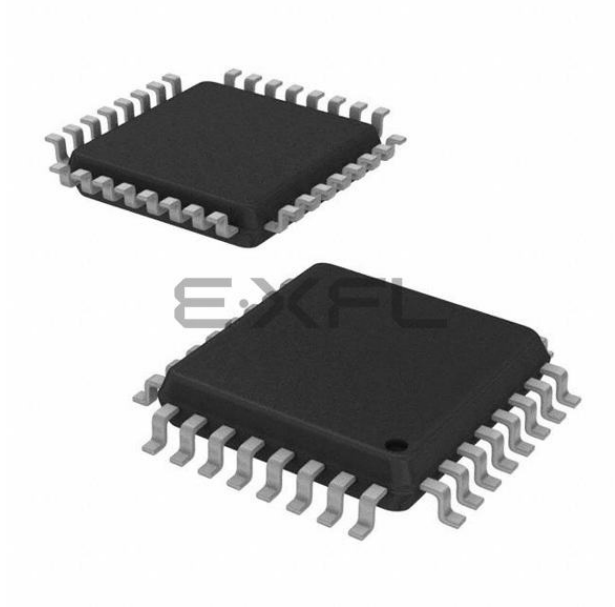

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	12V1
Core Size	16-Bit
Speed	25MHz
Connectivity	CANbus, IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	26
Program Memory Size	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	1.5K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3.13V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12g48f1vlcr

Table 2-32. PUCR Register Field Descriptions (continued)

Field	Description
2 PUPCE	**Port C Pullup Enable**—Enable pullup devices on all port input pins This bit configures whether a pullup device is activated on all associated port input pins. If a pin is used as output this bit has no effect. 1 Pullup devices enabled 0 Pullup devices disabled
1 PUPBE	**Port B Pullup Enable**—Enable pullup devices on all port input pins This bit configures whether a pullup device is activated on all associated port input pins. If a pin is used as output this bit has no effect. 1 Pullup devices enabled 0 Pullup devices disabled
0 PUPAE	**Port A Pullup Enable**—Enable pullup devices on all port input pins This bit configures whether a pullup device is activated on all associated port input pins. If a pin is used as output this bit has no effect. 1 Pullup devices enabled 0 Pullup devices disabled

[1] Read: Anytime
Write: Anytime

Table 2-73. PIEJ Register Field Descriptions

Field	Description
7-0 PIEJ	**Port J interrupt enable—** This bit enables or disables the edge sensitive pin interrupt on the associated pin. An interrupt can be generated if the pin is operating in input or output mode when in use with the general-purpose or related peripheral function. 1 Interrupt is enabled 0 Interrupt is disabled (interrupt flag masked)

2.4.3.48 Port J Interrupt Flag Register (PIFJ)

Address 0x026F (G1, G2) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PIFJ7	PIFJ6	PIFJ5	PIFJ4	PIFJ3	PIFJ2	PIFJ1	PIFJ0
Reset	0	0	0	0	0	0	0	0

Address 0x026F (G3) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	PIFJ3	PIFJ2	PIFJ1	PIFJ0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-48. Port J Interrupt Flag Register (PIFJ)

[1] Read: Anytime
Write: Anytime, write 1 to clear

Table 2-74. PIFJ Register Field Descriptions

Field	Description
7-0 PIFJ	**Port J interrupt flag—** This flag asserts after a valid active edge was detected on the related pin (see Section 2.5.4.2, “Pin Interrupts and Wakeup”). This can be a rising or a falling edge based on the state of the polarity select register. An interrupt will occur if the associated interrupt enable bit is set. Writing a logic “1” to the corresponding bit field clears the flag. 1 Active edge on the associated bit has occurred 0 No active edge occurred

Table 2-85. PPS0AD Register Field Descriptions

Field	Description
7-0 PPS0AD	**Port AD pull device select**—Configure pull device and pin interrupt edge polarity on input pin This bit selects a pullup or a pulldown device if enabled on the associated port input pin. This bit also selects the polarity of the active pin interrupt edge. 1 Pulldown device selected; rising edge selected 0 Pullup device selected; falling edge selected

2.4.3.60 Port AD Polarity Select Register (PPS1AD)

Address 0x027B Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PPS1AD7	PPS1AD6	PPS1AD5	PPS1AD4	PPS1AD3	PPS1AD2	PPS1AD1	PPS1AD0
Reset	0	0	0	0	0	0	0	0

Figure 2-59. Port AD Polarity Select Register (PPS1AD)

[1] Read: Anytime
Write: Anytime

Table 2-86. PPS1AD Register Field Descriptions

Field	Description
7-0 PPS1AD	**Port AD pull device select**—Configure pull device and pin interrupt edge polarity on input pin This bit selects a pullup or a pulldown device if enabled on the associated port input pin. This bit also selects the polarity of the active pin interrupt edge. 1 Pulldown device selected; rising edge selected 0 Pullup device selected; falling edge selected

7.4.7 Serial Interface Hardware Handshake Protocol

BDM commands that require CPU execution are ultimately treated at the MCU bus rate. Since the BDM clock source can be modified when changing the settings for the VCO frequency (CPMUSYNR), it is very helpful to provide a handshake protocol in which the host could determine when an issued command is executed by the CPU. The BDM clock frequency is always VCO frequency divided by 8. The alternative is to always wait the amount of time equal to the appropriate number of cycles at the slowest possible rate the clock could be running. This sub-section will describe the hardware handshake protocol.

