-DMA0[0:7]
	: PDMA1_TR (P-DMA1 trigger multiplexer)	
0:7	PDMA1_TR_IN[0:7]	Triggers to P-DMA1[0:7]
	: MDMA (M-DMA0 trigger multiplexer)	
0:3	MDMA_TR_IN[0:3]	Triggers to M-DMA0
	: TCPWM_TO_PDMA0 (TCPWM0 to P-DMA0 trigger multiplexer)	
0:7	PDMA0_TR_IN[8:15]	Triggers to P-DMA0[8:15]
	: TCPWM_OUT (TCPWM0 loop back multiplexer)	
0:15	TCPWM_ALL_CNT_TR_IN[0:15]	All counters trigger input
	: TCPWM_IN (TCPWM0 Trigger Multiplexer)	
0:10	TCPWM_ALL_CNT_TR_IN[16:26]	Triggers to TCPWM0
	: PASS (PASS SAR trigger multiplexer)	
0:11	PASS_GEN_TR_IN[0:11]	Triggers to SAR ADCs
	: CAN TT sync triggers	
0:3	CAN0_TT_TR_IN[0:3]	CAN0 TT Sync Inputs
4:7	CAN1_TT_TR_IN[0:3]	CAN1 TT Sync Inputs
	: DebugMain (Debug Multiplexer)	
0:1	HSIOM_IO_OUTPUT[0:1]	To HSIOM as an output
2:3	CTI_TR_IN[0:1]	To CPU Cross Trigger system
4		

(continued)

17	PASS0_CH_RANGEVIO_TR_OUT[17]	TCPWM0_16_ONE_CNT_TR_IN[13]	SAR0 ch#17, range violation to TCPWM0 Group #0 Counter #13 trig=2
18	PASS0_CH_RANGEVIO_TR_OUT[18]	TCPWM0_16_ONE_CNT_TR_IN[14]	SAR0 ch#18, range violation to TCPWM0 Group #0 Counter #14 trig=2
19	PASS0_CH_RANGEVIO_TR_OUT[19]	TCPWM0_16_ONE_CNT_TR_IN[15]	SAR0 ch#19, range violation to TCPWM0 Group #0 Counter #15 trig=2
20	PASS0_CH_RANGEVIO_TR_OUT[20]	TCPWM0_16_ONE_CNT_TR_IN[16]	SAR0 ch#20, range violation to TCPWM0 Group #0 Counter #16 trig=2
21	PASS0_CH_RANGEVIO_TR_OUT[21]	TCPWM0_16_ONE_CNT_TR_IN[17]	SAR0 ch#21, range violation to TCPWM0 Group #0 Counter #17 trig=2
22	PASS0_CH_RANGEVIO_TR_OUT[22]	TCPWM0_16_ONE_CNT_TR_IN[18]	SAR0 ch#22, range violation to TCPWM0 Group #0 Counter #18 trig=2
23	PASS0_CH_RANGEVIO_TR_OUT[23]	TCPWM0_16_ONE_CNT_TR_IN[19]	SAR0 ch#23, range violation to TCPWM0 Group #0 Counter #19 trig=2
24	PASS0_CH_RANGEVIO_TR_OUT[32]	TCPWM0_16M_ONE_CNT_TR_IN[1]	SAR1 ch#0, range violation to TCPWM0 Group #1 Counter #01 trig=2
25	PASS0_CH_RANGEVIO_TR_OUT[33]	TCPWM0_16M_ONE_CNT_TR_IN[4]	SAR1 ch#1, range violation to TCPWM0 Group #1 Counter #04 trig=2
26	PASS0_CH_RANGEVIO_TR_OUT[34]	TCPWM0_16M_ONE_CNT_TR_IN[7]	SAR1 ch#2, range violation to TCPWM0 Group #1 Counter #07 trig=2
27	PASS0_CH_RANGEVIO_TR_OUT[35]	TCPWM0_16M_ONE_CNT_TR_IN[10]	SAR1 ch#3, range violation to TCPWM0 Group #1 Counter #10 trig=2
[redacted]			0 Group #0 Counter #20 trig=2
			0 Group #0 Counter #21 trig=2
			0 Group #0 Counter #22 trig=2
			0 Group #0 Counter #23 trig=2
			0 Group #0 Counter #24 trig=2
			0 Group #0 Counter #25 trig=2
			M0 Group #0 Counter #26 trig=2
			M0 Group #0 Counter #27 trig=2
			M0 Group #0 Counter #28 trig=2
			M0 Group #0 Counter #29 trig=2
			M0 Group #0 Counter #30 trig=2
			M0 Group #0 Counter #31 trig=2
			M0 Group #0 Counter #32 trig=2
			M0 Group #0 Counter #33 trig=2
			M0 Group #0 Counter #34 trig=2
			M0 Group #0 Counter #35 trig=2
			M0 Group #0 Counter #36 trig=2
			M0 Group #0 Counter #37 trig=2
			M0 Group #0 Counter #38 trig=2
			M0 Group #0 Counter #39 trig=2
			M0 Group #0 Counter #40 trig=2
			M0 Group #0 Counter #41 trig=2
			M0 Group #0 Counter #42 trig=2
			M0 Group #0 Counter #43 trig=2
			M0 Group #0 Counter #44 trig=2
			M0 Group #0 Counter #45 trig=2
			M0 Group #0 Counter #46 trig=2
			M0 Group #0 Counter #47 trig=2
			0 Group #1 Counter #02 trig=2
			0 Group #1 Counter #05 trig=2
			0 Group #1 Counter #08 trig=2
			0 Group #1 Counter #11 trig=2
			0 Group #0 Counter #48 trig=2
			0 Group #0 Counter #49 trig=2

(continued)

62	PASS0_CH_RANGEVIO_TR_OUT[70]	TCPWM0_16_ONE_CNT_TR_IN[50]	SAR2 ch#6, range violation to TCPWM0 Group #0 Counter #50 trig=2
63	PASS0_CH_RANGEVIO_TR_OUT[71]	TCPWM0_16_ONE_CNT_TR_IN[51]	SAR2 ch#7, range violation to TCPWM0 Group #0 Counter #51 trig=2
	: CAN0 to P-DMA0 Triggers		
0	CAN0_DBG_TR_OUT[0]	PDMA0_TR_IN[16]	CAN0, Channel #0 P-DMA0 trigger
1	CAN0_FIFO0_TR_OUT[0]	PDMA0_TR_IN[17]	CAN0, Channel #0 FIFO0 trigger
2	CAN0_FIFO1_TR_OUT[0]	PDMA0_TR_IN[18]	CAN0, Channel #0 FIFO1 trigger
3	CAN0_DBG_TR_OUT[1]	PDMA0_TR_IN[19]	CAN0, Channel #1 P-DMA0 trigger
4	CAN0_FIFO0_TR_OUT[1]	PDMA0_TR_IN[20]	CAN0, Channel #1 FIFO0 trigger
5	CAN0_FIFO1_TR_OUT[1]	PDMA0_TR_IN[21]	CAN0, Channel #1 FIFO1 trigger
6	CAN0_DBG_TR_OUT[2]	PDMA0_TR_IN[22]	CAN0, Channel #2 P-DMA0 trigger
7	CAN0_FIFO0_TR_OUT[2]	PDMA0_TR_IN[23]	CAN0, Channel #2 FIFO0 trigger
8	CAN0_FIFO1_TR_OUT[2]	PDMA0_TR_IN[24]	CAN0, Channel #2 FIFO1 trigger
9	CAN0_DBG_TR_OUT[3]	PDMA0_TR_IN[25]	CAN0, Channel #3 P-DMA0 trigger
10	CAN0_FIFO0_TR_OUT[3]	PDMA0_TR_IN[26]	CAN0, Channel #3 FIFO0 trigger
11	CAN0_FIFO1_TR_OUT[3]	PDMA0_TR_IN[27]	CAN0, Channel #3 FIFO1 trigger
	: CAN1 to P-DMA1 triggers		
0	CAN1_DBG_TR_OUT[0]	PDMA1_TR_IN[24]	CAN1, Channel #0 P-DMA01 trigger
1	CAN1_FIFO0_TR_OUT[0]	PDMA1_TR_IN[25]	CAN1, Channel #0 FIFO0 trigger
2	CAN1_FIFO1_TR_OUT[0]	PDMA1_TR_IN[26]	CAN1, Channel #0 FIFO1 trigger
3	CAN1_DBG_TR_OUT[1]	PDMA1_TR_IN[27]	CAN1, Channel #1 P-DMA1 trigger
4	CAN1_FIFO0_TR_OUT[1]	PDMA1_TR_IN[28]	CAN1, Channel #1 FIFO0 trigger
5	CAN1_FIFO1_TR_OUT[1]	PDMA1_TR_IN[29]	CAN1, Channel #1 FIFO1 trigger
6	CAN1_DBG_TR_OUT[2]	PDMA1_TR_IN[30]	CAN1, Channel #2 P-DMA1 trigger
7	CAN1_FIFO0_TR_OUT[2]	PDMA1_TR_IN[31]	CAN1, Channel #2 FIFO0 trigger
8	CAN1_FIFO1_TR_OUT[2]	PDMA1_TR_IN[32]	CAN1, Channel #2 FIFO1 trigger
9	CAN1_DBG_TR_OUT[3]	PDMA1_TR_IN[33]	CAN1, Channel #3 P-DMA1 trigger
10	CAN1_FIFO0_TR_OUT[3]	PDMA1_TR_IN[34]	CAN1, Channel #3 FIFO0 trigger
11	CAN1_FIFO1_TR_OUT[3]	PDMA1_TR_IN[35]	CAN1, Channel #3 FIFO1 trigger
	:Acknowledge triggers from P-DMA0 to CAN0		
0	PDMA0_TR_OUT[16]	CAN0_DBG_TR_ACK[0]	CAN0, Channel #0 P-DMA0 acknowledge
1	PDMA0_TR_OUT[19]	CAN0_DBG_TR_ACK[1]	CAN0, Channel #1 P-DMA0 acknowledge
2	PDMA0_TR_OUT[22]	CAN0_DBG_TR_ACK[2]	CAN0, Channel #2 P-DMA0 acknowledge
3	PDMA0_TR_OUT[25]	CAN0_DBG_TR_ACK[3]	CAN0, Channel #3 P-DMA0 acknowledge
	: Acknowledge triggers from P-DMA1 to CAN1		
0	PDMA1_TR_OUT[24]	CAN1_DBG_TR_ACK[0]	CAN1, Channel #0 P-DMA1 acknowledge
1	PDMA1_TR_OUT[27]	CAN1_DBG_TR_ACK[1]	CAN1, Channel #1 P-DMA1 acknowledge
2	PDMA1_TR_OUT[30]	CAN1_DBG_TR_ACK[2]	CAN1, Channel #2 P-DMA1 acknowledge
3	PDMA1_TR_OUT[33]	CAN1_DBG_TR_ACK[3]	CAN1, Channel #3 P-DMA1 acknowledge
	: SCBx to P-DMA1 Triggers		
0	SCB0_TX_TR_OUT	PDMA1_TR_IN[8]	SCB0 TX to P-DMA1 Trigger
1	SCB0_RX_TR_OUT	PDMA1_TR_IN[9]	SCB0 RX to P-DMA1 Trigger
2	SCB1_TX_TR_OUT	PDMA1_TR_IN[10]	SCB1 TX to P-DMA1 Trigger
3	SCB1_RX_TR_OUT	PDMA1_TR_IN[11]	SCB1 RX to P-DMA1 Trigger
4	SCB2_TX_TR_OUT	PDMA1_TR_IN[12]	SCB2 TX to P-DMA1 Trigger

(continued)

5	SCB2_RX_TR_OUT	PDMA1_TR_IN[13]	SCB2 RX to P-DMA1 Trigger
6	SCB3_TX_TR_OUT	PDMA1_TR_IN[14]	SCB3 TX to P-DMA1 Trigger
7	SCB3_RX_TR_OUT	PDMA1_TR_IN[15]	SCB3 RX to P-DMA1 Trigger
8	SCB4_TX_TR_OUT	PDMA1_TR_IN[16]	SCB4 TX to P-DMA1 Trigger
9	SCB4_RX_TR_OUT	PDMA1_TR_IN[17]	SCB4 RX to P-DMA1 Trigger
10	SCB5_TX_TR_OUT	PDMA1_TR_IN[18]	SCB5 TX to P-DMA1 Trigger
11	SCB5_RX_TR_OUT	PDMA1_TR_IN[19]	SCB5 RX to P-DMA1 Trigger
12	SCB6_TX_TR_OUT	PDMA1_TR_IN[20]	SCB6 TX to P-DMA1 Trigger
13	SCB6_RX_TR_OUT	PDMA1_TR_IN[21]	SCB6 RX to P-DMA1 Trigger
14	SCB7_TX_TR_OUT	PDMA1_TR_IN[22]	SCB7 TX to P-DMA1 Trigger
15	SCB7_RX_TR_OUT	PDMA1_TR_IN[23]	SCB7 RX to P-DMA1 Trigger
	: TCPWM0 to CXPI Triggers		
0:3	TCPWM0_16_TR_OUT0[16:19]	CXPI0_CMD_TR_IN[0:3]	TCPWM0 Group #0 (Counter #16 to #19) to CXPI0 Trigger
	: CXPI to P-DMA1 Triggers		
0:3	CXPI_TX_TR_OUT[0:3]	PDMA1_TR_IN[36:39]	CXPI0 TX to P-DMA1 Triggers
4:7	CXPI_RX_TR_OUT[0:3]	PDMA1_TR_IN[40:43]	CXPI0 RX to P-DMA1 Triggers

0	PCLK_CPUSS_CLOCK_TRACE_IN	Trace clock
1	PCLK_SMARTIO12_CLOCK	SMART I/O #12
2	PCLK_SMARTIO13_CLOCK	SMART I/O #13
3	PCLK_SMARTIO14_CLOCK	SMART I/O #14
4	PCLK_SMARTIO15_CLOCK	SMART I/O #15
5	PCLK_SMARTIO17_CLOCK	SMART I/O #17
6	PCLK_CANFD0_CLOCK_CAN0	CAN0, Channel #0
7	PCLK_CANFD0_CLOCK_CAN1	CAN0, Channel #1
8	PCLK_CANFD0_CLOCK_CAN2	CAN0, Channel #2
9	PCLK_CANFD0_CLOCK_CAN3	CAN0, Channel #3
10	PCLK_CANFD1_CLOCK_CAN0	CAN1, Channel #0
11	PCLK_CANFD1_CLOCK_CAN1	CAN1, Channel #1
12	PCLK_CANFD1_CLOCK_CAN2	CAN1, Channel #2
13	PCLK_CANFD1_CLOCK_CAN3	CAN1, Channel #3
14	PCLK_LIN0_CLOCK_CH_EN0	LIN0, Channel #0
15	PCLK_LIN0_CLOCK_CH_EN1	LIN0, Channel #1
16	PCLK_LIN0_CLOCK_CH_EN2	LIN0, Channel #2
17	PCLK_LIN0_CLOCK_CH_EN3	LIN0, Channel #3
18	PCLK_LIN0_CLOCK_CH_EN4	LIN0, Channel #4
19	PCLK_LIN0_CLOCK_CH_EN5	LIN0, Channel #5
20	PCLK_LIN0_CLOCK_CH_EN6	LIN0, Channel #6
21	PCLK_LIN0_CLOCK_CH_EN7	LIN0, Channel #7
22	PCLK_LIN0_CLOCK_CH_EN8	LIN0, Channel #8
23	PCLK_LIN0_CLOCK_CH_EN9	LIN0, Channel #9
24	PCLK_LIN0_CLOCK_CH_EN10	LIN0, Channel #10
25	PCLK_LIN0_CLOCK_CH_EN11	LIN0, Channel #11
26	PCLK_CXPI0_CLOCK_CH_EN0	CXPI0 Channel #0
27	PCLK_CXPI0_CLOCK_CH_EN1	CXPI0 Channel #1
28	PCLK_CXPI0_CLOCK_CH_EN2	CXPI0 Channel #2
29	PCLK_CXPI0_CLOCK_CH_EN3	CXPI0 Channel #3
30	PCLK_SCB0_CLOCK	SCB0
31	PCLK_SCB1_CLOCK	SCB1
32	PCLK_SCB2_CLOCK	SCB2
33	PCLK_SCB3_CLOCK	SCB3
34	PCLK_SCB4_CLOCK	SCB4
35	PCLK_SCB5_CLOCK	SCB5
36	PCLK_SCB6_CLOCK	SCB6
37	PCLK_SCB7_CLOCK	SCB7

(continued)