The hardware handshake protocol signals to the host controller when an issued command was successfully executed by the target. This protocol is implemented by a 16 serial clock cycle low pulse followed by a brief speedup pulse in the BKGD pin. This pulse is generated by the target MCU when a command, issued by the host, has been successfully executed (see Figure 7-10). This pulse is referred to as the ACK pulse. After the ACK pulse has finished: the host can start the bit retrieval if the last issued command was a read command, or start a new command if the last command was a write command or a control command (BACKGROUND, GO, GO_UNTIL or TRACE1). The ACK pulse is not issued earlier than 32 serial clock cycles after the BDM command was issued. The end of the BDM command is assumed to be the 16th tick of the last bit. This minimum delay assures enough time for the host to perceive the ACK pulse. Note also that, there is no upper limit for the delay between the command and the related ACK pulse, since the command execution depends upon the CPU bus, which in some cases could be very slow due to long accesses taking place.This protocol allows a great flexibility for the POD designers, since it does not rely on any accurate time measurement or short response time to any event in the serial communication.

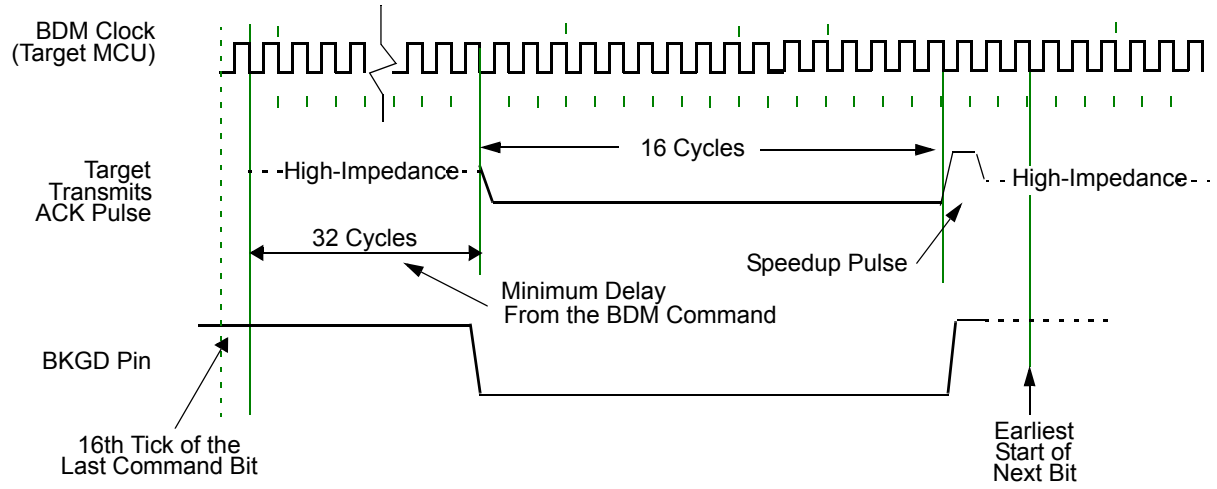

Figure 7-10. Target Acknowledge Pulse (ACK)

NOTE

If the ACK pulse was issued by the target, the host assumes the previous command was executed. If the CPU enters wait or stop prior to executing a hardware command, the ACK pulse will not be issued meaning that the BDM command was not executed. After entering wait or stop mode, the BDM command is no longer pending.

Figure 7-11 shows the ACK handshake protocol in a command level timing diagram. The READ_BYTE instruction is used as an example. First, the 8-bit instruction opcode is sent by the host, followed by the address of the memory location to be read. The target BDM decodes the instruction. A bus cycle is grabbed (free or stolen) by the BDM and it executes the READ_BYTE operation. Having retrieved the data, the BDM issues an ACK pulse to the host controller, indicating that the addressed byte is ready to be retrieved. After detecting the ACK pulse, the host initiates the byte retrieval process. Note that data is sent in the form of a word and the host needs to determine which is the appropriate byte based on whether the address was odd or even.

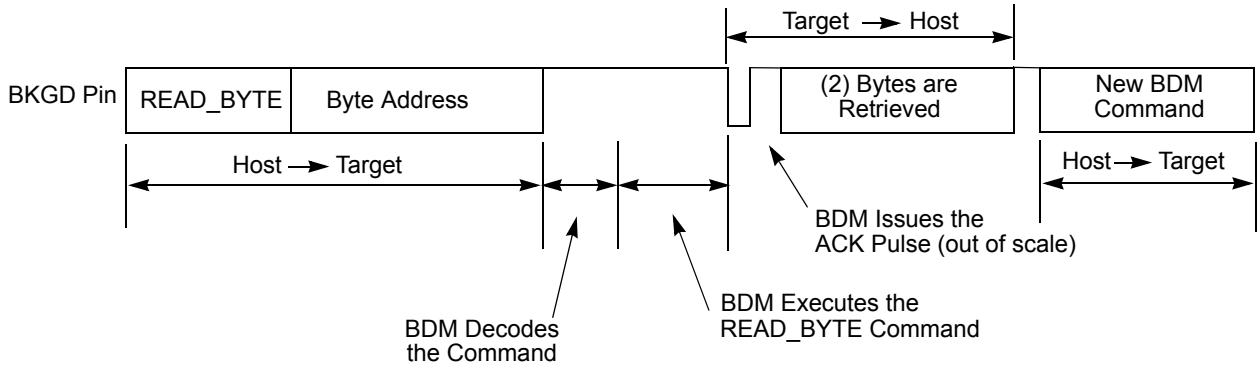

Figure 7-11. Handshake Protocol at Command Level

Differently from the normal bit transfer (where the host initiates the transmission), the serial interface ACK handshake pulse is initiated by the target MCU by issuing a negative edge in the BKGD pin. The hardware handshake protocol in Figure 7-10 specifies the timing when the BKGD pin is being driven, so the host should follow this timing constraint in order to avoid the risk of an electrical conflict in the BKGD pin.

NOTE

The only place the BKGD pin can have an electrical conflict is when one side is driving low and the other side is issuing a speedup pulse (high). Other “highs” are pulled rather than driven. However, at low rates the time of the speedup pulse can become lengthy and so the potential conflict time becomes longer as well.

The ACK handshake protocol does not support nested ACK pulses. If a BDM command is not acknowledge by an ACK pulse, the host needs to abort the pending command first in order to be able to issue a new BDM command. When the CPU enters wait or stop while the host issues a hardware command (e.g., WRITE_BYTE), the target discards the incoming command due to the wait or stop being detected. Therefore, the command is not acknowledged by the target, which means that the ACK pulse will not be issued in this case. After a certain time the host (not aware of stop or wait) should decide to abort any possible pending ACK pulse in order to be sure a new command can be issued. Therefore, the protocol provides a mechanism in which a command, and its corresponding ACK, can be aborted.

11.1.2 Modes of Operation

11.1.2.1 Conversion Modes

There is software programmable selection between performing **single** or **continuous conversion** on a **single channel** or **multiple channels**.

11.1.2.2 MCU Operating Modes

- **Stop Mode**
  Entering Stop Mode aborts any conversion sequence in progress and if a sequence was aborted restarts it after exiting stop mode. This has the same effect/consequences as starting a conversion sequence with write to ATDCTL5. So after exiting from stop mode with a previously aborted sequence all flags are cleared etc.
- **Wait Mode**
  ADC10B8C behaves same in Run and Wait Mode. For reduced power consumption continuous conversions should be aborted before entering Wait mode.
- **Freeze Mode**
  In Freeze Mode the ADC10B8C will either continue or finish or stop converting according to the FRZ1 and FRZ0 bits. This is useful for debugging and emulation.

[1]If only AN0 should be converted use MULT=0.

14.3.2.2 ATD Control Register 1 (ATDCTL1)

Writes to this register will abort current conversion sequence.

Module Base + 0x0001

	7	6	5	4	3	2	1	0
R W	ETRIGSEL	SRES1	SRES0	SMP_DIS	ETRIGCH3	ETRIGCH2	ETRIGCH1	ETRIGCH0
Reset	0	0	1	0	1	1	1	1

Figure 14-4. ATD Control Register 1 (ATDCTL1)

Read: Anytime

Write: Anytime

Table 14-3. ATDCTL1 Field Descriptions

Field	Description
7 ETRIGSEL	**External Trigger Source Select** — This bit selects the external trigger source to be either one of the AD channels or one of the ETRIG3-0 inputs. See device specification for availability and connectivity of ETRIG3-0 inputs. If a particular ETRIG3-0 input option is not available, writing a 1 to ETRISEL only sets the bit but has no effect, this means that one of the AD channels (selected by ETRIGCH3-0) is configured as the source for external trigger. The coding is summarized in Table 14-5.
6–5 SRES[1:0]	**A/D Resolution Select** — These bits select the resolution of A/D conversion results. See Table 14-4 for coding.
4 SMP_DIS	**Discharge Before Sampling Bit** 0 No discharge before sampling. 1 The internal sample capacitor is discharged before sampling the channel. This adds 2 ATD clock cycles to the sampling time. This can help to detect an open circuit instead of measuring the previous sampled channel.
3–0 ETRIGCH[3:0]	**External Trigger Channel Select** — These bits select one of the AD channels or one of the ETRIG3-0 inputs as source for the external trigger. The coding is summarized in Table 14-5.

Table 14-4. A/D Resolution Coding

SRES1	SRES0	A/D Resolution
0	0	8-bit data
0	1	10-bit data
1	0	12-bit data
1	1	Reserved

Table 19-2. PWME Field Descriptions

Note: Bits related to available channels have functional significance. Writing to unavailable bits has no effect. Read from unavailable bits return a zero