38	PCLK_PASS0_CLOCK_SAR0	SAR0
39	PCLK_PASS0_CLOCK_SAR1	SAR1
40	PCLK_PASS0_CLOCK_SAR2	SAR2
41	PCLK_TCPWM0_CLOCKS0	TCPWM0 Group #0, Counter #0
42	PCLK_TCPWM0_CLOCKS1	TCPWM0 Group #0, Counter #1
43	PCLK_TCPWM0_CLOCKS2	TCPWM0 Group #0, Counter #2
44	PCLK_TCPWM0_CLOCKS3	TCPWM0 Group #0, Counter #3
45	PCLK_TCPWM0_CLOCKS4	TCPWM0 Group #0, Counter #4
46	PCLK_TCPWM0_CLOCKS5	TCPWM0 Group #0, Counter #5
47	PCLK_TCPWM0_CLOCKS6	TCPWM0 Group #0, Counter #6
48	PCLK_TCPWM0_CLOCKS7	TCPWM0 Group #0, Counter #7
49	PCLK_TCPWM0_CLOCKS8	TCPWM0 Group #0, Counter #8
50	PCLK_TCPWM0_CLOCKS9	TCPWM0 Group #0, Counter #9
51	PCLK_TCPWM0_CLOCKS10	TCPWM0 Group #0, Counter #10
52	PCLK_TCPWM0_CLOCKS11	TCPWM0 Group #0, Counter #11
53	PCLK_TCPWM0_CLOCKS12	TCPWM0 Group #0, Counter #12
54	PCLK_TCPWM0_CLOCKS13	TCPWM0 Group #0, Counter #13
55	PCLK_TCPWM0_CLOCKS14	TCPWM0 Group #0, Counter #14
56	PCLK_TCPWM0_CLOCKS15	TCPWM0 Group #0, Counter #15
57	PCLK_TCPWM0_CLOCKS16	TCPWM0 Group #0, Counter #16
58	PCLK_TCPWM0_CLOCKS17	TCPWM0 Group #0, Counter #17
59	PCLK_TCPWM0_CLOCKS18	TCPWM0 Group #0, Counter #18
60	PCLK_TCPWM0_CLOCKS19	TCPWM0 Group #0, Counter #19
61	PCLK_TCPWM0_CLOCKS20	TCPWM0 Group #0, Counter #20
62	PCLK_TCPWM0_CLOCKS21	TCPWM0 Group #0, Counter #21
63	PCLK_TCPWM0_CLOCKS22	TCPWM0 Group #0, Counter #22
64	PCLK_TCPWM0_CLOCKS23	TCPWM0 Group #0, Counter #23
65	PCLK_TCPWM0_CLOCKS24	TCPWM0 Group #0, Counter #24
66	PCLK_TCPWM0_CLOCKS25	TCPWM0 Group #0, Counter #25
67	PCLK_TCPWM0_CLOCKS26	TCPWM0 Group #0, Counter #26
68	PCLK_TCPWM0_CLOCKS27	TCPWM0 Group #0, Counter #27
69	PCLK_TCPWM0_CLOCKS28	TCPWM0 Group #0, Counter #28
70	PCLK_TCPWM0_CLOCKS29	TCPWM0 Group #0, Counter #29
71	PCLK_TCPWM0_CLOCKS30	TCPWM0 Group #0, Counter #30
72	PCLK_TCPWM0_CLOCKS31	TCPWM0 Group #0, Counter #31
73	PCLK_TCPWM0_CLOCKS32	TCPWM0 Group #0, Counter #32
74	PCLK_TCPWM0_CLOCKS33	TCPWM0 Group #0, Counter #33
75	PCLK_TCPWM0_CLOCKS34	TCPWM0 Group #0, Counter #34
76	PCLK_TCPWM0_CLOCKS35	TCPWM0 Group #0, Counter #35
77	PCLK_TCPWM0_CLOCKS36	TCPWM0 Group #0, Counter #36

0	CPUSS_MPU_VIO_0	CM0+ SMPU violation DATA0[31:0]: Violating address. DATA1[0]: User read. DATA1[1]: User write. DATA1[2]: User execute. DATA1[3]: Privileged read. DATA1[4]: Privileged write. DATA1[5]: Privileged execute. DATA1[6]: Non-secure. DATA1[11:8]: Master identifier. DATA1[15:12]: Protection context identifier. DATA1[31]: '0' MPU violation; '1': SMPU violation.
1	CPUSS_MPU_VIO_1	CRYPTO SMPU violation. See CPUSS_MPU_VIO_0 description.
2	CPUSS_MPU_VIO_2	P-DMA0 MPU/SMPU violation. See CPUSS_MPU_VIO_0 description.
3	CPUSS_MPU_VIO_3	P-DMA1 MPU/SMPU violation. See CPUSS_MPU_VIO_0 description.
4	CPUSS_MPU_VIO_4	M-DMA0 MPU/SMPU violation. See CPUSS_MPU_VIO_0 description.
15	CPUSS_MPU_VIO_15	Test Controller MPU/SMPU violation. See CPUSS_MPU_VIO_0 description.
16	CPUSS_MPU_VIO_16	CM4 system bus AHB-Lite interface MPU violation. See CPUSS_MPU_VIO_0 description
17	CPUSS_MPU_VIO_17	CM4 code bus AHB-Lite interface MPU violation for non flash controller accesses. See CPUSS_MPU_VIO_0 description.
18	CPUSS_MPU_VIO_18	CM4 code bus AHB-Lite interface MPU violation for flash controller accesses. See CPUSS_MPU_VIO_0 description.
26	PERI_PERI_C_ECC	Peripheral protection SRAM correctable ECC violation DATA0[10:0]: Violating address. DATA1[7:0]: Syndrome of SRAM word.
27	PERI_PERI_NC_ECC	Peripheral protection SRAM non-correctable ECC violation
28	PERI_MS_VIO_0	CM0+ Peripheral Master Interface PPU violation DATA0[31:0]: Violating address. DATA1[0]: User read. DATA1[1]: User write. DATA1[2]: User execute. DATA1[3]: Privileged read. DATA1[4]: Privileged write. DATA1[5]: Privileged execute. DATA1[6]: Non-secure. DATA1[11:8]: Master identifier. DATA1[15:12]: Protection context identifier. DATA1[31:28]: “0”: master interface, PPU violation, “1': timeout detected, “2”: bus error, other: undefined.
29	PERI_MS_VIO_1	CM4 Peripheral Master Interface PPU violation. See PERI_MS_VIO_0 description.
30	PERI_MS_VIO_2	P-DMA0 Peripheral Master Interface PPU violation. See PERI_MS_VIO_0 description.
31	PERI_MS_VIO_3	P-DMA1 Peripheral Master Interface PPU violation. See PERI_MS_VIO_0 description.
32	PERI_GROUP_VIO_0	Peripheral Group #0 violation. DATA0[31:0]: Violating address. DATA1[0]: User read. DATA1[1]: User write. DATA1[2]: User execute. DATA1[3]: Privileged read. DATA1[4]: Privileged write. DATA1[5]: Privileged execute. DATA1[6]: Non-secure. DATA1[11:8]: Master identifier. DATA1[15:12]: Protection context identifier. DATA1[31:28]: “0”: decoder or peripheral bus error, other: undefined.
33	PERI_GROUP_VIO_1	Peripheral Group #1 violation. See PERI_GROUP_VIO_0 description.
34	PERI_GROUP_VIO_2	Peripheral Group #2 violation. See PERI_GROUP_VIO_0 description.
35	PERI_GROUP_VIO_3	Peripheral Group #3 violation. See PERI_GROUP_VIO_0 description.
37	PERI_GROUP_VIO_5	Peripheral Group #5 violation. See PERI_GROUP_VIO_0 description.
38	PERI_GROUP_VIO_6	Peripheral Group #6 violation. See PERI_GROUP_VIO_0 description.
41	PERI_GROUP_VIO_9	Peripheral Group #9 violation. See PERI_GROUP_VIO_0 description.

48	CPUSS_FLASHC_MAIN_BUS_ERROR	Flash controller main flash bus error FAULT_DATA0[26:0]: Violating address. Append 5'b00010 as most significant bits to derive 32-bit system address. FAULT_DATA1[11:8]: Master identifier.
49	CPUSS_FLASHC_MAIN_C_ECC	Flash controller main flash correctable ECC violation DATA[26:0]: Violating address. Append 5'b00010 as most significant bits to derive 32-bit system address. DATA1[7:0]: Syndrome of 64-bit word (at address offset 0x00). DATA1[15:8]: Syndrome of 64-bit word (at address offset 0x08).

(continued)

80	CANFD_0_CAN_C_ECC	CAN0 message buffer correctable ECC violation: DATA0[15:0]: Violating address. DATA0[22:16]: ECC violating data[38:32] from MRAM. DATA0[27:24]: Master ID: 0-7 = CAN channel ID within mxttcanfd cluster, 8 = AHB I/F DATA1[31:0]: ECC violating data[31:0] from MRAM.
81	CANFD_0_CAN_NC_ECC	CAN0 message buffer non-correctable ECC violation: DATA0[15:0]: Violating address. DATA0[22:16]: ECC violating data[38:32] from MRAM (not for Address Error). DATA0[27:24]: Master ID: 0-7 = CAN channel ID within mxttcanfd cluster, 8 = AHB I/F DATA0[30]: Write access, only possible for Address Error DATA0[31]: Address Error: a CAN channel did an MRAM access above MRAM_SIZE DATA1[31:0]: ECC violating data[31:0] from MRAM (not for Address Error).
82	CANFD_1_CAN_C_ECC	CAN1 message buffer correctable ECC violation. See CANFD_0_CAN_C_ECC description.
83	CANFD_1_CAN_NC_ECC	CAN1 message buffer non-correctable ECC violation. See CANFD_0_CAN_NC_ECC description.
90	SRSS_FAULT_CSV	Consolidated fault output for clock supervisors. Multiple CSV can detect a violation at the same time. DATA0[15:0]: CLK_HF* root CSV violation flags. DATA0[24]: CLK_REF CSV violation flag (reference clock for CLK_HF CSVs) DATA0[25]: CLK_LF CSV violation flag DATA0[26]: CLK_HVILO CSV violation flag
91	SRSS_FAULT_SSV	Consolidated fault output for supply supervisors. Multiple CSV can detect a violation at the same time. DATA0[0]: BOD on VDDA DATA[1]: OVD on VDDA DATA[16]: LVD/HVD #1 DATA0[17]: LVD/HVD #2
92	SRSS_FAULT_MCWDT0	Fault output for MCWDT0 (all sub-counters) Multiple counters can detect a violation at the same time. DATA0[0]: MCWDT sub counter 0 LOWER_LIMIT DATA0[1]: MCWDT sub counter 0 UPPER_LIMIT DATA0[2]: MCWDT sub counter 1 LOWER_LIMIT DATA0[3]: MCWDT sub counter 1 UPPER_LIMIT
93	SRSS_FAULT_MCWDT1	Fault output for MCWDT1 (all sub-counters). See SRSS_FAULT_MCWDT0 description.

Protection pair is a pair PPU structures, a master and a slave structure. The master structure protects the slave structure, and the slave structure protects resources such as peripheral registers, or the peripheral itself.

0	PERI_MS_PPU_FX_PERI_MAIN	0x40000000	0x00002000	Peripheral interconnect main
1	PERI_MS_PPU_FX_PERI_SECURE	0x40002000	0x00000004	Peripheral interconnect secure
2	PERI_MS_PPU_FX_PERI_GR0_GROUP	0x40004010	0x00000004	Peripheral group #0 main
3	PERI_MS_PPU_FX_PERI_GR1_GROUP	0x40004030	0x00000004	Peripheral group #1 main
4	PERI_MS_PPU_FX_PERI_GR2_GROUP	0x40004050	0x00000004	Peripheral group #2 main
5	PERI_MS_PPU_FX_PERI_GR3_GROUP	0x40004060	0x00000020	Peripheral group #3 main
6	PERI_MS_PPU_FX_PERI_GR5_GROUP	0x400040A0	0x00000020	Peripheral group #5 main
7	PERI_MS_PPU_FX_PERI_GR6_GROUP	0x400040C0	0x00000020	Peripheral group #6 main
8	PERI_MS_PPU_FX_PERI_GR9_GROUP	0x40004120	0x00000020	Peripheral group #9 main
9	PERI_MS_PPU_FX_PERI_TR	0x40008000	0x00008000	Peripheral trigger multiplexer
10	PERI_MS_PPU_FX_CRYPTO_MAIN	0x40100000	0x00000400	Crypto main
11	PERI_MS_PPU_FX_CRYPTO_CRYPTO	0x40101000	0x00000800	Crypto MMIO (memory mapped I/O)
12	PERI_MS_PPU_FX_CRYPTO_BOOT	0x40102000	0x00000100	Crypto boot
13	PERI_MS_PPU_FX_CRYPTO_KEY0	0x40102100	0x00000004	Crypto Key #0
14	PERI_MS_PPU_FX_CRYPTO_KEY1	0x40102120	0x00000004	Crypto Key #1
15	PERI_MS_PPU_FX_CRYPTO_BUF	0x40108000	0x00002000	Crypto buffer
16	PERI_MS_PPU_FX_CPUSS_CM4	0x40200000	0x00000400	CM4 CPU core
17	PERI_MS_PPU_FX_CPUSS_CM0	0x40201000	0x00001000	CM0+ CPU core
18	PERI_MS_PPU_FX_CPUSS_BOOT[34]	0x40202000	0x00000200	CPUSS boot
19	PERI_MS_PPU_FX_CPUSS_CM0_INT	0x40208000	0x00000800	CPUSS CM0+ interrupts
20	PERI_MS_PPU_FX_CPUSS_CM4_INT	0x4020A000	0x00000800	CPUSS CM4 interrupts
21	PERI_MS_PPU_FX_FAULT_STRUCT0_MAIN	0x40210000	0x00000100	CPUSS fault structure #0 main
22	PERI_MS_PPU_FX_FAULT_STRUCT1_MAIN	0x40210100	0x00000100	CPUSS fault structure #1 main
23	PERI_MS_PPU_FX_FAULT_STRUCT2_MAIN	0x40210200	0x00000100	CPUSS fault structure #2 main
24	PERI_MS_PPU_FX_FAULT_STRUCT3_MAIN	0x40210300	0x00000100	CPUSS fault structure #3 main
25	PERI_MS_PPU_FX_IPC_STRUCT0_IPC	0x40220000	0x00000020	CPUSS IPC structure #0
26	PERI_MS_PPU_FX_IPC_STRUCT1_IPC	0x40220020	0x00000020	CPUSS IPC structure #1
27	PERI_MS_PPU_FX_IPC_STRUCT2_IPC	0x40220040	0x00000020	CPUSS IPC structure #2
28	PERI_MS_PPU_FX_IPC_STRUCT3_IPC	0x40220060	0x00000020	CPUSS IPC structure #3
29	PERI_MS_PPU_FX_IPC_STRUCT4_IPC	0x40220080	0x00000020	CPUSS IPC structure #4
30	PERI_MS_PPU_FX_IPC_STRUCT5_IPC	0x402200A0	0x00000020	CPUSS IPC structure #5
31	PERI_MS_PPU_FX_IPC_STRUCT6_IPC	0x402200C0	0x00000020	CPUSS IPC structure #6
32	PERI_MS_PPU_FX_IPC_STRUCT7_IPC	0x402200E0	0x00000020	CPUSS IPC structure #7
33	PERI_MS_PPU_FX_IPC_INTR_STRUCT0_INTR	0x40221000	0x00000010	CPUSS IPC interrupt structure #0
34	PERI_MS_PPU_FX_IPC_INTR_STRUCT1_INTR	0x40221020	0x00000010	CPUSS IPC interrupt structure #1
35	PERI_MS_PPU_FX_IPC_INTR_STRUCT2_INTR	0x40221040	0x00000010	CPUSS IPC interrupt structure #2
36	PERI_MS_PPU_FX_IPC_INTR_STRUCT3_INTR	0x40221060	0x00000010	CPUSS IPC interrupt structure #3
37	PERI_MS_PPU_FX_IPC_INTR_STRUCT4_INTR	0x40221080	0x00000010	CPUSS IPC interrupt structure #4
38	PERI_MS_PPU_FX_IPC_INTR_STRUCT5_INTR	0x402210A0	0x00000010	CPUSS IPC interrupt structure #5

34.Fixed PPU is configured inside the Boot and user is not allowed to change the attributes of this PPU.