Field	Description
7 PWME7	Pulse Width Channel 7 Enable 0 Pulse width channel 7 is disabled. 1 Pulse width channel 7 is enabled. The pulse modulated signal becomes available at PWM output bit 7 when its clock source begins its next cycle.
6 PWME6	Pulse Width Channel 6 Enable 0 Pulse width channel 6 is disabled. 1 Pulse width channel 6 is enabled. The pulse modulated signal becomes available at PWM output bit 6 when its clock source begins its next cycle. If CON67=1, then bit has no effect and PWM output line 6 is disabled.
5 PWME5	Pulse Width Channel 5 Enable 0 Pulse width channel 5 is disabled. 1 Pulse width channel 5 is enabled. The pulse modulated signal becomes available at PWM output bit 5 when its clock source begins its next cycle.
4 PWME4	Pulse Width Channel 4 Enable 0 Pulse width channel 4 is disabled. 1 Pulse width channel 4 is enabled. The pulse modulated signal becomes available at PWM, output bit 4 when its clock source begins its next cycle. If CON45 = 1, then bit has no effect and PWM output line 4 is disabled.
3 PWME3	Pulse Width Channel 3 Enable 0 Pulse width channel 3 is disabled. 1 Pulse width channel 3 is enabled. The pulse modulated signal becomes available at PWM, output bit 3 when its clock source begins its next cycle.
2 PWME2	Pulse Width Channel 2 Enable 0 Pulse width channel 2 is disabled. 1 Pulse width channel 2 is enabled. The pulse modulated signal becomes available at PWM, output bit 2 when its clock source begins its next cycle. If CON23 = 1, then bit has no effect and PWM output line 2 is disabled.
1 PWME1	Pulse Width Channel 1 Enable 0 Pulse width channel 1 is disabled. 1 Pulse width channel 1 is enabled. The pulse modulated signal becomes available at PWM, output bit 1 when its clock source begins its next cycle.
0 PWME0	Pulse Width Channel 0 Enable 0 Pulse width channel 0 is disabled. 1 Pulse width channel 0 is enabled. The pulse modulated signal becomes available at PWM, output bit 0 when its clock source begins its next cycle. If CON01 = 1, then bit has no effect and PWM output line 0 is disabled.

19.3.2.2 PWM Polarity Register (PWMPOL)

The starting polarity of each PWM channel waveform is determined by the associated PPOLx bit in the PWMPOL register. If the polarity bit is one, the PWM channel output is high at the beginning of the cycle and then goes low when the duty count is reached. Conversely, if the polarity bit is zero, the output starts low and then goes high when the duty count is reached.

Clock A is used as an input to an 8-bit down counter. This down counter loads a user programmable scale value from the scale register (PWMSCLA). When the down counter reaches one, a pulse is output and the 8-bit counter is re-loaded. The output signal from this circuit is further divided by two. This gives a greater range with only a slight reduction in granularity. Clock SA equals clock A divided by two times the value in the PWMSCLA register.

NOTE

Clock SA = Clock A / (2 * PWMSCLA)

When PWMSCLA = $00, PWMSCLA value is considered a full scale value of 256. Clock A is thus divided by 512.

Similarly, clock B is used as an input to an 8-bit down counter followed by a divide by two producing clock SB. Thus, clock SB equals clock B divided by two times the value in the PWMSCLB register.

NOTE

Clock SB = Clock B / (2 * PWMSCLB)

When PWMSCLB = $00, PWMSCLB value is considered a full scale value of 256. Clock B is thus divided by 512.

As an example, consider the case in which the user writes $FF into the PWMSCLA register. Clock A for this case will be E (bus clock) divided by 4. A pulse will occur at a rate of once every 255x4 E cycles. Passing this through the divide by two circuit produces a clock signal at an E divided by 2040 rate. Similarly, a value of $01 in the PWMSCLA register when clock A is E divided by 4 will produce a clock at an E divided by 8 rate.

Writing to PWMSCLA or PWMSCLB causes the associated 8-bit down counter to be re-loaded. Otherwise, when changing rates the counter would have to count down to $01 before counting at the proper rate. Forcing the associated counter to re-load the scale register value every time PWMSCLA or PWMSCLB is written prevents this.

NOTE

Writing to the scale registers while channels are operating can cause irregularities in the PWM outputs.

19.4.1.3 Clock Select

Each PWM channel has the capability of selecting one of four clocks, clock A, clock SA, clock B or clock SB. The clock selection is done with the PCLKx control bits in the PWMCLK register and PCLKABx control bits in PWMCLKAB register. For backward compatibility consideration, the reset value of PWMCLK and PWMCLKAB configures following default clock selection.

For channels 0, 1, 4, and 5 the clock choices are clock A.

For channels 2, 3, 6, and 7 the clock choices are clock B.

NOTE

Changing clock control bits while channels are operating can cause irregularities in the PWM outputs.