39	PERI_MS_PPU_FX_IPC_INTR_STRUCT6_INTR	0x402210C0	0x00000010	CPUSS IPC Interrupt Structure #6
40	PERI_MS_PPU_FX_IPC_INTR_STRUCT7_INTR	0x402210E0	0x00000010	CPUSS IPC Interrupt Structure #7
41	PERI_MS_PPU_FX_PROT_SMPU_MAIN	0x40230000	0x00000040	Peripheral protection SMPU main
42	PERI_MS_PPU_FX_PROT_MPU0_MAIN	0x40234000	0x00000004	Peripheral protection MPU #0 main
43	PERI_MS_PPU_FX_PROT_MPU14_MAIN	0x40237800	0x00000004	Peripheral protection MPU #14 main
44	PERI_MS_PPU_FX_PROT_MPU15_MAIN	0x40237C00	0x00000400	Peripheral protection MPU #15 main
45	PERI_MS_PPU_FX_FLASHC_MAIN	0x40240000	0x00000008	Flash controller main
46	PERI_MS_PPU_FX_FLASHC_CMD	0x40240008	0x00000004	Flash controller command
47	PERI_MS_PPU_FX_FLASHC_DFT	0x40240200	0x00000100	Flash controller tests
48	PERI_MS_PPU_FX_FLASHC_CM0	0x40240400	0x00000080	Flash controller CM0+
49	PERI_MS_PPU_FX_FLASHC_CM4	0x40240480	0x00000080	Flash controller CM4
50	PERI_MS_PPU_FX_FLASHC_CRYPTO	0x40240500	0x00000004	Flash controller Crypto
51	PERI_MS_PPU_FX_FLASHC_DW0	0x40240580	0x00000004	Flash controller P-DMA0
52	PERI_MS_PPU_FX_FLASHC_DW1	0x40240600	0x00000004	Flash controller P-DMA1
53	PERI_MS_PPU_FX_FLASHC_DMAC	0x40240680	0x00000004	Flash controller M-DMA

(continued)

129	PERI_MS_PPU_FX_DW0_CH_STRUCT58_CH	0x40288E80	0x00000040	P-DMA0 Channel #58
130	PERI_MS_PPU_FX_DW0_CH_STRUCT59_CH	0x40288EC0	0x00000040	P-DMA0 Channel #59
131	PERI_MS_PPU_FX_DW0_CH_STRUCT60_CH	0x40288F00	0x00000040	P-DMA0 Channel #60
132	PERI_MS_PPU_FX_DW0_CH_STRUCT61_CH	0x40288F40	0x00000040	P-DMA0 Channel #61
133	PERI_MS_PPU_FX_DW0_CH_STRUCT62_CH	0x40288F80	0x00000040	P-DMA0 Channel #62
134	PERI_MS_PPU_FX_DW0_CH_STRUCT63_CH	0x40288FC0	0x00000040	P-DMA0 Channel #63
135	PERI_MS_PPU_FX_DW0_CH_STRUCT64_CH	0x40289000	0x00000040	P-DMA0 Channel #64
136	PERI_MS_PPU_FX_DW0_CH_STRUCT65_CH	0x40289040	0x00000040	P-DMA0 Channel #65
137	PERI_MS_PPU_FX_DW0_CH_STRUCT66_CH	0x40289080	0x00000040	P-DMA0 Channel #66
138	PERI_MS_PPU_FX_DW0_CH_STRUCT67_CH	0x402890C0	0x00000040	P-DMA0 Channel #67
139	PERI_MS_PPU_FX_DW0_CH_STRUCT68_CH	0x40289100	0x00000040	P-DMA0 Channel #68
140	PERI_MS_PPU_FX_DW0_CH_STRUCT69_CH	0x40289140	0x00000040	P-DMA0 Channel #69
141	PERI_MS_PPU_FX_DW0_CH_STRUCT70_CH	0x40289180	0x00000040	P-DMA0 Channel #70
142	PERI_MS_PPU_FX_DW0_CH_STRUCT71_CH	0x402891C0	0x00000040	P-DMA0 Channel #71
143	PERI_MS_PPU_FX_DW0_CH_STRUCT72_CH	0x40289200	0x00000040	P-DMA0 Channel #72
144	PERI_MS_PPU_FX_DW0_CH_STRUCT73_CH	0x40289240	0x00000040	P-DMA0 Channel #73
145	PERI_MS_PPU_FX_DW0_CH_STRUCT74_CH	0x40289280	0x00000040	P-DMA0 Channel #74
146	PERI_MS_PPU_FX_DW0_CH_STRUCT75_CH	0x402892C0	0x00000040	P-DMA0 Channel #75
147	PERI_MS_PPU_FX_DW0_CH_STRUCT76_CH	0x40289300	0x00000040	P-DMA0 Channel #76
148	PERI_MS_PPU_FX_DW0_CH_STRUCT77_CH	0x40289340	0x00000040	P-DMA0 Channel #77
149	PERI_MS_PPU_FX_DW0_CH_STRUCT78_CH	0x40289380	0x00000040	P-DMA0 Channel #78
150	PERI_MS_PPU_FX_DW0_CH_STRUCT79_CH	0x402893C0	0x00000040	P-DMA0 Channel #79
151	PERI_MS_PPU_FX_DW0_CH_STRUCT80_CH	0x40289400	0x00000040	P-DMA0 Channel #80
152	PERI_MS_PPU_FX_DW0_CH_STRUCT81_CH	0x40289440	0x00000040	P-DMA0 Channel #81
153	PERI_MS_PPU_FX_DW0_CH_STRUCT82_CH	0x40289480	0x00000040	P-DMA0 Channel #82
154	PERI_MS_PPU_FX_DW0_CH_STRUCT83_CH	0x402894C0	0x00000040	P-DMA0 Channel #83
155	PERI_MS_PPU_FX_DW0_CH_STRUCT84_CH	0x40289500	0x00000040	P-DMA0 Channel #84
156	PERI_MS_PPU_FX_DW0_CH_STRUCT85_CH	0x40289540	0x00000040	P-DMA0 Channel #85
157	PERI_MS_PPU_FX_DW0_CH_STRUCT86_CH	0x40289580	0x00000040	P-DMA0 Channel #86
158	PERI_MS_PPU_FX_DW0_CH_STRUCT87_CH	0x402895C0	0x00000040	P-DMA0 Channel #87
159	PERI_MS_PPU_FX_DW0_CH_STRUCT88_CH	0x40289600	0x00000040	P-DMA0 Channel #88
160	PERI_MS_PPU_FX_DW0_CH_STRUCT89_CH	0x40289640	0x00000040	P-DMA0 Channel #89
161	PERI_MS_PPU_FX_DW0_CH_STRUCT90_CH	0x40289680	0x00000040	P-DMA0 Channel #90
162	PERI_MS_PPU_FX_DW0_CH_STRUCT91_CH	0x402896C0	0x00000040	P-DMA0 Channel #91
163	PERI_MS_PPU_FX_DW1_CH_STRUCT0_CH	0x40298000	0x00000040	P-DMA1 Channel #0
164	PERI_MS_PPU_FX_DW1_CH_STRUCT1_CH	0x40298040	0x00000040	P-DMA1 Channel #1
165	PERI_MS_PPU_FX_DW1_CH_STRUCT2_CH	0x40298080	0x00000040	P-DMA1 Channel #2
166	PERI_MS_PPU_FX_DW1_CH_STRUCT3_CH	0x402980C0	0x00000040	P-DMA1 Channel #3
167	PERI_MS_PPU_FX_DW1_CH_STRUCT4_CH	0x40298100	0x00000040	P-DMA1 Channel #4
168	PERI_MS_PPU_FX_DW1_CH_STRUCT5_CH	0x40298140	0x00000040	P-DMA1 Channel #5
169	PERI_MS_PPU_FX_DW1_CH_STRUCT6_CH	0x40298180	0x00000040	P-DMA1 Channel #6
170	PERI_MS_PPU_FX_DW1_CH_STRUCT7_CH	0x402981C0	0x00000040	P-DMA1 Channel #7
171	PERI_MS_PPU_FX_DW1_CH_STRUCT8_CH	0x40298200	0x00000040	P-DMA1 Channel #8
172	PERI_MS_PPU_FX_DW1_CH_STRUCT9_CH	0x40298240	0x00000040	P-DMA1 Channel #9
173	PERI_MS_PPU_FX_DW1_CH_STRUCT10_CH	0x40298280	0x00000040	P-DMA1 Channel #10

(continued)

174	PERI_MS_PPU_FX_DW1_CH_STRUCT11_CH	0x402982C0	0x00000040	P-DMA1 Channel #11
175	PERI_MS_PPU_FX_DW1_CH_STRUCT12_CH	0x40298300	0x00000040	P-DMA1 Channel #12
176	PERI_MS_PPU_FX_DW1_CH_STRUCT13_CH	0x40298340	0x00000040	P-DMA1 Channel #13
177	PERI_MS_PPU_FX_DW1_CH_STRUCT14_CH	0x40298380	0x00000040	P-DMA1 Channel #14
178	PERI_MS_PPU_FX_DW1_CH_STRUCT15_CH	0x402983C0	0x00000040	P-DMA1 Channel #15
179	PERI_MS_PPU_FX_DW1_CH_STRUCT16_CH	0x40298400	0x00000040	P-DMA1 Channel #16
180	PERI_MS_PPU_FX_DW1_CH_STRUCT17_CH	0x40298440	0x00000040	P-DMA1 Channel #17
181	PERI_MS_PPU_FX_DW1_CH_STRUCT18_CH	0x40298480	0x00000040	P-DMA1 Channel #18
182	PERI_MS_PPU_FX_DW1_CH_STRUCT19_CH	0x402984C0	0x00000040	P-DMA1 Channel #19
183	PERI_MS_PPU_FX_DW1_CH_STRUCT20_CH	0x40298500	0x00000040	P-DMA1 Channel #20
184	PERI_MS_PPU_FX_DW1_CH_STRUCT21_CH	0x40298540	0x00000040	P-DMA1 Channel #21
185	PERI_MS_PPU_FX_DW1_CH_STRUCT22_CH	0x40298580	0x00000040	P-DMA1 Channel #22
186	PERI_MS_PPU_FX_DW1_CH_STRUCT23_CH	0x402985C0	0x00000040	P-DMA1 Channel #23
187	PERI_MS_PPU_FX_DW1_CH_STRUCT24_CH	0x40298600	0x00000040	P-DMA1 Channel #24
188	PERI_MS_PPU_FX_DW1_CH_STRUCT25_CH	0x40298640	0x00000040	P-DMA1 Channel #25
189	PERI_MS_PPU_FX_DW1_CH_STRUCT26_CH	0x40298680	0x00000040	P-DMA1 Channel #26
190	PERI_MS_PPU_FX_DW1_CH_STRUCT27_CH	0x402986C0	0x00000040	P-DMA1 Channel #27
191	PERI_MS_PPU_FX_DW1_CH_STRUCT28_CH	0x40298700	0x00000040	P-DMA1 Channel #28
192	PERI_MS_PPU_FX_DW1_CH_STRUCT29_CH	0x40298740	0x00000040	P-DMA1 Channel #29
193	PERI_MS_PPU_FX_DW1_CH_STRUCT30_CH	0x40298780	0x00000040	P-DMA1 Channel #30
194	PERI_MS_PPU_FX_DW1_CH_STRUCT31_CH	0x402987C0	0x00000040	P-DMA1 Channel #31
195	PERI_MS_PPU_FX_DW1_CH_STRUCT32_CH	0x40298800	0x00000040	P-DMA1 Channel #32
196	PERI_MS_PPU_FX_DW1_CH_STRUCT33_CH	0x40298840	0x00000040	P-DMA1 Channel #33
197	PERI_MS_PPU_FX_DW1_CH_STRUCT34_CH	0x40298880	0x00000040	P-DMA1 Channel #34
198	PERI_MS_PPU_FX_DW1_CH_STRUCT35_CH	0x402988C0	0x00000040	P-DMA1 Channel #35
199	PERI_MS_PPU_FX_DW1_CH_STRUCT36_CH	0x40298900	0x00000040	P-DMA1 Channel #36
200	PERI_MS_PPU_FX_DW1_CH_STRUCT37_CH	0x40298940	0x00000040	P-DMA1 Channel #37
201	PERI_MS_PPU_FX_DW1_CH_STRUCT38_CH	0x40298980	0x00000040	P-DMA1 Channel #38
202	PERI_MS_PPU_FX_DW1_CH_STRUCT39_CH	0x402989C0	0x00000040	P-DMA1 Channel #39
203	PERI_MS_PPU_FX_DW1_CH_STRUCT40_CH	0x40298A00	0x00000040	P-DMA1 Channel #40
204	PERI_MS_PPU_FX_DW1_CH_STRUCT41_CH	0x40298A40	0x00000040	P-DMA1 Channel #41
205	PERI_MS_PPU_FX_DW1_CH_STRUCT42_CH	0x40298A80	0x00000040	P-DMA1 Channel #42
206	PERI_MS_PPU_FX_DW1_CH_STRUCT43_CH	0x40298AC0	0x00000040	P-DMA1 Channel #43
207	PERI_MS_PPU_FX_DMAC_TOP	0x402A0000	0x00000010	M-DMA0 main
208	PERI_MS_PPU_FX_DMAC_CH0_CH	0x402A1000	0x00000100	M-DMA0 Channel #0
209	PERI_MS_PPU_FX_DMAC_CH1_CH	0x402A1100	0x00000100	M-DMA0 Channel #1
210	PERI_MS_PPU_FX_DMAC_CH2_CH	0x402A1200	0x00000100	M-DMA0 Channel #2
211	PERI_MS_PPU_FX_DMAC_CH3_CH	0x402A1300	0x00000100	M-DMA0 Channel #3
212	PERI_MS_PPU_FX_EFUSE_CTL	0x402C0000	0x00000200	EFUSE control
213	PERI_MS_PPU_FX_EFUSE_DATA	0x402C0800	0x00000200	EFUSE data
214	PERI_MS_PPU_FX_BIST	0x402F0000	0x00001000	Built-in self test
215	PERI_MS_PPU_FX_HSIOM_PRT0_PRT	0x40300000	0x00000008	HSIOm Port #0
216	PERI_MS_PPU_FX_HSIOM_PRT1_PRT	0x40300010	0x00000008	HSIOm Port #1
217	PERI_MS_PPU_FX_HSIOM_PRT2_PRT	0x40300020	0x00000008	HSIOm Port #2
218	PERI_MS_PPU_FX_HSIOM_PRT3_PRT	0x40300030	0x00000008	HSIOm Port #3

(continued)

309	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT12_CNT	0x40380600	0x00000080	TCPWM0 Group #0, Counter #12
310	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT13_CNT	0x40380680	0x00000080	TCPWM0 Group #0, Counter #13
311	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT14_CNT	0x40380700	0x00000080	TCPWM0 Group #0, Counter #14
312	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT15_CNT	0x40380780	0x00000080	TCPWM0 Group #0, Counter #15
313	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT16_CNT	0x40380800	0x00000080	TCPWM0 Group #0, Counter #16
314	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT17_CNT	0x40380880	0x00000080	TCPWM0 Group #0, Counter #17
315	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT18_CNT	0x40380900	0x00000080	TCPWM0 Group #0, Counter #18
316	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT19_CNT	0x40380980	0x00000080	TCPWM0 Group #0, Counter #19
317	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT20_CNT	0x40380A00	0x00000080	TCPWM0 Group #0, Counter #20
318	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT21_CNT	0x40380A80	0x00000080	TCPWM0 Group #0, Counter #21
319	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT22_CNT	0x40380B00	0x00000080	TCPWM0 Group #0, Counter #22
320	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT23_CNT	0x40380B80	0x00000080	TCPWM0 Group #0, Counter #23
321	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT24_CNT	0x40380C00	0x00000080	TCPWM0 Group #0, Counter #24
322	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT25_CNT	0x40380C80	0x00000080	TCPWM0 Group #0, Counter #25
323	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT26_CNT	0x40380D00	0x00000080	TCPWM0 Group #0, Counter #26
324	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT27_CNT	0x40380D80	0x00000080	TCPWM0 Group #0, Counter #27
325	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT28_CNT	0x40380E00	0x00000080	TCPWM0 Group #0, Counter #28
326	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT29_CNT	0x40380E80	0x00000080	TCPWM0 Group #0, Counter #29
327	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT30_CNT	0x40380F00	0x00000080	TCPWM0 Group #0, Counter #30
328	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT31_CNT	0x40380F80	0x00000080	TCPWM0 Group #0, Counter #31
329	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT32_CNT	0x40381000	0x00000080	TCPWM0 Group #0, Counter #32
330	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT33_CNT	0x40381080	0x00000080	TCPWM0 Group #0, Counter #33
331	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT34_CNT	0x40381100	0x00000080	TCPWM0 Group #0, Counter #34
332	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT35_CNT	0x40381180	0x00000080	TCPWM0 Group #0, Counter #35
333	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT36_CNT	0x40381200	0x00000080	TCPWM0 Group #0, Counter #36
334	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT37_CNT	0x40381280	0x00000080	TCPWM0 Group #0, Counter #37
335	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT38_CNT	0x40381300	0x00000080	TCPWM0 Group #0, Counter #38
336	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT39_CNT	0x40381380	0x00000080	TCPWM0 Group #0, Counter #39
337	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT40_CNT	0x40381400	0x00000080	TCPWM0 Group #0, Counter #40
338	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT41_CNT	0x40381480	0x00000080	TCPWM0 Group #0, Counter #41
339	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT42_CNT	0x40381500	0x00000080	TCPWM0 Group #0, Counter #42
340	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT43_CNT	0x40381580	0x00000080	TCPWM0 Group #0, Counter #43
341	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT44_CNT	0x40381600	0x00000080	TCPWM0 Group #0, Counter #44
342	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT45_CNT	0x40381680	0x00000080	TCPWM0 Group #0, Counter #45
343	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT46_CNT	0x40381700	0x00000080	TCPWM0 Group #0, Counter #46
344	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT47_CNT	0x40381780	0x00000080	TCPWM0 Group #0, Counter #47
345	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT48_CNT	0x40381800	0x00000080	TCPWM0 Group #0, Counter #48
346	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT49_CNT	0x40381880	0x00000080	TCPWM0 Group #0, Counter #49
347	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT50_CNT	0x40381900	0x00000080	TCPWM0 Group #0, Counter #50
348	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT51_CNT	0x40381980	0x00000080	TCPWM0 Group #0, Counter #51
349	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT52_CNT	0x40381A00	0x00000080	TCPWM0 Group #0, Counter #52
350	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT53_CNT	0x40381A80	0x00000080	TCPWM0 Group #0, Counter #53
351	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT54_CNT	0x40381B00	0x00000080	TCPWM0 Group #0, Counter #54
352	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT55_CNT	0x40381B80	0x00000080	TCPWM0 Group #0, Counter #55
353	PERI_MS_PPU_FX_TCPWM0_GRP0_CNT56_CNT	0x40381C00	0x00000080	TCPWM0 Group #0, Counter #56

399	PERI_MS_PPU_FX_CANFD0_CH0_CH	0x40520000	0x00000200	CANFD0, Channel #0
400	PERI_MS_PPU_FX_CANFD0_CH1_CH	0x40520200	0x00000200	CANFD0, Channel #1
401	PERI_MS_PPU_FX_CANFD0_CH2_CH	0x40520400	0x00000200	CANFD0, Channel #2
402	PERI_MS_PPU_FX_CANFD0_CH3_CH	0x40520600	0x00000200	CANFD0, Channel #3
403	PERI_MS_PPU_FX_CANFD1_CH0_CH	0x40540000	0x00000200	CANFD1, Channel #0
404	PERI_MS_PPU_FX_CANFD1_CH1_CH	0x40540200	0x00000200	CANFD1, Channel #1
405	PERI_MS_PPU_FX_CANFD1_CH2_CH	0x40540400	0x00000200	CANFD1, Channel #2
406	PERI_MS_PPU_FX_CANFD1_CH3_CH	0x40540600	0x00000200	CANFD1, Channel #3
407	PERI_MS_PPU_FX_CANFD0_MAIN	0x40521000	0x00000100	CANFD0, main
408	PERI_MS_PPU_FX_CANFD1_MAIN	0x40541000	0x00000100	CANFD1, main
409	PERI_MS_PPU_FX_CANFD0_BUF	0x40530000	0x00010000	CANFD0, buffer
410	PERI_MS_PPU_FX_CANFD1_BUF	0x40550000	0x00010000	CANFD1, buffer
411	PERI_MS_PPU_FX_SCB0	0x40600000	0x00010000	SCB0
412	PERI_MS_PPU_FX_SCB1	0x40610000	0x00010000	SCB1
413	PERI_MS_PPU_FX_SCB2	0x40620000	0x00010000	SCB2
414	PERI_MS_PPU_FX_SCB3	0x40630000	0x00010000	SCB3
415	PERI_MS_PPU_FX_SCB4	0x40640000	0x00010000	SCB4
416	PERI_MS_PPU_FX_SCB5	0x40650000	0x00010000	SCB5
417	PERI_MS_PPU_FX_SCB6	0x40660000	0x00010000	SCB6
418	PERI_MS_PPU_FX_SCB7	0x40670000	0x00010000	SCB7

The Arbiter (part of flash controller) performs priority-based arbitration based on the master identifier. Each bus master has a dedicated 4-bit master identifier. This master identifier is used for bus arbitration and IPC functionality.

0	CPUSS_MS_ID_CM0	Master ID for Cortex®-M0+ CPU
1	CPUSS_MS_ID_CRYPTO	Master ID for Crypto
2	CPUSS_MS_ID_DW0	Master ID for P-DMA 0
3	CPUSS_MS_ID_DW1	Master ID for P-DMA 1
4	CPUSS_MS_ID_DMAC	Master ID for M-DMA0
14	CPUSS_MS_ID_CM4	Master ID for Cortex®-M4F CPU
15	CPUSS_MS_ID_TC	Master ID for DAP Tap controller

0	SRSS_NUM_CLKPATH	4	Number of clock paths. One for each of FLL, PLL, Direct and CSV
1			

Use of this device under conditions outside the Min and Max limits listed in may cause permanent damage to the device. Exposure to conditions within the limits of but beyond those of normal operation for extended periods of time may affect device reliability. The maximum storage temperature is 150 °C in compliance with JEDEC Standard JESD22-A103, High Temperature Storage Life. When operated under conditions within the limits of but beyond those of normal operation, the device may not operate to specification.

The average chip-junction temperature, T_J, in °C, may be calculated using Equation 1:

$$T_J \quad T_A \quad P_D \quad T\theta_{JA}\theta_{JA}P_D \qquad \text{Equation. 1}$$

Where:

T_A is the ambient temperature in °C.

θ_{JA} is the package junction-to-ambient thermal resistance, in °C/W.

P_D is the sum of P_{INT} and P_{IO} ($P_D = P_{INT} + P_{IO}$).

P_{INT} is the chip internal power. ($P_{INT} = V_{DDD} \times I_{DD} + V_{DDA} \times I_A$)

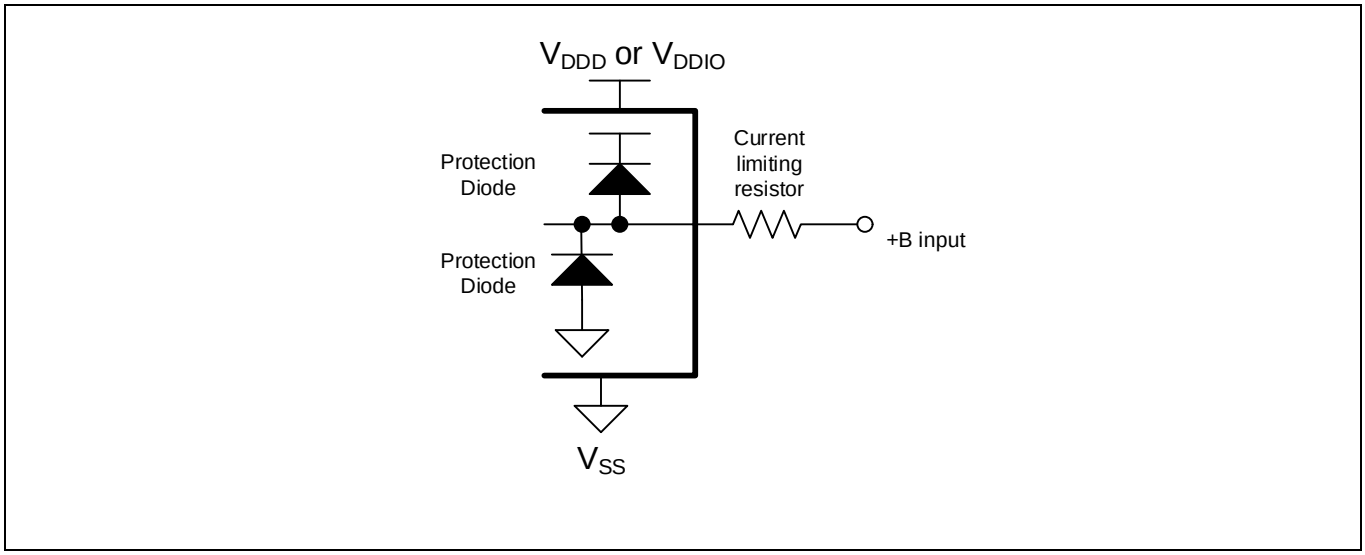
P_{IO} represents the power dissipation on input and output pins; user determined.

For most applications, $P_{IO} < P_{INT}$ and may be neglected.

On the other hand, P_{IO} may be significant if the device is configured to continuously drive external modules and/or memories.

SID10	V_{DDD_ABS}	V_{DDD} power supply voltage[35]	$V_{SSD} - 0.3$	–	$V_{SSD} + 6.0$	V	For ports 0, 1, 2, 3, 4, 5, 16, 17, 18, 19, 20, 21, 22, 23
SID10B	$V_{DDIO_1_ABS}$	V_{DDIO_1} power supply voltage[35]	$V_{SSD} - 0.3$	–	$V_{SSD} + 6.0$	V	For ports 6, 7, 8, 9[36]
SID10C1	$V_{DDIO_2_ABS}$	V_{DDIO_2} power supply voltage[35]	$V_{SSD} - 0.3$	–	$V_{SSD} + 6.0$	V	For ports 10, 11, 12, 13, 14, 15
SID11	V_{DDA_ABS}	V_{DDA} analog power supply voltage[35]	$V_{SSA} - 0.3$	–	$V_{SSA} + 6.0$	V	$V_{DDIO_2} = V_{DDA}$
SID12	V_{REFH_ABS}	Analog reference voltage, HIGH[35]	$V_{SSA} - 0.3$	–	$V_{SSA} + 6.0$	V	V_{REFH} $V_{DDA} + 0.3$ V
SID12A	V_{REFL_ABS}	Analog reference voltage, LOW[35]	$V_{SSA} - 0.3$	–	$V_{SSA} + 0.3$	V	
SID15A	V_{I0_ABS0}	Input voltage[35]	$V_{SSD} - 0.5$	–	$V_{DDD} + 0.5$	V	For ports 0, 1, 2, 3, 4, 5, 16, 17, 18, 19, 20, 21, 22, 23
SID15B	V_{I1_ABS1}	Input voltage[35]	$V_{SSD} - 0.5$	–	$V_{DDIO_1} + 0.5$	V	For ports 6, 7, 8, 9[36]
SID15C	V_{I2_ABS2}	Input voltage[35]	$V_{SSD} - 0.5$	–	$V_{DDIO_2} + 0.5$	V	For ports 10, 11, 12, 13, 14, 15
SID16	V_{IA_ABS}	Analog input voltage[35]	$V_{SSA} - 0.3$	–	$V_{DDA} + 0.3$	V	
SID17A	V_{O0_ABS0}	Output voltage[35]	$V_{SSD} - 0.3$	–	$V_{DDD} + 0.3$	V	For ports 0, 1, 2, 3, 4, 5, 16, 17, 18, 19, 20, 21, 22, 23
SID17B	V_{O1_ABS1}	Output voltage[35]	$V_{SSD} - 0.3$	–	$V_{DDIO_1} + 0.3$	V	For ports 6, 7, 8, 9[36]
SID17C	V_{O2_ABS2}	Output voltage[35]	$V_{SSD} - 0.3$	–	$V_{DDIO_2} + 0.3$	V	For ports 10, 11, 12, 13, 14, 15
SID18	I_{CLAMP_ABS}	Maximum clamp current [37, 38, 39]	–5	–	5	mA	

SID18A	$I_{CLAMP_SUPPLY_POS_ABS}$	Maximum positive clamp current per I/O supply pin. Limit applies to I/O supply pin closest to the B+ injected current[39]	–	–	10	mA	+B injected DC currents are not allowed for Ports 11 and 21.
SID18B	$I_{CLAMP_SUPPLY_NEG_ABS}$	Maximum negative clamp current per I/O ground pin. Limit applies to I/O supply pin closest to the B+ injected current					



[43]

WARNING:
Semiconductor devices may be permanently damaged by application of stress (including, without limitation, voltage, current, or temperature) in excess of absolute maximum ratings. Do not exceed any of these ratings.

SID40	V_{DDD}, V_{DDA}, V_{DDIO_1}, V_{DDIO_2},	Power supply voltage[44]	2.7[45]	–	5.5[46]	V	
SID40A	$V_{DDIO_1_EFP}$	Power supply voltage for eFuse programming[47]	3	–	5.5	V	
SID41	C_{S1}	Smoothing capacitor[48, 49]	3.76	–	11	µF	

43. +B is the positive battery voltage around 45 V.
44. V_{DDD}, V_{DDIO_1}, V_{DDIO_2}, and V_{DDA} do not have any sequencing limitation and can establish in any order. These supplies (except for V_{DDA} and V_{DDIO_2}) are independent in voltage level. See 12-Bit SAR ADC DC Specifications when using ADC units.
45. 3.0 V ±10% is supported with a lower BOD setting option for V_{DDD} and V_{DDA}. This setting provides robust protection for internal timing but BOD reset occurs at a voltage below the specified operating conditions. A higher BOD setting option is available (consistent with down to 3.0 V) and guarantees that all operating conditions are met.

All specifications are valid for –40 °C T_A 125 °C and for 2.7 V to 5.5 V except where noted.

SID49C1	$I_{DD1_CM04_8_1}$	LP Active mode (CM4 and CM0+ at 8 MHz, all peripherals are disabled)	–	5	10	mA	CM0+ and CM4 clocked at 8 MHz with IMO. All peripherals are disabled. No I/O toggling. TYP: T_A = 25 °C, V_{DDD} = 5.0 V, process typ (TT), CM0+ and CM4 executing Dhrystone from flash with cache enabled MAX: T_A = 25 °C, V_{DDD} = 5.5 V, process worst (FF), CM0+ and CM4 executing Dhrystone from flash with cache enabled.
SID49C4	$I_{DD1_CM04_8_4M}$	LP Active mode (CM4 and CM0+ at 8 MHz, all peripherals are enabled)	–	6	73	mA	CM0+ and CM4 clocked at 8 MHz with IMO. All peripherals are enabled. No I/O toggling. M-DMA transferring data from code + work flash, P-DMA chains with maximum trigger activity. TYP: T_A = 25 °C, V_{DDD} = 5.0 V, process typ (TT), CM0+ and CM4 executing Dhrystone from flash with cache enabled MAX: T_A = 125 °C, V_{DDD} = 5.5 V, process worst (FF), CM0+ and CM4 executing max_power.c from flash with cache enabled.
SID49E4	$I_{DD1_F160_4M}$	Active mode (CM4 at 160 MHz, CM0+ at 80 MHz, all peripherals are enabled)	–	52	127	mA	PLL enabled at 160 MHz with ECO reference. All peripherals are enabled. No I/O toggling. M-DMA transferring data from code + work flash, P-DMA chains with maximum trigger activity. TYP: T_A = 25 °C, V_{DDD} = 5.0 V, process typ (TT), CM4 and CM0+ executing Dhrystone from flash with cache enabled. MAX: T_A = 125 °C, V_{DDD} = 5.5 V, process worst (FF), CM4 and CM0+ executing max_power.c from flash with cache enabled
SID53A4	$I_{DD2_8_4M}$	All CPUs in Sleep mode	–	4	68	mA	PLL disabled, CM4 and CM0+ are sleeping at 8 MHz with IMO. All peripherals, peripheral clocks, interrupts, CSV, DMA, FLL, ECO are disabled. No I/O toggling. Typ: T_A = 25 °C, V_{DDD} = 5.0 V, process typ (TT) Max: T_A = 125 °C, V_{DDD} = 5.5 V, process worst (FF)

(continued)

All specifications are valid for –40 °C ≤ T_A ≤ 125 °C and for 2.7 V to 5.5 V except where noted.