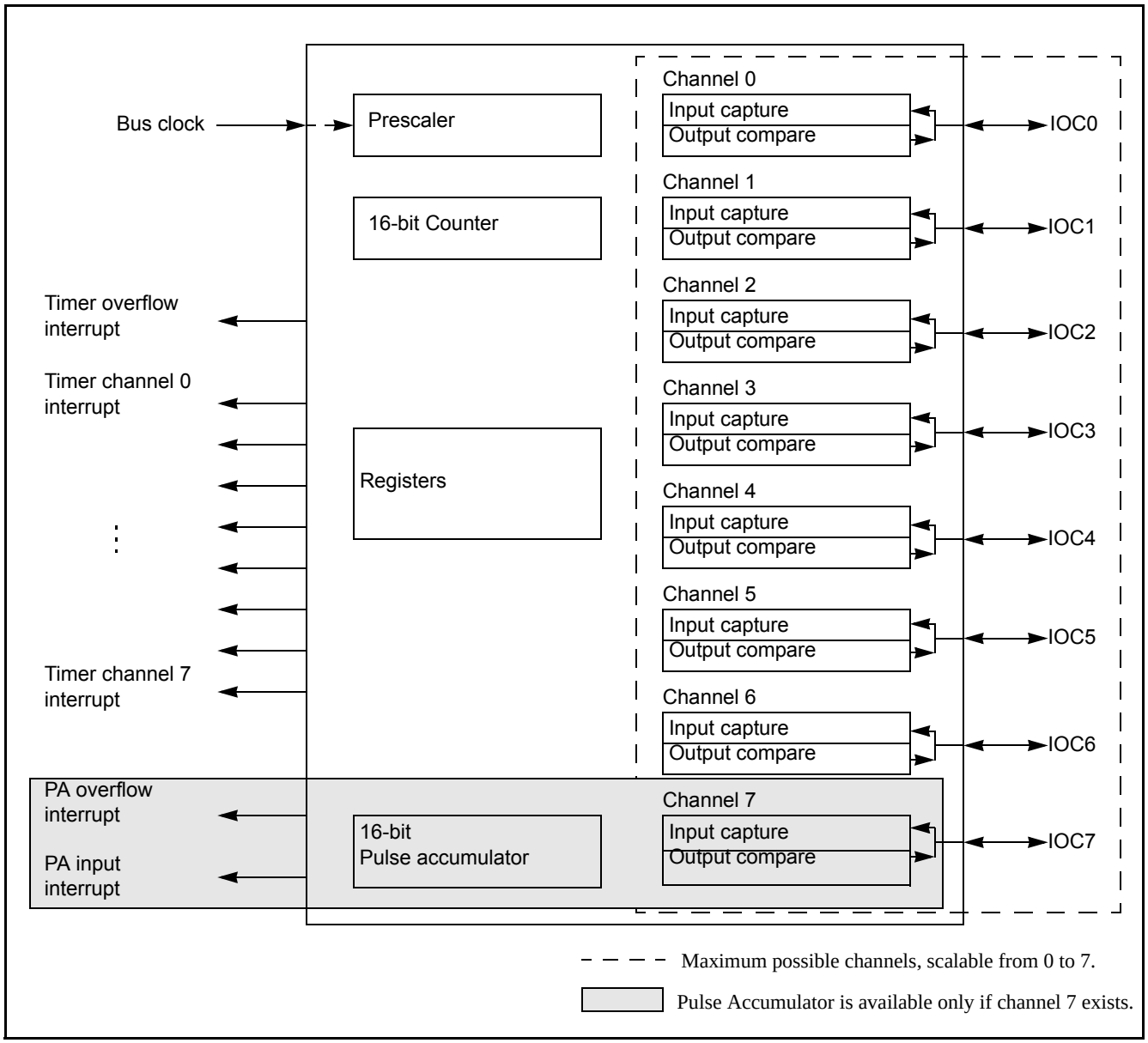

Figure 23-1. TIM16B8CV3 Block Diagram

24.4.6.2 Erase Verify Block Command

The Erase Verify Block command allows the user to verify that an entire P-Flash or EEPROM block has been erased. The FCCOB FlashBlockSelectionCode[1:0] bits determine which block must be verified.

Table 24-31. Erase Verify Block Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x02	Flash block selection code [1:0]. See Table 24-32

Table 24-32. Flash block selection code description

Selection code[1:0]	Flash block to be verified
00	EEPROM
01	Invalid (ACCERR)
10	Invalid (ACCERR)
11	P-Flash

Upon clearing CCIF to launch the Erase Verify Block command, the Memory Controller will verify that the selected P-Flash or EEPROM block is erased. The CCIF flag will set after the Erase Verify Block operation has completed.If the block is not erased, it means blank check failed, both MGSTAT bits will be set.

Table 24-33. Erase Verify Block Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if an invalid FlashBlockSelectionCode[1:0] is supplied[1]
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read[2] or if blank check failed.
	MGSTAT0	Set if any non-correctable errors have been encountered during the read[2] or if blank check failed.

1 As defined by the memory map for FTMRG32K1.

2 As found in the memory map for FTMRG32K1.

24.4.6.3 Erase Verify P-Flash Section Command

The Erase Verify P-Flash Section command will verify that a section of code in the P-Flash memory is erased. The Erase Verify P-Flash Section command defines the starting point of the code to be verified and the number of phrases.

NOTE

User margin levels can be used to check that Flash memory contents have adequate margin for normal level read operations. If unexpected results are encountered when checking Flash memory contents at user margin levels, a potential loss of information has been detected.

24.4.6.13 Set Field Margin Level Command

The Set Field Margin Level command, valid in special modes only, causes the Memory Controller to set the margin level specified for future read operations of the P-Flash or EEPROM block.

Upon clearing CCIF to launch the Set Field Margin Level command, the Memory Controller will set the

Table 24-55. Set Field Margin Level Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0E	Flash block selection code [1:0]. See Table 24-32
001	Margin level setting.	

field margin level for the targeted block and then set the CCIF flag.

NOTE

When the EEPROM block is targeted, the EEPROM field margin levels are applied only to the EEPROM reads. However, when the P-Flash block is targeted, the P-Flash field margin levels are applied to both P-Flash and EEPROM reads. It is not possible to apply field margin levels to the P-Flash block only.

Valid margin level settings for the Set Field Margin Level command are defined in Table 24-56.

Table 24-56. Valid Set Field Margin Level Settings

CCOB (CCOBIX=001)	Level Description
0x0000	Return to Normal Level
0x0001	User Margin-1 Level[1]
0x0002	User Margin-0 Level[2]
0x0003	Field Margin-1 Level[1]
0x0004	Field Margin-0 Level[2]

[1] Read margin to the erased state

[2] Read margin to the programmed state

25.4.6 Flash Command Description

This section provides details of all available Flash commands launched by a command write sequence. The ACCERR bit in the FSTAT register will be set during the command write sequence if any of the following illegal steps are performed, causing the command not to be processed by the Memory Controller:

- Starting any command write sequence that programs or erases Flash memory before initializing the FCLKDIV register
- Writing an invalid command as part of the command write sequence
- For additional possible errors, refer to the error handling table provided for each command

If a Flash block is read during execution of an algorithm (CCIF = 0) on that same block, the read operation will return invalid data if both flags SFDIF and DFDIF are set. If the SFDIF or DFDIF flags were not previously set when the invalid read operation occurred, both the SFDIF and DFDIF flags will be set.

If the ACCERR or FPVIOL bits are set in the FSTAT register, the user must clear these bits before starting any command write sequence (see Section 25.3.2.7).

CAUTION

A Flash word or phrase must be in the erased state before being programmed. Cumulative programming of bits within a Flash word or phrase is not allowed.

25.4.6.1 Erase Verify All Blocks Command

The Erase Verify All Blocks command will verify that all P-Flash and EEPROM blocks have been erased.

Table 25-31. Erase Verify All Blocks Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x01	Not required

Upon clearing CCIF to launch the Erase Verify All Blocks command, the Memory Controller will verify that the entire Flash memory space is erased. The CCIF flag will set after the Erase Verify All Blocks operation has completed. If all blocks are not erased, it means blank check failed, both MGSTAT bits will be set.

Table 25-32. Erase Verify All Blocks Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read[1] or if blank check failed .
	MGSTAT0	Set if any non-correctable errors have been encountered during the read[1] or if blank check failed.

[1] As found in the memory map for FTMRG32K1.

27.3.2.9.1 P-Flash Protection Restrictions

The general guideline is that P-Flash protection can only be added and not removed. Table 27-21 specifies all valid transitions between P-Flash protection scenarios. Any attempt to write an invalid scenario to the FPROT register will be ignored. The contents of the FPROT register reflect the active protection scenario. See the FPHS and FPLS bit descriptions for additional restrictions.

Table 27-21. P-Flash Protection Scenario Transitions

From Protection Scenario	To Protection Scenario[1]							
	0	1	2	3	4	5	6	7
0	X	X	X	X				
1		X		X				
2			X	X				
3				X				
4				X	X			
5			X	X	X	X		
6		X		X	X		X	
7	X	X	X	X	X	X	X	X

1 Allowed transitions marked with X, see Figure 27-14 for a definition of the scenarios.

27.3.2.10 EEPROM Protection Register (EEPROT)

The EEPROT register defines which EEPROM sectors are protected against program and erase operations.

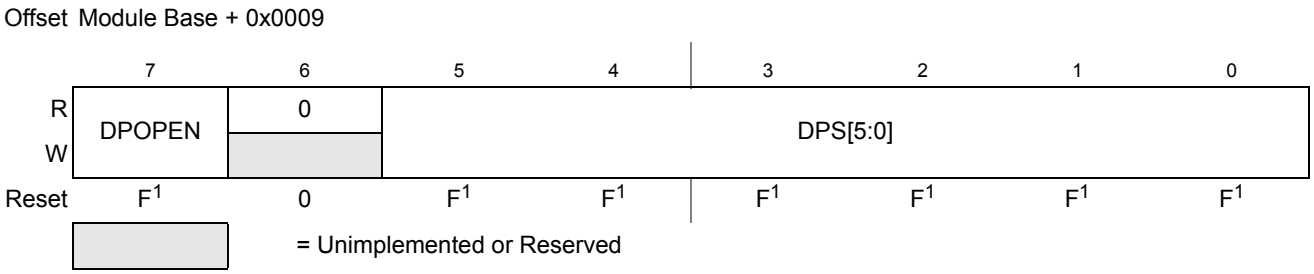

Figure 27-15. EEPROM Protection Register (EEPROT)

1 Loaded from IFR Flash configuration field, during reset sequence.

The (unreserved) bits of the EEPROT register are writable with the restriction that protection can be added but not removed. Writes must increase the DPS value and the DPOPEN bit can only be written from 1 (protection disabled) to 0 (protection enabled). If the DPOPEN bit is set, the state of the DPS bits is irrelevant.

Upon clearing CCIF to launch the Read Once command, a Read Once phrase is fetched and stored in the FCCOB indexed register. The CCIF flag will set after the Read Once operation has completed. Valid phrase index values for the Read Once command range from 0x0000 to 0x0007. During execution of the Read Once command, any attempt to read addresses within P-Flash block will return invalid data.

Table 27-39. Read Once Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if command not available in current mode (see Table 27-27)
		Set if an invalid phrase index is supplied
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read
	MGSTAT0	Set if any non-correctable errors have been encountered during the read

27.4.6.5 Program P-Flash Command

The Program P-Flash operation will program a previously erased phrase in the P-Flash memory using an embedded algorithm.

CAUTION

A P-Flash phrase must be in the erased state before being programmed. Cumulative programming of bits within a Flash phrase is not allowed.

Table 27-40. Program P-Flash Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x06	Global address [17:16] to identify P-Flash block
001	Global address [15:0] of phrase location to be programmed[1]	
010	Word 0 program value	
011	Word 1 program value	
100	Word 2 program value	
101	Word 3 program value	

1 Global address [2:0] must be 000

Upon clearing CCIF to launch the Program P-Flash command, the Memory Controller will program the data words to the supplied global address and will then proceed to verify the data words read back as expected. The CCIF flag will set after the Program P-Flash operation has completed.