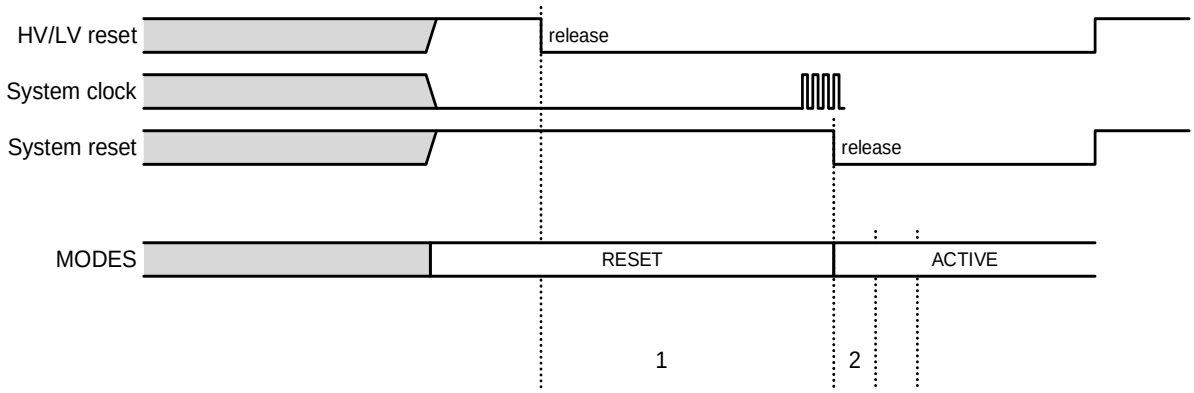
SID81B	t_{FB_A}	Flash boot with app authentication time in NORMAL/SECURE protection state	–	–	5000	µs	Guaranteed by Design, TOC2_FLAGS = 0x24F, CM0+ clocked at 100 MHz, Listen window = 0 ms, Public key exponent e = 0x010001, APP size is 64 KB with the last 256 bytes being a digital signature in RSASSA-PKCS1-v1.5. Valid for RSA-2048. (Flash boot version 3.1.0.556 and later)

SID600	V_{CCD}	Core supply voltage	1.05	1.1	1.15	V	
SID601	I_{DD_ACT}	Regulator operating current in Active/Sleep mode	–	80	150	µA	Guaranteed by design
SID602	I_{DD_DPSLP}	Regulator operating current in DeepSleep mode	–	1.5	20	µA	Guaranteed by design
SID604	I_{OUT}	Available regulator output current for operation	–	–	150	mA	Without triggering OVD
SID603	I_{RUSH}	In-rush current	–	–	375	mA	Average V_{DDD} current until C_{s1} (connected to V_{CCD} pin) is charged after Active regulator is turned on

50.At cold temperature -5 °C to - 40 °C, the DeepSleep to Active transition time can be higher than the max time indicated by as much as 20 µs

All specifications are valid for –40 °C ≤ T_A ≤ 125 °C and for 2.7 V to 5.5 V except where noted.

SID73	I_{DD_XRES}	I_{DD} when XRES_L asserted	–	–	0.9	mA	TA = 125 °C, V_{DDD} = 5.5 V, process worst (FF)
SID74	V_{IH}	Input voltage HIGH threshold	0.7 × V_{DDD}	–	–	V	CMOS input
SID75	V_{IL}	Input voltage LOW threshold	–	–	0.3 × V_{DDD}	V	CMOS input
SID76	R_{PULLUP}	Pull-up resistor	7	–	20	kΩ	
SID77	C_{IN}	Input capacitance	–	–	5	pF	
SID78	$V_{HYSXRES}$	Input voltage hysteresis	0.05 × V_{DDD}	–	–	V	
SID70	t_{XRES_ACT}	XRES_L release to Active transition time	–	–	265	µs	Without boot runtime. Guaranteed by design
SID71	t_{XRES_PW}	XRES_L pulse width	5	–	–	µs	
SID72	t_{XRES_FT}	Pulse suppression width	100	–	–	ns	



1: **SID68/68A/68B**: Time from HV/LV reset release until CM0+ begins executing ROM boot
2: **SID80A/80B**: ROM boot code operation
3: **SID81A/81B**: Flash boot code operation

All specifications are valid for –40 °C T_A 125 °C and for 2.7 V to 5.5 V except where noted.

SID650	$V_{OL1_GPIO_STD}$	Output voltage LOW level	–	–	0.6	V	I_{OL} = 6 mA drive_sel<1:0> = 0b0X, 4.5 V V_{DDD} or V_{DDIO_1} or V_{DDIO_2} 2168.2(P)33.3(a)-3(r)2

(continued)

SID657C	$V_{OH4C_GPIO_STD}$	Output voltage HIGH level	(V_{DDD} or V_{DDIO_1} or V_{DDIO_2}) – 0.5	–	–	V	I_{OH} = –1 mA drive_sel<1:0> = 0b11, 4.5 V V_{DDD} or V_{DDIO_1} or V_{DDIO_2} 5.5 V
SID658	$R_{PD_GPIO_STD}$	Pull-down resistance	25	50	100	kΩ	
SID659	$R_{PU_GPIO_STD}$	Pull-up resistance	25	50	100	kΩ	
SID660	$V_{IH_CMOS_GPIO_STD}$	Input voltage HIGH threshold in CMOS mode	0.7 × (V_{DDD} or V_{DDIO_1} or V_{DDIO_2})	–	–	V	
SID661	$V_{IH_TTL_GPIO_STD}$	Input voltage HIGH threshold in TTL mode	2.0	–	–	V	
SID662	$V_{IH_AUTO_GPIO_STD}$	Input voltage HIGH threshold in AUTO mode	0.8 × (V_{DDD} or V_{DDIO_1} or V_{DDIO_2})	–	–	V	
SID663	$V_{IL_CMOS_GPIO_STD}$	Input voltage LOW threshold in CMOS mode	–	–	0.3 × (V_{DDD} or V_{DDIO_1} or V_{DDIO_2})	V	
SID664	$V_{IL_TTL_GPIO_STD}$	Input voltage LOW threshold in TTL mode	–	–	0.8	V	
SID665	$V_{IL_AUTO_GPIO_STD}$	Input voltage LOW threshold in AUTO mode	–	–	0.5 × (V_{DDD} or V_{DDIO_1} or V_{DDIO_2})	V	
SID666	$V_{HYST_CMOS_GPIO_STD}$	Hysteresis in CMOS mode	0.05 × (V_{DDD} or V_{DDIO_1} or V_{DDIO_2})	–	–	V	
SID668	$V_{HYST_AUTO_GPIO_STD}$	Hysteresis in AUTO mode	0.05 × (V_{DDD} or V_{DDIO_1} or V_{DDIO_2})	–	–	V	
SID669	$C_{in_GPIO_STD}$	Input pin capacitance	–	–	5	pF	For 10 MHz and 100 MHz
SID670	$I_{IL_GPIO_STD}$	Input leakage current	–250	0.02	250	nA	For GPIO_STD except P21.0, P21.1, P21.2, P21.3, P21.4, P23.3, P23.4. $V_{DDIO_1} = V_{DDIO_2} = V_{DDD} = V_{DDA}$ = 5.5 V, $V_{SSD} < V_I < V_{DDD}$, V_{DDIO_1}, V_{DDIO_2} –40 °C TA 125 °C TYP: T_A = 25 °C, $V_{DDIO_1} = V_{DDIO_2} = V_{DDD} = V_{DDA}$ = 5.0 V

(continued)

SID670C	$I_{IL_GPIO_STD_B}$	Input leakage current	–700	0.02	700	nA	Only for P21.0, P21.1, P21.2, P21.3, P21.4, P23.3, P23.4. $V_{DDIO_1} = V_{DDIO_2} = V_{DDD} = V_{DDA} = 5.5$ V, $V_{SSD} < V_I < V_{DDD}$, V_{DDIO_1}, V_{DDIO_2} –40 °C TA 125 °C TYP: $T_A = 25$ °C, $V_{DDIO_1} = V_{DDIO_2} = V_{DDD} = V_{DDA} = 5.0$ V
SID671	t_R or t_F (fast)$_{20_0_GPIO_STD}$	Rise time or fall time (10% to 90% of V_{DDIO})	1	–	10	ns	20-pF load, drive_sel<1:0> = 0b00
SID672	t_R or t_F (fast)$_{50_0_GPIO_STD}$	Rise time or fall time (10% to 90% of V_{DDIO})	1	–	20	ns	50-pF load, drive_sel<1:0> = 0b00
SID673	t_R or t_F (fast)$_{20_1_GPIO_STD}$	Rise time or fall time (10% to 90% of V_{DDIO})	1	–	20	ns	20-pF load, drive_sel<1:0> = 0b01, guaranteed by design
SID674	t_R or t_F (fast)$_{10_2_GPIO_STD}$	Rise time or fall time (10% to 90% of V_{DDIO})	1	–	20	ns	10-pF load, drive_sel<1:0> = 0b10, guaranteed by design
SID675	t_R or t_F (fast)$_{6_3_GPIO_STD}$	Rise time or fall time (10% to 90% of V_{DDIO})	1	–	20	ns	6-pF load, drive_sel<1:0> = 0b11, guaranteed by design
SID676	t_F (fast)$_{100_GPIO_STD}$	Fall time (30% to 70% of V_{DDIO})	0.35	–	250	ns	10-pF to 400-pF load, RPU = 767 Ω, drive_sel<1:0>= 0b00, Freq = 100 kHz
SID677	t_F (fast)$_{400_GPIO_STD}$	Fall time (30% to 70% of V_{DDIO})	0.35	–	250	ns	10-pF to 400-pF load, RPU = 350 Ω, drive_sel<1:0>= 0b00, Freq = 400 kHz
SID678	$f_{IN_GPIO_STD}$	Input frequency	–	–	100	MHz	
SID679	$f_{OUT_GPIO_STD0H}$	Output frequency	–	–	50	MHz	20-pF load, drive_sel<1:0>= 00, 4.5 V V_{DDD} or V_{DDIO_1} or V_{DDIO_2} 5.5 V
SID680	$f_{OUT_GPIO_STD0L}$	Output frequency	–	–	32	MHz	20-pF load, drive_sel<1:0>= 00, 2.7 V V_{DDD} or V_{DDIO_1} or V_{DDIO_2} < 4.5 V
SID681	$f_{OUT_GPIO_STD1H}$	Output frequency	–	–	25	MHz	20-pF load, drive_sel<1:0>= 01, 4.5 V V_{DDD} or V_{DDIO_1} or V_{DDIO_2} 5.5 V

(continued)

SID682	$f_{OUT_GPIO_STD1L}$	Output frequency	–	–	15	MHz	20-pF load, drive_sel<1:0>= 01, 2.7 V ≤ V_{DDD} or V_{DDIO_1} or V_{DDIO_2} < 4.5 V
SID683	$f_{OUT_GPIO_STD2H}$	Output frequency	–	–	25	MHz	10-pF load, drive_sel<1:0>= 10, 4.5 V ≤ V_{DDD} or V_{DDIO_1} or V_{DDIO_2} ≤ 5.5 V
SID684	$f_{OUT_GPIO_STD2L}$	Output frequency	–	–	15	MHz	10-pF load, drive_sel<1:0>= 10, 2.7 V ≤ V_{DDD} or V_{DDIO_1} or V_{DDIO_2} < 4.5 V
SID685	$f_{OUT_GPIO_STD3H}$	Output frequency	–	–	15	MHz	6-pF load, drive_sel<1:0>= 11, 4.5 V ≤ V_{DDD} or V_{DDIO_1} or V_{DDIO_2} ≤ 5.5 V
SID686	$f_{OUT_GPIO_STD3L}$	Output frequency	–	–	10	MHz	6-pF load, drive_sel<1:0>= 11, 2.7 V ≤ V_{DDD} or V_{DDIO_1} or V_{DDIO_2} < 4.5 V
SID650A	$V_{OL1_GPIO_ENH}$	Output voltage LOW level	–	–	0.6	V	I_{OL} = 6 mA drive_sel<1:0> = 0b00/01, 2.7 V ≤ V_{DDD} ≤ 5.5 V
SID650D	$V_{OL1D_GPIO_ENH}$	Output voltage LOW level	–	–	0.4	V	I_{OL} = 5 mA drive_sel<1:0> = 0b00/01, 4.5 V ≤ V_{DDD} ≤ 5.5 V
SID651A	$V_{OL2_GPIO_ENH}$	Output voltage LOW level	–	–	0.4	V	I_{OL} = 2 mA drive_sel<1:0> = 0b00/01, 2.7 V ≤ V_{DDD} < 4.5 V
SID652A	$V_{OL3_GPIO_ENH}$	Output voltage LOW level	–	–	0.4	V	I_{OL} = 1 mA drive_sel<1:0> = 0b10, 2.7 V ≤ V_{DDD} < 4.5 V
SID652D	$V_{OL3D_GPIO_ENH}$	Output voltage LOW level	–	–	0.4	V	I_{OL} = 2 mA drive_sel<1:0> = 0b10, 4.5 V ≤ V_{DDD} ≤ 5.5 V
SID653A	$V_{OL4_GPIO_ENH}$	Output voltage LOW level	–	–	0.4	V	I_{OL} = 0.5 mA drive_sel<1:0> = 0b11, 2.7 V ≤ V_{DDD} < 4.5 V
SID653D	$V_{OL4D_GPIO_ENH}$	Output voltage LOW level	–	–	0.4	V	I_{OL} = 1 mA drive_sel<1:0> = 0b11, 4.5 V ≤ V_{DDD} ≤ 5.5 V
SID654A	$V_{OH1_GPIO_ENH}$	Output voltage HIGH level	V_{DDD} – 0.5	–	–	V	I_{OL} = –2 mA drive_sel<1:0> = 0b00/01, 2.7 V ≤ V_{DDD} < 4.5 V

SID655A	$V_{OH2_GPIO_ENH}$	Output voltage HIGH level	V_{DDD} – 0.5	–	–	V	I_{OL} = –5 mA drive_sel<1:0> = 0b00/01, 4.5 V ≤ V_{DDD} ≤ 5.5 V
SID656A	$V_{OH3_GPIO_ENH}$	Output voltage HIGH level	V_{DDD} – 0.5	–	–	V	I_{OL} = –1 mA drive_sel<1:0> = 0b10, 2.7 V ≤ V_{DDD} < 4.5 V
SID656D	$V_{OH3D_GPIO_ENH}$	Output voltage HIGH level	V_{DDD} – 0.5	–	–	V	I_{OL} = –2 mA drive_sel<1:0> = 0b10, 4.5 V ≤ V_{DDD} ≤ 5.5 V
SID657A	$V_{OH4_GPIO_ENH}$	Output voltage HIGH level	V_{DDD} – 0.5	–	–	V	I_{OL} = –0.5 mA drive_sel<1:0> = 0b11, 2.7 V ≤ V_{DDD} < 4.5 V
SID657D	$V_{OH4D_GPIO_ENH}$	Output voltage HIGH level	V_{DDD} – 0.5	–	–	V	I_{OL} = –1 mA drive_sel<1:0> = 0b11, 4.5 V ≤ V_{DDD}

SID671A	t_R or t_F (fast)$_{20_0_GPIO_ENH}$	Rise time or fall time (10% to 90% of V_{DDIO})	1	–	10	ns	20-pF load, drive_sel<1:0> = 0b00, slow = 0
SID672A	t_R or t_F (fast)$_{50_0_GPIO_ENH}$	Rise time or fall time (10% to 90% of V_{DDIO})	1	–	20	ns	50-pF load, drive_sel<1:0> = 0b00, slow = 0
SID673A	t_R or t_F (fast)$_{20_1_GPIO_ENH}$	Rise time or fall time (10% to 90% of V_{DDIO})	1	–	20	ns	20-pF load, drive_sel<1:0> = 0b01, slow = 0, guaranteed by design
SID674A	t_R or t_F (fast)$_{10_2_GPIO_ENH}$	Rise time or fall time (10% to 90% of V_{DDIO})	1	–	20	ns	10-pF load, drive_sel<1:0> = 0b10, slow = 0, guaranteed by design
SID675A	t_R or t_F (fast)$_{6_3_GPIO_ENH}$	Rise time or fall time (10% to 90% of V_{DDIO})	1	–	20	ns	6-pF load, drive_sel<1:0> = 0b11, slow = 0, guaranteed by design
SID676A	t_{F_I2C} (slow)$_{GPIO_ENH}$	Fall time (30% to 70% of V_{DDIO})	$20 \times (V_{DDD} / 5.5)$	–	250	ns	10-pF to 400-pF load, drive_sel<1:0> = 0b00, slow = 1, minimum R_{PU} = 400 Ω
SID677A	t_R or t_F (slow)$_{20_GPIO_ENH}$	Rise time or fall time (10% to 90% of V_{DDIO}					

(continued)

SID684A	$f_{OUT_GPIO_ENH2H}$	Output frequency	–	–	25	MHz	10-pF load, drive_sel<1:0>= 0b10, 4.5 V ≤ V_{DDD} ≤ 5.5 V
SID685A	$f_{OUT_GPIO_ENH2L}$	Output frequency	–	–	15	MHz	10-pF load, drive_sel<1:0>= 0b10, 2.7 V ≤ V_{DDD} < 4.5 V
SID686A	$f_{OUT_GPIO_ENH3H}$	Output frequency	–	–	15	MHz	6-pF load, drive_sel<1:0>= 0b11, 4.5 V ≤ V_{DDD} ≤ 5.5 V
SID687A	$f_{OUT_GPIO_ENH3L}$	Output frequency	–	–	10	MHz	6-pF load, drive_sel<1:0>= 0b11, 2.7 V ≤ V_{DDD} < 4.5 V
SID98	t_{FT}	Analog glitch filter (pulse suppression width)	–	–	50[51]	ns	One filter per port group
SID99	t_{INT}	Minimum pulse width for GPIO interrupt	160	–	–	ns	

51.If longer pulse suppression width is required, use Smart I/O.

All specifications are valid for –40 °C ≤ T_A ≤ 125 °C and for 2.7 V to 5.5 V except where noted.

SID100	A_RES	SAR ADC resolution	–	–	12	bits	
SID101	A_V_{INS}	Input voltage range	V_{REFL}	–	V_{REFH}	V	
SID102	A_V_{REFH}	V_{REFH} voltage range	2.7	–	V_{DDA}	V	ADC performance degrades when high reference is higher than supply
SID102A	A_V_{DDA}[52]	V_{DDA} voltage range	2.7	–	5.5	V	
SID103	A_V_{REFL}	V_{REFL} voltage range	V_{SSA}	–	V_{SSA}	V	ADC performance degrades when low reference is lower than ground
SID103A	V_{band_gap}	Internal band gap reference voltage	0.882	0.9	0.918	V	

The three ADC specifications based on SID19A, SID19B, and SID19C, can be used to calculate the pin leakage and resulting ADC offset caused by injection current using the below formula:

$I_{LEAK} = I_{INJECTED} \times CLAMP_COUPLING_RATIO$
$V_{ERROR} = I_{LEAK} \times (R_{CLAMP_INTERNAL} + R_{SOURCE})$
$Code\ Error = V_{ERROR} \times 2^{12} / V_{REF}$

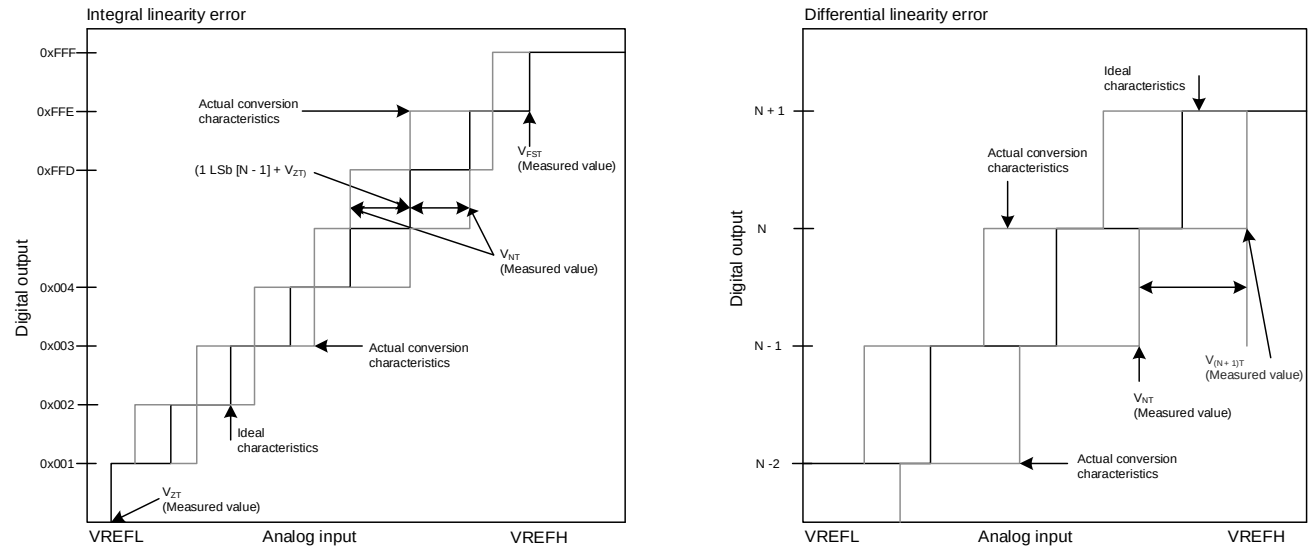
Where:

$I_{INJECTED}$ is the injected current in mA.

I_{LEAK} is the calculated leakage current in mA.

V_{ERROR} is the voltage error calculated due to leakage currents in V.

V_{REF} is the ADC reference voltage in V.

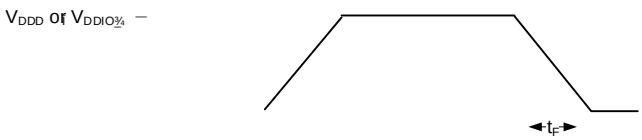


Integral linearity error of digital output N = $(V_{NT} – \{1\ LSb \times (N – 1) + V_{ZT}\}) / 1\ LSb$	[LSb]
Differential linearity error of digital output N = $(V_{(N+1)T} – V_{NT} – 1\ LSb\ ) / 1\ LSb$	[LSb]
$1\ LSb = (V_{FST} – V_{ZT}\ ) / 4094$	[V]

SID104	V_{ZT}	Zero transition voltage	–20	–	20	mV	V_{DDA} = 2.7 V to 5.5 V, –40 °C T_A

SID113F	$t_{S_2P7_A}$	Analog input sample time for channels of another SARMUXn (n=1,2)	1648	–	–	ns	2.7V ≤ V_{DDA} ≤ 4.5 V When ADC0 borrows the SARMUX of another ADC, guaranteed by design
SID106	t_{ST_4P5}	ADC max throughput (samples per second) when using the SARMUX of own ADC	–	–	1	Msps	4.5 V V_{DDA} 5.5 V, 80 MHz / 3 = 26.67 MHz, 11 sampling cycles, 15 conversion cycles
SID106A	t_{ST_2P7}	ADC max throughput (samples per second) when using the SARMUX of own ADC	–	–	0.5	Msps	2.7 V V_{DDA} < 4.5 V 80 MHz / 6 = 13.3 MHz, 11 sampling cycles, 15 conversion cycles
SID106B	$t_{ST_4P5_A}$						

Unless otherwise noted, the timings are defined with the guidelines mentioned in the

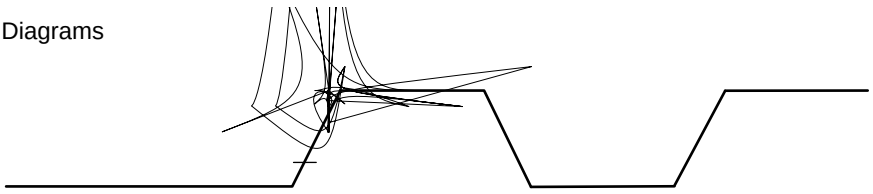


All specifications are valid for –40 °C ≤ T_A ≤ 125 °C and for 2.7 V to 5.5 V except where noted.

SID120	f_C	TCPWM operating frequency	–	–	100	MHz	f_C = peripheral clock
SID121	$t_{PWMENEXT}$	Input trigger pulse width for all trigger events	2 / f_C	–	–	ns	Trigger Events can be Stop, Start, Reload, Count, Capture, or Kill depending on which mode of operation is selected.
SID122	t_{PWMEXT}	Output trigger pulse widths	2 / f_C	–	–	ns	Minimum possible width of Overflow, Underflow, and Counter = Compare (CC) value trigger outputs
SID123	t_{CRES}	Resolution of counter	1 / f_C	–	–	ns	Minimum time between successive counts
SID124	t_{PWMRES}	PWM resolution	1 / f_C	–	–	ns	Minimum pulse width of PWM output
SID125	t_{QRES}	Quadrature inputs resolution	2 / f_C	–	–	ns	Minimum pulse width between Quadrature phase inputs.

TCPWM Timing Diagrams

Input Signal

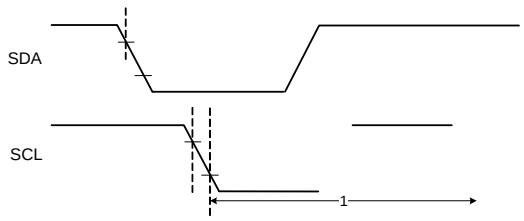


SID129	f_{SCB}	SCB operating frequency	–	–	100	MHz	
SID130	f_{SCL}	SCL clock frequency	–	–	100	kHz	
SID131	$t_{HD;STA}$	Hold time, START condition	4000	–	–	ns	
SID132	t_{LOW}	Low period of SCL	4700	–	–	ns	
SID133	t_{HIGH}	High period of SCL	4000	–	–	ns	
SID134	$t_{SU;STA}$	Setup time for a repeated START	4700	–	–	ns	
SID135	$t_{HD;DAT}$	Data hold time, for receiver	0	–	–	ns	
SID136	$t_{SU;DAT}$	Data setup time	250	–	–	ns	
SID138	t_F	Fall time of SCL and SDA	–	–	300	ns	Input and output
SID139	$t_{SU;STO}$	Setup time for STOP	4000	–	–	ns	
SID140	t_{BUF}						

SID162	$t_{VD;DAT_F}$	

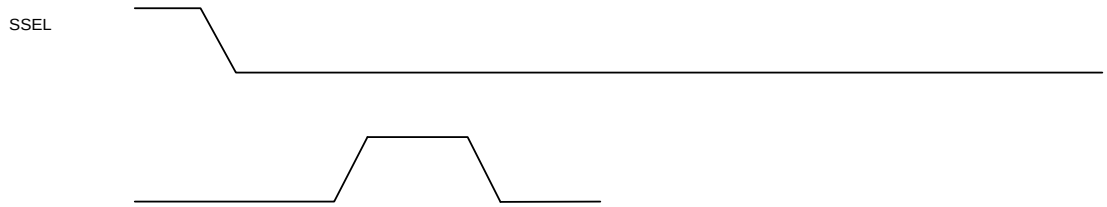
(continued)

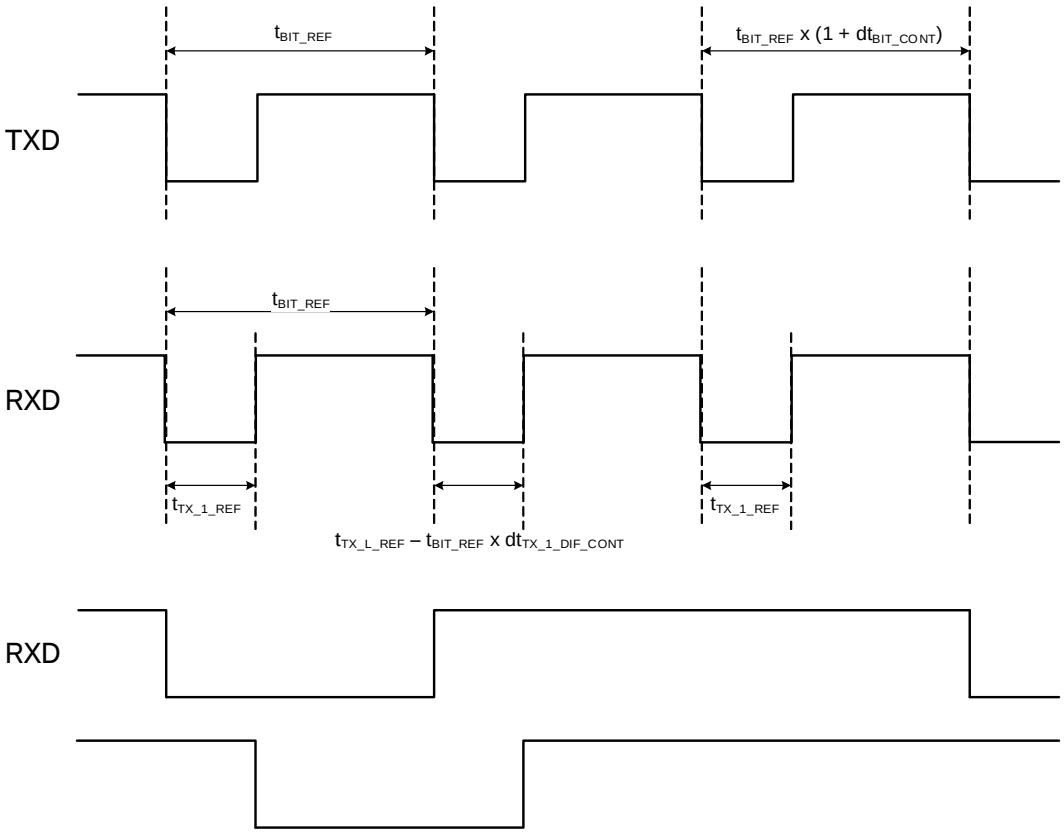
SID222	t_{DSO_EXT}	SPI Slave: MISO valid after Sclock driving edge, in the external-clocked mode	–	–	32	ns
SID223	t_{HSO_EXT}	SPI Slave: Previous MISO data hold time	3	–	–	ns
SID224	$t_{EN_SETUP_EXT}$	SPI Slave: SSEL valid to first SCK valid edge	40	–	–	ns
SID225	$t_{EN_HOLD_EXT}$	SPI Slave Select active (LOW) from last SCLK hold	40	–	–	ns
SID226	$t_{W_DIS_EXT}$	SPI Slave Select inactive time	80	–	–	ns
SID227	$t_{W_SCLKH_EXT}$	SPI SCLK pulse width HIGH	34	–	–	ns
SID228	$t_{W_SCLKL_EXT}$	SPI SCLK pulse width LOW	34	–	–	ns
SID229	t_{SIH_EXT}	SPI MOSI hold from SCLK	20	–	–	ns
SID230	C_{SPIS_EXT}	SPI capacitive load	–	–	10	pF
SID231	t_{VSS_EXT}	SPI Slave: MISO valid after SSEL falling edge (CPHA = 0)	–	–	33	ns
SID240	f_{BPS}	Data rate	–	–	10	Mbps



START condition

SPI Slave Timing Diagrams
CPHA=0

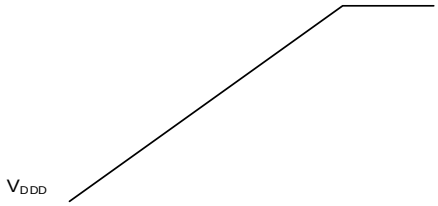



TXD
t_{BIT_REF}
t_{BIT_REF} x (1 + dt_{BIT_CONT})
RXD
t_{BIT_REF}
$t_{TX_1_REF}$
$t_{TX_1_REF}$
$t_{TX_L_REF} - t_{BIT_REF}$ x $dt_{TX_1_DIF_CONT}$
RXD

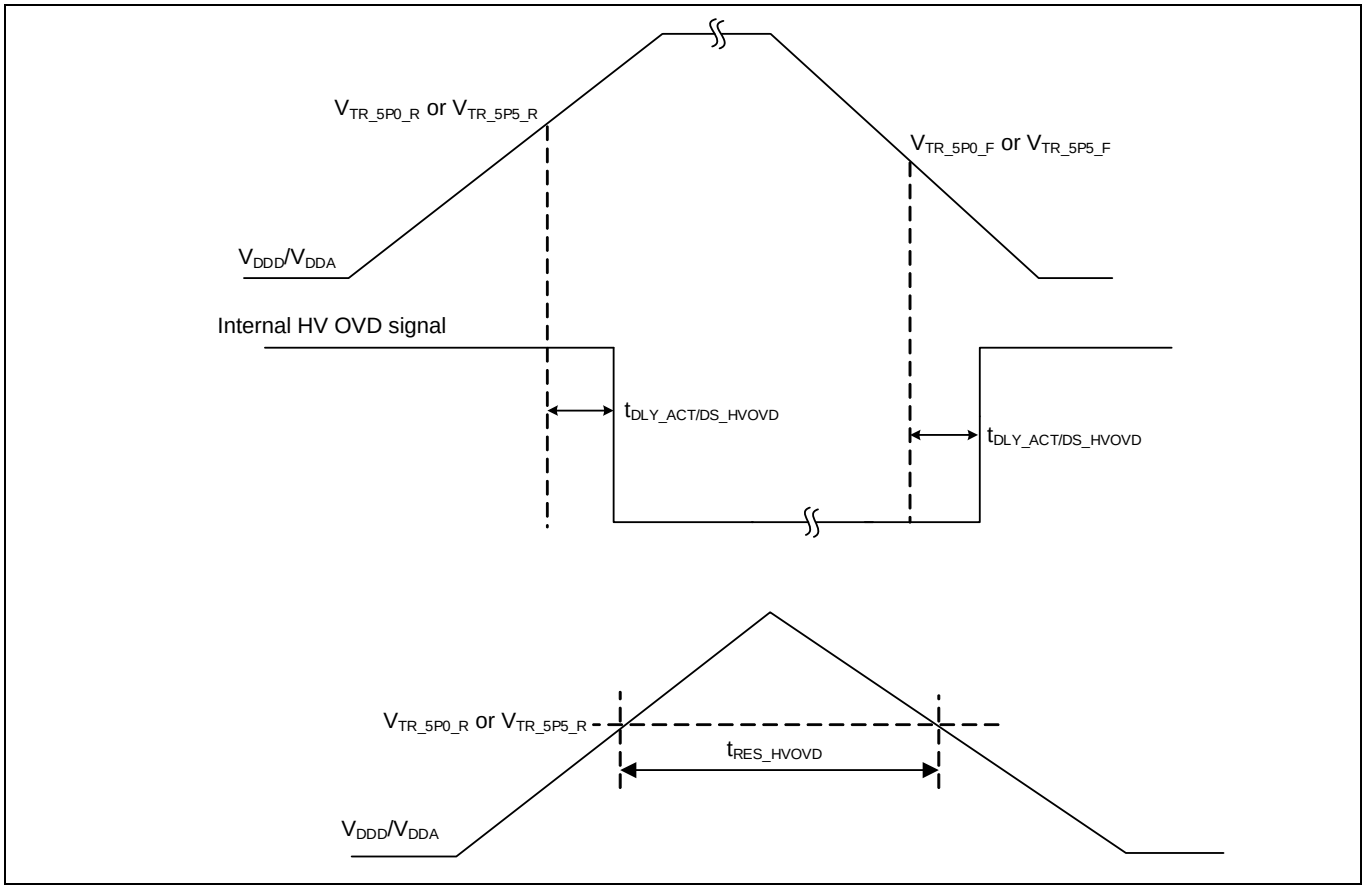
All specifications are valid for –40 °C ≤ T_A ≤ 125 °C and for 2.7 V to 5.5 V except where noted.