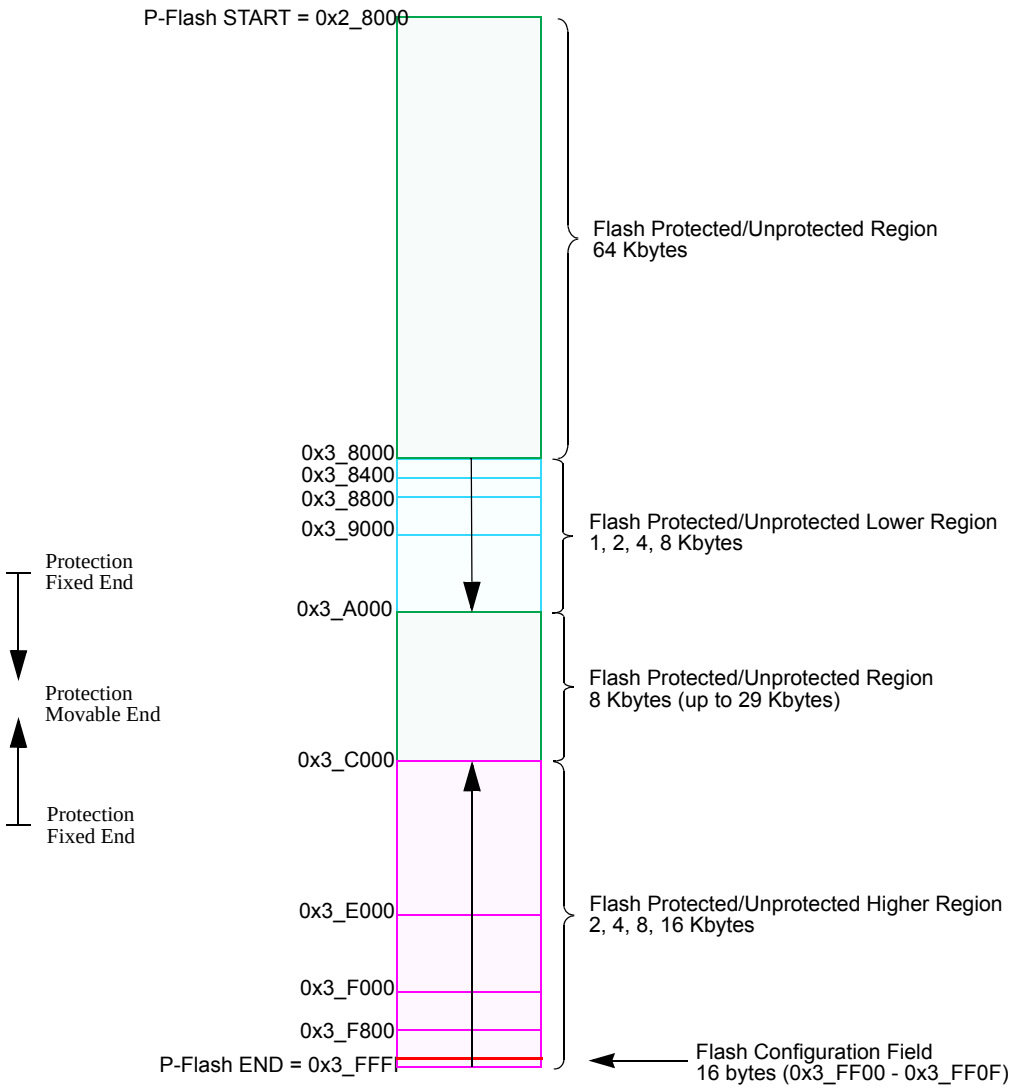

Figure 28-2. P-Flash Memory Map

Table 28-5. Program IFR Fields

Global Address	Size (Bytes)	Field Description
0x0_4000 – 0x0_4007	8	Reserved
0x0_4008 – 0x0_40B5	174	Reserved
0x0_40B6 – 0x0_40B7	2	Version ID[1]
0x0_40B8 – 0x0_40BF	8	Reserved
0x0_40C0 – 0x0_40FF	64	Program Once Field Refer to Section 28.4.6.6, “Program Once Command”

[1] Used to track firmware patch versions, see Section 28.4.2

A summary of the Flash module registers is given in Figure 28-4 with detailed descriptions in the following subsections.

Address & Name		7	6	5	4	3	2	1	0
0x0000 FCLKDIV	R	FDIVLD	FDIVLCK	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
	W								
0x0001 FSEC	R	KEYEN1	KEYEN0	RNV5	RNV4	RNV3	RNV2	SEC1	SEC0
	W								
0x0002 FCCOBIX	R	0	0	0	0	0	CCOBIX2	CCOBIX1	CCOBIX0
	W								
0x0003 FRSV0	R	0	0	0	0	0	0	0	0
	W								
0x0004 FCNFG	R	CCIE	0	0	IGNSF	0	0	FDFD	FSFD
	W								
0x0005 FERCNFG	R	0	0	0	0	0	0	DFDIE	SFDIE
	W								
0x0006 FSTAT	R	CCIF	0	ACCERR	FPVIOL	MGBUSY	RSVD	MGSTAT1	MGSTAT0
	W								
0x0007 FERSTAT	R	0	0	0	0	0	0	DFDIF	SFDIF
	W								
0x0008 FPROT	R	FPOPEN	RNV6	FPHDIS	FPHS1	FPHS0	FPLDIS	FPLS1	FPLS0
	W								
0x0009 EEPROT	R	DPOPEN	DPS6	DPS5	DPS4	DPS3	DPS2	DPS1	DPS0
	W								
0x000A FCCOBHI	R	CCOB15	CCOB14	CCOB13	CCOB12	CCOB11	CCOB10	CCOB9	CCOB8
	W								
0x000B FCCOBLO	R	CCOB7	CCOB6	CCOB5	CCOB4	CCOB3	CCOB2	CCOB1	CCOB0
	W								
0x000C FRSV1	R	0	0	0	0	0	0	0	0
	W								

Figure 28-4. FTMRG96K1 Register Summary

Table 29-49. Erase P-Flash Sector Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if command not available in current mode (see Table 29-27)
		Set if an invalid global address [17:16] is supplied (see Table 29-3)
		Set if a misaligned phrase address is supplied (global address [2:0] != 000)
	FPVIOL	Set if the selected P-Flash sector is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

29.4.6.10 Unsecure Flash Command

The Unsecure Flash command will erase the entire P-Flash and EEPROM memory space and, if the erase is successful, will release security.

Table 29-50. Unsecure Flash Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0B	Not required

Upon clearing CCIF to launch the Unsecure Flash command, the Memory Controller will erase the entire P-Flash and EEPROM memory space and verify that it is erased. If the Memory Controller verifies that the entire Flash memory space was properly erased, security will be released. If the erase verify is not successful, the Unsecure Flash operation sets MGSTAT1 and terminates without changing the security state. During the execution of this command (CCIF=0) the user must not write to any Flash module register. The CCIF flag is set after the Unsecure Flash operation has completed.

Table 29-51. Unsecure Flash Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if command not available in current mode (see Table 29-27)
	FPVIOL	Set if any area of the P-Flash or EEPROM memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

29.4.6.11 Verify Backdoor Access Key Command

The Verify Backdoor Access Key command will only execute if it is enabled by the KEYEN bits in the FSEC register (see Table 29-10). The Verify Backdoor Access Key command releases security if user-supplied keys match those stored in the Flash security bytes of the Flash configuration field (see

Upon clearing CCIF to launch the Set User Margin Level command, the Memory Controller will set the user margin level for the targeted block and then set the CCIF flag.

NOTE

When the EEPROM block is targeted, the EEPROM user margin levels are applied only to the EEPROM reads. However, when the P-Flash block is targeted, the P-Flash user margin levels are applied to both P-Flash and EEPROM reads. It is not possible to apply user margin levels to the P-Flash block only.

Valid margin level settings for the Set User Margin Level command are defined in Table 30-55.

Table 30-55. Valid Set User Margin Level Settings

CCOB (CCOBIX=001)	Level Description
0x0000	Return to Normal Level
0x0001	User Margin-1 Level[1]
0x0002	User Margin-0 Level[2]

[1] Read margin to the erased state

[2] Read margin to the programmed state

Table 30-56. Set User Margin Level Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch.
		Set if command not available in current mode (see Table 30-27).
		Set if an invalid margin level setting is supplied.
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

NOTE

User margin levels can be used to check that Flash memory contents have adequate margin for normal level read operations. If unexpected results are encountered when checking Flash memory contents at user margin levels, a potential loss of information has been detected.

30.4.6.13 Set Field Margin Level Command

The Set Field Margin Level command, valid in special modes only, causes the Memory Controller to set the margin level specified for future read operations of the P-Flash or EEPROM block.

Table A-29. ADC Conversion Performance 5V range, RVA enabled

Supply voltage V_{DDA} =5.0 V, -40°C < T_J < 150°C. V_{RH} = 5.0V. f_{ADCCLK} = 0.25 .. 2MHz [1] The values are tested to be valid with no port AD/C output drivers switching simultaneous with conversions.								
Num	**C**	**Rating**		**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	P	Resolution	12-Bit	LSB		0.61		mV
2	P	Differential Nonlinearity	12-Bit	DNL		±3	±4	counts
3	P	Integral Nonlinearity	12-Bit	INL		±3.5	±5	counts
4	C	Absolute Error[2]	12-Bit	AE			±8	counts
5	P	internal VRH reference voltage	LQFP48, LQFP64, LQFP100	Vvrh_int	4.495		4.505	V
			KGD	Vvrh_int	4.490		4.510	V
6	P	internal VRL reference voltage	LQFP48, LQFP64, LQFP100	Vvrh_int	1.995		2.005V	V
			KGD	Vvrl_int	1.990		2.010V	V
7	C	VRH_INT drift vs temperature[3]		Vvrh_drift	-2		2	mV
8	C	VRL_INT drift vs temperature		Vvrl_drift	-2.5		2.5	mV
9	C	rva turn on settling time		$t_{settling\_on}$			2.5	µs
10	C	rva turn off settling time		$t_{settling\_off}$			1	µs

1 Upper limit of f_{ADCCLK} is restricted when RVA attenuation mode is engaged.

2 These values include the quantization error which is inherently 1/2 count for any A/D converter and the error of the internally generated reference values..

3 Please note: although different in value, drift of vrh_int and vrl_int will go in the same direction.