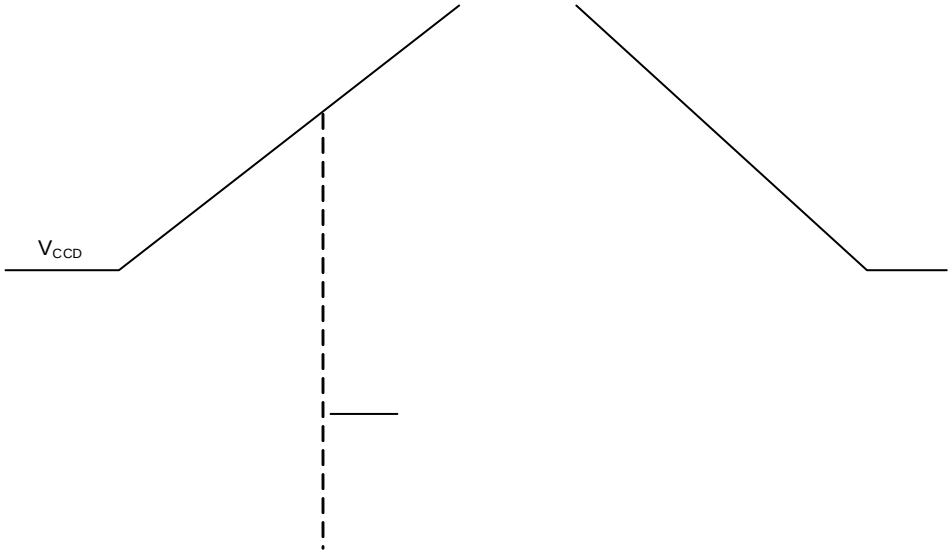
SID260	V_{PE}	Erase and program voltage	2.7	–	5.5	V	
SID257	f_{FO}	Maximum flash memory operation frequency	–	–	100	MHz	Zero wait access to code-flash memory up to 100 MHz Zero wait access with cache hit up to 160 MHz
SID254	t_{ERS_SUS}	Maximum time from erase suspend command till erase is indeed suspend	–	–	37.5	µs	
SID255	$t_{ERS_RES_SUS}$	Minimum time allowed from erase resume to erase suspend	250	–	–	µs	Guaranteed by design
SID258	t_{BC_WF}	Blank check time for N-bytes of work-flash	–	–	10 + 0.3 × N	µs	At 100 MHz, N ≥ 4 and multiple of 4, excludes system overhead time
SID259	$t_{SECTORERASE1}$	Sector erase time (Code-flash: 32 KB)	–	45	90	ms	Includes internal preprogramming time
SID259A	$t_{SECTORERASE2}$	Sector erase time (Code-flash: 8 KB)	–	15	30	ms	Includes internal preprogramming time
SID261	$t_{SECTORERASE3}$	Sector erase time					

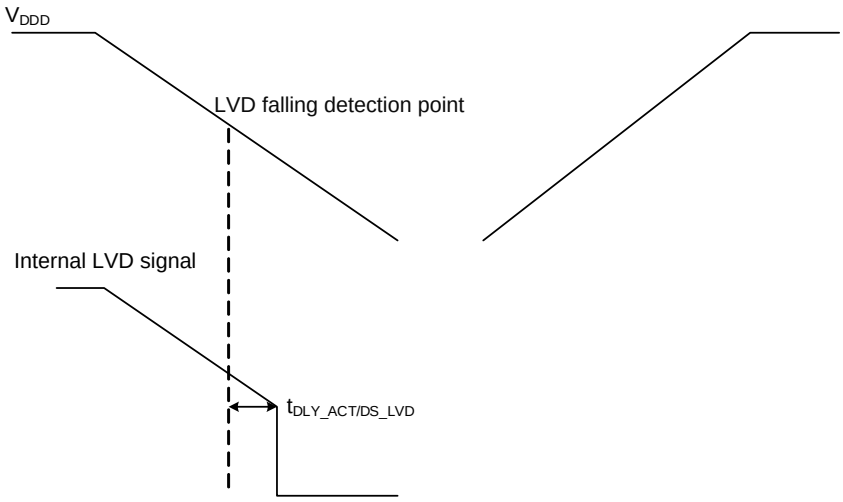
SID576	t_{RES_LVD}	Response time of LVD, V_{DDD} supply. LVD guaranteed to generate pulse for V_{DDD} pulse width greater than this. (For falling-then-rising supply at max ramp rate; pulse width is time below LVD falling point)	100	–	–	ns	Guaranteed by design
SID580	$V_{TR_5P0_R}$	HV OVD 5.0-V rising detection point for V_{DDD} and V_{DDA}	5.049	5.205	5.361	V	
SID581	$V_{TR_5P0_F}$						



Internal reset by POR


$V_{TR_5P0_R}$ or $V_{TR_5P5_R}$
$V_{TR_5P0_F}$ or $V_{TR_5P5_F}$
V_{DDD}/V_{DDA}
Internal HV OVD signal
t_{DLY_ACT/DS_HVOVD}
t_{DLY_ACT/DS_HVOVD}
$V_{TR_5P0_R}$ or $V_{TR_5P5_R}$
t_{RES_HVOVD}
V_{DDD}/V_{DDA}


V_{CCD}


V_{DDD}
LVD falling detection point
Internal LVD signal
t_{DLY_ACT/DS_LVD}

All specifications are valid for –40 °C T_A 125 °C and for 2.7 V to 5.5 V except where noted.

The basic requirement on the clock-frequency dependency of the cores is that the Cortex-M0+ core should run at an integer divider from the Cortex-M4F core clock. Example combinations are listed in .

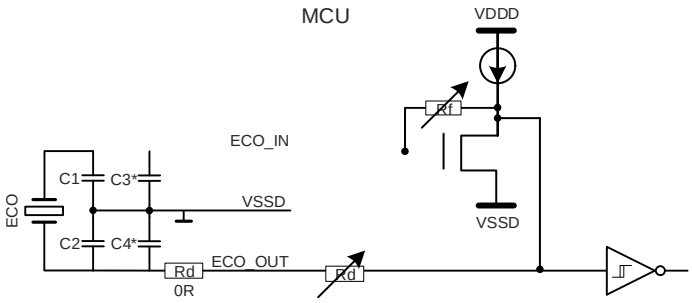
160	80
120	60
100	100
80	80

[56]

	[57]		[57]		
			[58]	[59]	
CLK_HF0	160	PLL200#0	160	155	Root clock for CPUSS, PERI
		FLL	100	96	
	100	PLL200#0	100	98	
		FLL	100	96	
CLK_HF1	100	PLL200#0	100	98	Event generator (CLK_REF), Clock output on EXT_CLK pins (when used as output)
		FLL	100	96	
CLK_HF2	2	ILO	NA	NA	CSV
CLK_FAST	160	PLL200#0	160	155	Generated by dividing CLK_HF0, intermediate clock for CM4
		FLL	100	96	
	100	PLL200#0	100	98	
		FLL	100	96	
CLK_SLOW	100	PLL200#0	100	98	Generated by clock gating CLK_PERI, intermediate clock for CM0+, Crypto, P-DMA, M-DMA
		FLL	100	96	
CLK_PERI	100	PLL200#0	100	98	Generated by clock gating CLK_HF0, intermediate clock for LIN, SCB, PASS, CAN, TCPWM, CXPI, IOSS, CPU trace
		FLL	100	96	

56. Intermediate clocks that are not listed have the same limitations as that of their parent clock.
57. Maximum clock frequency after the corresponding clock source (PLL/FLL + dividers). All internal tolerances and affects are covered by these frequencies.
58. For ECO: up to ±150 ppm uncertainty of the external clock source are tolerated by design.
59. The IMO operation frequency tolerance is included. When DeepSleep mode isn't used, maximum permitted clock frequency setting of clock source IMO case is equal to clock source ECO case.

SID310A	$f_{IMOTOLA}$	IMO operating frequency	7.68	8	8.32	MHz	
SID311	$t_{STARTIMO}$	IMO startup time	–	–	7.5	µs	Startup time to 90% of final 100865.3(nHz)]T3/SI2311t



ECO_IN: External crystal oscillator input pin
ECO_OUT: External crystal oscillator output pin

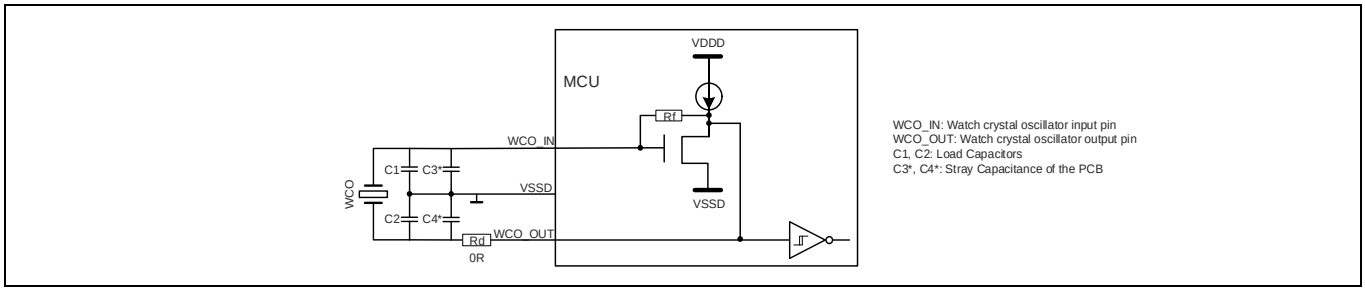
[61]

(continued)

SID348A	I_{PLL_80M2}	PLL operating current (f_{OUT} = 80 MHz)	–	530	795	µA	f_{IN} = 8 MHz, f_{PFD} = 8 MHz, f_{VCO} = 240 MHz, f_{OUT} = 80 MHz
SID348B	I_{PLL_80M3}	PLL operating current (f_{OUT} = 80 MHz)	–	530	795	µA	f_{IN} = 16 MHz, f_{PFD} = 8 MHz, f_{VCO} = 240 MHz, f_{OUT} = 80 MHz
SID348C	f_{PLL_VCO}	VCO frequency	170	–	400	MHz	
SID349C	f_{PLL_PFD}	PFD frequency	3.988	–	8	MHz	

SID350	t_{FLL_WAKE}	FLL wake up time	–	–	5	µs	Wakeup with < 10 °C temperature change while in DeepSleep. f_{FLL_IN} = 8 MHz, f_{FLL_OUT} = 100 MHz, Time from stable reference clock until FLL frequency is within 5% of final value
SID351	f_{FLL_OUT}	Output frequency from FLL block	24	–	100	MHz	Output range of FLL divided-by-2 output
SID352	FLL_CJIT	FLL frequency accuracy	–1	–	1	%	This is added to the error of the source
SID353	f_{FLL_IN}	Input frequency	0.25	–	80	MHz	
SID354	I_{FLL}	FLL operating current	–	250	360	µA	Reference clock: IMO, CCO frequency: 200 MHz, FLL frequency: 100 MHz, guaranteed by design

SID360	f_{WCO}	Watch Crystal frequency	–	32.768	–	kHz	Maximum drive level: 0.5 µW
SID361	WCO_DC	WCO duty cycle	10	–	90	%	
SID362	t_{START_WCO}	WCO start-up time[62]	–	–	1000	ms	For Grade-S devices Time from set CTL.WCO_EN to 1 until STATUS.WCO_OK is set to 1. (See Clock Timing Diagrams)
SID362E	t_{START_WCOE}	WCO start-up time[62]	–	–	1400	ms	For Grade-E devices Time from set CTL.WCO_EN to 1 until STATUS.WCO_OK is set to 1. (See Clock Timing Diagrams)
SID363	I_{WCO}	WCO current	–	1.4	–	µA	



[63]

SID366	f_{EXT}	External clock input frequency	0.25	–	80	MHz	For EXT_CLK pin (all input level settings: CMOS, TTL, Automotive)
SID367	EXT_DC	External clock duty cycle	45	–	55	%	

62.Mainly depends on the external crystal.
63.See the family specific Architecture TRM for more information on crystal requirements (002-19314, TRAVEO™ T2G Automotive MCU body controller entry architecture technical reference manual).



SID410	t_{MCWDT1}	Minimum MCWDT timeout	58.12	–	–	µs	When using the ILO (32.768 kHz + 5%) and 16-bit MCWDT counter Guaranteed by design
SID411	t_{MCWDT2}	Maximum MCWDT timeout	–	–	2.11	s	When using the ILO (32.768 kHz – 5%) and 16-bit MCWDT counter Guaranteed by design

SID412	t_{WDT1}	Minimum WDT timeout	58.12	–	–	µs	When using the ILO (32.768 kHz + 5%) and 32-bit WDT counter Guaranteed by design
SID413	t_{WDT2}	Maximum WDT timeout	–	–	38.33	h	When using the ILO (32.768 kHz – 5%) and 32-bit WDT counter Guaranteed by design
SID414	t_{WDT3}	Default WDT timeout	–	1000	–	ms	When using the ILO and 32-bit WDT counter at 0x8000 (default value), guaranteed by design

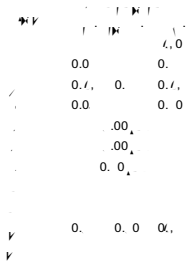
Ordering information

CYT2BL is offered in the packages listed in the .

176-LQFP	24 × 24 × 1.7 mm (max)	0.5 mm	a1[71]

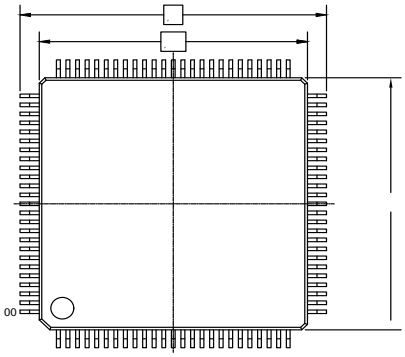
176-LQFP	260	30	3
144-LQFP	260	30	3
100-LQFP	260	30	3
80-LQFP	260	30	3
64-LQFP	260	30	3

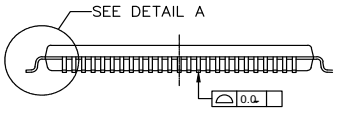
NOTES

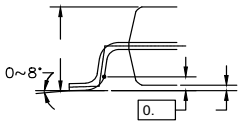


NOTES

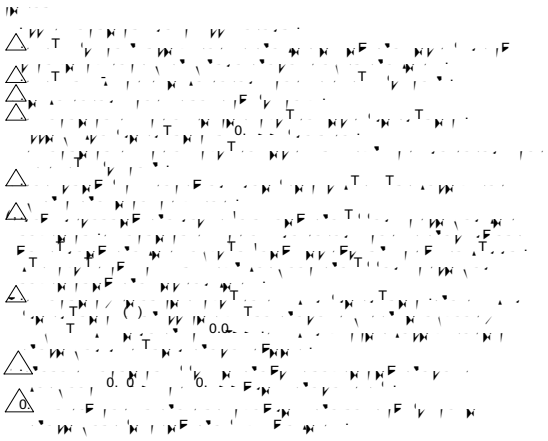
2. DATUM PLANE H IS LOCATED AT THE BOTTOM OF THE MOLD PARTING LINE COINCIDENT WITH WHERE THE LEAD EXITS THE BODY.
3. DATUMS A–B AND D TO BE DETERMINED AT DATUM PLANE H.
4. TO BE DETERMINED AT SEATING PLANE C.
5. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
   ALLOWABLE PROTRUSION IS 0.2







	—	—	1.0
	0.0	—	0.
	0.	—	0.1
	0.0	—	0. 0
	.00		
	.00		
	0. 0		
	.00		
	.00		
	0.	0. 0	0.
	.00		



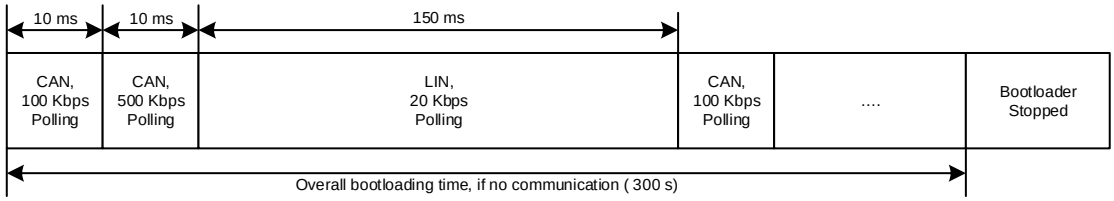
Packaging

	—	—	1.60
	0.00	—	0.50
	0.50		0.60
	0.08	—	0.20
	.00		
	0.00		
	0.50		
	.00		
	0.00		
	0°	3.5°	7°
	.00		

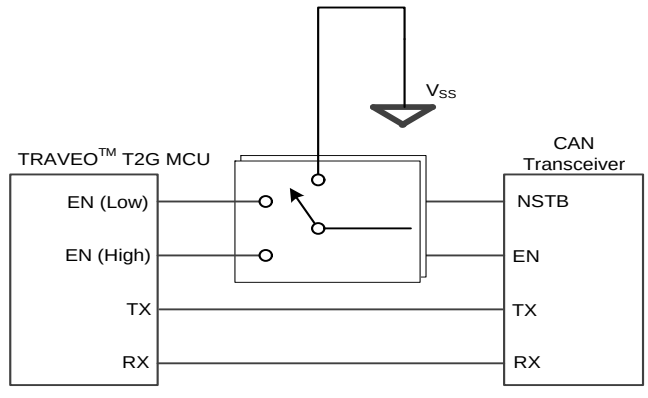
NOTES

1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. DATUM PLANE H IS LOCATED AT THE BOTTOM OF THE MOLD PARTING LINE COINCIDENT WITH WHERE THE LEAD EXITS THE BODY.
3. DATUMS A–B AND D TO BE DETERMINED AT DATUM PLANE H.
4. TO BE DETERMINED AT SEATING PLANE C.
5. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTR

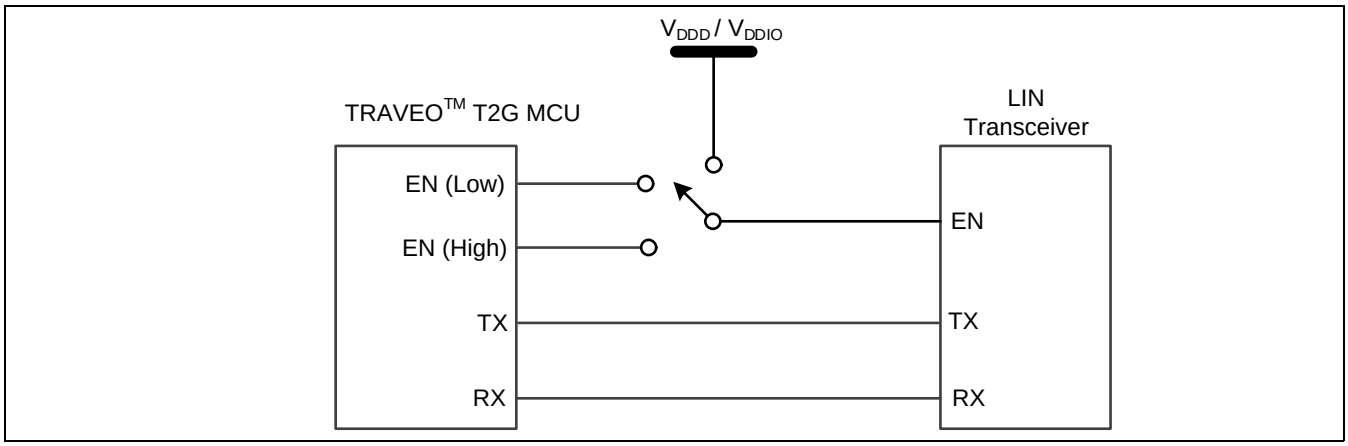
- Triggered at device startup, if a trigger condition is applied
- Either CAN or LIN communication may be used
- Bootloader polls for the communication on CAN or LIN at separate time frames, until the overall 300-second timeout is reached
- If a bootloader command is received on either communication interface, the polling stops and bootloader starts using this interface



1	CAN Mode	Classic CAN
2	CAN Instance	CAN0, Channel#1
3	CAN TX	P0.2 / CAN0_1_TX
4	CAN RX	P0.3 / CAN0_1_RX
5	CAN Transceiver NSTB / EN (Low)	P23.3 (optional)
6	CAN Transceiver EN / EN (High)	P2.1 (optional)
7	CAN RX Message ID	0x1A1
8	CAN TX Message ID	0x1B1
9	Baud	100 or 500 kbps alternating



1	LIN Type	LIN0, Channel#1
2	LIN Mode	Slave
3	LIN Checksum Type	Classic
4	LIN TX	P0.1 / LIN1_TX
5	LIN RX	P0.0 / LIN1_RX
6	LIN EN / EN (High)	P2.1 (optional)
7	LIN EN (Low)	P23.3 (optional)
8	LIN TX PID	0x46
9	LIN RX PID	0x45
10	Baud	20 or 115.2 kbps
11	Break Field Length	11
12	Break Delimiter Length	1 bit



CAN FD	mxttcanfd	M_TTCAN IP revision: Rev.3.2.3	Bosch
Arm® Cortex®-M0+	armcm0p	Cortex®-M0+-r0p1	Arm®
Arm® Cortex®-M4F	armcm4	Cortex®-M4-r0p1	Arm®
Arm® Coresight	armcoresighttk	CoreSight-SoC-TM100-r3p2	Arm®

A/D	Analog to Digital	IRQ	Interrupt request
ABS	Absolute	JTAG	Joint test action group
ADC	Analog to Digital converter	LDO	Low drop out regulators
AES	Advanced encryption standard	LIN	Local Interconnect Network, a communications protocol
AHB	AMBA (advanced microcontroller bus architecture) high-performance bus, Arm® data transfer bus	LVD	Low voltage detection
Arm®	Advanced RISC machine, a CPU architecture	OTA	Over-the-air programming

(continued)

IOSS	Input/output sub-system	SCB	Serial communication block
IPC	Inter-processor communication	SCL	I^2C serial clock
IrDA	Infrared interface	SDA	I^2C serial data
SECDED	Single error correction, double error detection	TCPWM	Timer/Counter Pulse-width modulator
SHA	Secure hash algorithm	TTL	Transistor-transistor logic
SHE	Secure hardware extension	TRNG	True random number generator
SMPU	Shared memory protection unit	UART	Universal Asynchronous Transmitter Receiver
SPI	Serial peripheral interface, a communications protocol	WCO	Watch crystal oscillator
SRAM	Static random access memory	WDT	Watchdog timer reset
SWD	Serial wire debug	XRES_L	External reset I/O pin
SWJ	Serial wire JTAG		

STATUS_ADDR_PROTECTED will be returned if the ConfigureFmInterrupt API is called with arguments stored in SRAM with 4-byte boundary (available SRAM or protected boundary SRAM).

RX FIFO top pointer function calculates the address for received messages in Message RAM by hardware. This address should restart from the start address after reading all messages of RX FIFO n size (n: 0 or 1). However, the address does not restart from the start address when RX FIFO n size is set to 1(CANFD_CH_RXFnC.FnS = 0x01). This results in CPU/DMA reading messages from the wrong address in Message RAM.

NA

The RX FIFO top pointer function is used when RX FIFO n size is set to 1 element (CANFD_CH_RXFnC.FnS = 0x01).

Received message cannot be correctly read by using the RX FIFO top pointer function, when RX FIFO n size is set to 1 element.

Any of the following can be used as a workaround:
1) Set RX FIFO n size to 2 or more when using RX FIFO top pointer function.
2) Do not use the RX FIFO top pointer function when RX FIFO n size is set to 1 element. Instead of RX FIFO top pointer, read received messages from the Message RAM directly.

No silicon fix planned. Use workaround.

If either of the CANFD_CH_CCCR.INIT bits 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.

NA

Either of the CANFD_CH_CCCR.INIT bits is set by the Host or when the M_TTCAN module enters BusOff state.

The errata is limited to the use case when the debug on CAN functionality is active. Normal operation of the 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 CANFD_CH_RXF1S.DMS bit. In case CANFD_CH_RXF1S.DMS is set to 0b11, the DMA request remains active.

In case the debug message handling state machine has stopped whil

Configuration:
Several Tx buffers are configured with same Message ID. Transmission of these Tx buffers is requested sequen-

In the case the dedicated Tx buffers with the same Message ID are not requested in ascending order or at the same time or in case of multiple Tx Queue buffers with the same Message ID, it cannot be guaranteed, that these messages are transmitted in ascending order with lowest buffer number first.

Code Flash memory can be erased in “Non-Blocking” mode; a Non-Blocking mode supported option allows users to suspend an ongoing erase sector operation. When an ongoing erase operation is interrupted using “Erase Suspend” and “Erase Resume”, Flash cells may not have been erased completely, even after the erase operation complete is indicated by FLASHC_STATUS register. Only Code Flash is impacted by this issue, Work Flash and Supervisory Flash (SFlash) are not impacted.

N/A

Using EraseSector System Call in Non-Blocking mode for CM0+ to erase Code Flash and the ongoing erase operation is interrupted using EraseSuspend and EraseResume System calls.

When Code Flash sectors are erased in Non-Blocking mode and the ongoing erase operation is interrupted by Erase Suspend / Erase Resume, it cannot be guaranteed that the Code Flash cells are fully erased. Any read on the Code Flash area after the erase is complete or read on the programmed data after ProgramRow is complete can trigger ECC errors.

Use any of the following:
1) User can use Non-Blocking mode for EraseSector, but must not interrupt the erase operation using Erase Suspend / Erase Resume.
2) If a Code Flash sector erase operation is interrupted using Erase Suspend / Erase Resume, then erase the same sector again without Erase Suspend / Erase Resume before reading the sector or programming the sector.

Fixed to update the Flash settings from date code 304xxxxx.

The port state is not retained when the port selects peripheral IP (except LIN or CAN FD) and MCU wakes up from DeepSleep.

N/A

	DeepSleep should not be entered while any FLL/PLL is enabled and uses ECO as its reference clock. Since the unstable ECO clock after wakeup is outside the allowed reference clock limits for FLL/PLL, it is possible for DeepSleep wakeup to fail.
	N/A
	DeepSleep transition while any FLL/PLL is enabled and uses ECO as its reference clock.
	Possibility of DeepSleep wakeup failure.
	If any FLL/PLL operates with the ECO as its reference clock, change the clock to either ECO direct or IMO direct or IMO with FLL/PLL before entering DeepSleep.
	No silicon fix planned. TRM was updated.

	The following registers are described as 'Retained' in the Register TRM while it is not guaranteed that the value before entering DeepSleep mode is still readable from the register. - SARADC: PASSx_SARy_CHz_RESULT - SRSS: PWR_LVD_STATUS - SRSS: PWR_LVD_STATUS2 - SRSS: CLK_CAL_CNT1 - SRSS: CLK_CAL_CNT2 - SRSS: CLK_FLL_STATUS - SRSS: WDT_INTR - SRSS: WDT_INTR_MASKED - SRSS: CLK_PLL400Mx_STATUS
	N/A
	Use of the related function and wakeup from DeepSleep mode.
	The values before entering DeepSleep are not retained.
	For PASSx_SARy_CHz_RESULT, do any of the following: 1) Store the conversion values at another memory location before entering DeepSleep mode 2) Restart the conversion after wakeup from DeepSleep mode For the other registers: Rewrite the register value or read the status flags again after wakeup.
	No silicon fix planned. TRM was updated.

	There is possibility of setting the SCBx_INTR_TX.UNDERFLOW bit even if the FIFO is not empty.
	N/A
	Using the TX FIFO for SCB when the AHB-Lite interface clock (CLK_GR6) frequency of the AHB bus is greater than 3x the SCB functionality clock (PCLK_SCBx_CLOCK).
	SCBx_INTR_TX.UNDERFLOW bit may be set unintentionally.
	Ignore the SCBx_INTR_TX.UNDERFLOW bit if the FIFO is not empty.
	No silicon fix planned. TRM was updated.

	ReadSWPU or WriteSWPU 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.
	N/A
	Calling ReadSWPU or WriteSWPU while executing EraseSector or ProgramRow in non-blocking mode on bank#0 (or bank#1 if dual bank mode with mapping B is used).
	ReadSWPU or WriteSWPU can't 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).
	Do not use ReadSWPU or WriteSWPU while executing EraseSector or ProgramRow in non-blocking mode on bank#0 (or bank#1 if dual bank mode with mapping B is used).
	No silicon fix planned. TRM was updated.
	S-LLD, HSM-Perf-Lib: 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 any of following: a) call CySldProt_GetSwpuFlashStructCfg b) call CySldProt_VerifySecureDomainFlashWriteProtection if CySldProt_SwpuFlashStructGroupConfigurations is non-empty

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).

Under the following conditions, a received message has corruto th9the CAN 17.58 lfQ1 Tf-45...DBiv19.1(s)1(s)-7.1(c)das-6(div19.(

1.M_TTCAN always starts 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 runs 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.

N/A

Under the following conditions a received message has 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.

The erratum is limited to the case when the Host clock frequency used in the actual device is below the limit shown in .
Corrupted data is written to the Rx FIFO element from the respective dedicated Rx Buffer.
The received frame is nevertheless signaled as valid.

Check whether the minimum Host clock frequency (shown in) 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.

Try a different configuration by changing the following parameters until the actual Host clock frequency (CLK_GR5) is above the minimum host frequency shown in :

- 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.

	No silicon fix planned. Use workaround.
	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.

**	2019-11-26	New datasheet for new device family.
*A	2019-12-12	Updated .
*B	2020-04-01	Changed datasheet status to Preliminary. Updated . Updated and .
*C	2021-07-08	Updated . Updated . Updated . Updated . Updated and . Updated . Updated . Updated . Updated . Updated and . Updated .
*D	2021-10-13	Updated Updated Updated Updated

SID17A	Output voltage	Units	(none)	V	Correction
SID113	Analog input sample time for channels of own SARMUX	Description	Description: Analog input sample time	Description: Analog input sample time for channels of own SARMUX	Updated description
SID113A	Analog input sample time for channels of own SARMUX	Description	Description: Analog input sample time	Description: Analog input sample time for channels of own SARMUX	Updated description
SID113E	Analog input sample time for channels of another SARMUXn (n=1,2)	All	(none)	Parameter: tS_4P5_A Description: Analog input sample time for channels of another SARMUXn (n=1,2) Min: 824 ns Typ: – Max: – Details/Conditions: 4.5 V <= VDDA <= 5.5 V When ADC0 borrows the SARMUX of another ADC, guaranteed by design	Added spec
SID113F	Analog input sample time for channels of another SARMUXn (n=1,2)	All	(none)	Parameter: tS_2P7_A Description: Analog input sample time for channels of another SARMUXn (n=1,2) Min: 1648 ns Typ: – Max: – Details/Conditions: 2.7V ≤ VDDA < 4.5 V When ADC0 borrows the SARMUX of another ADC, guaranteed by design	Added spec
SID106	ADC max throughput (samples per second) when using the SARMUX of own ADC	Description	Description: Max throughput (Sample per second)	Description: ADC max throughput (samples per second) when using the SARMUX of own ADC	Updated description
SID106A	ADC max throughput (samples per second) when using the SARMUX of own ADC	Description	Description: Max throughput (Sample per second)	Description: ADC max throughput (samples per second) when using the SARMUX of own ADC	Updated description
SID106B	ADC0 max throughput (samples per second) when borrowing the SARMUXn of another ADC (n=1,2)	All	(none)	Parameter: tST_4P5_A Description: ADC0 max throughput (samples per second) when borrowing the SARMUXn of another ADC (n=1,2) Min: – Typ: – Max: 0.5 Msps Details/Conditions: 4.5 V ≤ VDDA ≤ 5.5 V, 80 MHz/6 = 13.3 MHz, 11 sampling cycles, 15 conversion cycles	Added spec
SID106C	ADC0 max throughput (samples per second) when borrowing the SARMUXn of another ADC (n=1,2)	All	(none)	Parameter: tST_2P7_A Description: ADC0 max throughput (samples per second) when borrowing the SARMUXn of another ADC (n=1,2) Min: – Typ: – Max: 0.25 Msps Details/Conditions: 2.7V ≤ VDDA < 4.5 V, 80 MHz / 12 = 6.67 MHz, 11 sampling cycles, 15 conversion cycles	Added spec

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

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”).

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, 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.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

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 ( ).

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.