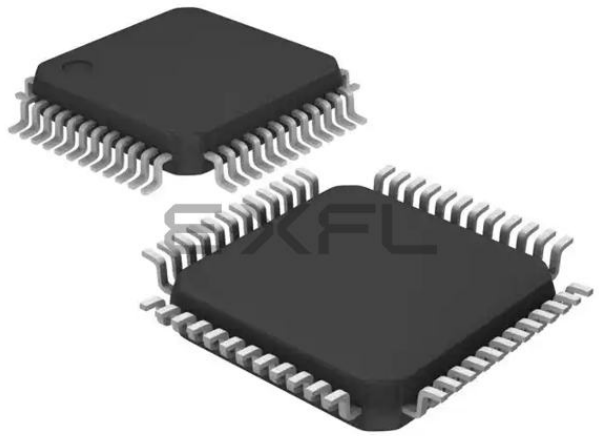

Welcome to E-XFL.COM

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	S12Z
Core Size	16-Bit
Speed	32MHz
Connectivity	CANbus, I²C, IrDA, LINbus, SCI, SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	34
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	5.5V ~ 18V
Data Converters	A/D 10x10b; D/A 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 150°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvla12f0wlfr

Chapter 2
Port Integration Module (S12ZVLPIMV2)

Table 1-6. Pin Summary

LQFP / QFN[2]		QFN[1]		Function						Power Supply	Internal Pull Resistor	
48	32	32	Pin	1st Func.	2nd Func.	3rd Func.	4th Func.	5th Func.	6th Func.		CTRL	Reset State
1	1	1	VSUP	—	—	—	—	—	—	V_{SUP}	—	—
2	2	2	PL0	HVI0	KWL0	—	—	—	—	V_{DDX}	—	—
3	3	3	PE0	ETRIG0	EXTAL	—	—	—	—	V_{DDX}	PERE/ PPSE	Down
4	4	4	PE1	XTAL	—	—	—	—	—	V_{DDX}	PERE/ PPSE	Down
5	—	—	PAD9	KWAD9	AN9	—	—	—	—	V_{DDA}	PERADH/ PPSADH	Off
6	—	—	PAD8	KWAD8	AN8	AMPM[2]	—	—	—	V_{DDA}	PERADH/ PPSADH	Off
7	—	—	PAD7	KWAD7	AN7	AMPP[(2)]	—	—	—	V_{DDA}	PERADL/ PPSADL	Off
8	—	—	PAD6	KWAD6	AN6	AMP[(2)]	—	—	—	V_{DDA}	PERADL/ PPSADL	Off
9	5	5	PAD5	KWAD5	ETRIG0	AN5	—	—	—	V_{DDA}	PERADL/ PPSADL	Off
10	6	6	PAD4	KWAD4	AN4	ACMP_1[(2)]	—	—	—	V_{DDA}	PERADL/ PPSADL	Off
11	7	7	PAD3	KWAD3	AN3	PGA_IN1[(2)]	—	—	—	V_{DDA}	PERADL/ PPSADL	Off
12	8	8	PAD2	KWAD2	AN2	PGA_IN0[(2)]	—	—	—	V_{DDA}	PERADL/ PPSADL	Off
13	9	9	PAD1	KWAD1	AN1	VRL[3]	PGA_REF1[(2)]	—	—	V_{DDA}	PERADL/ PPSADL	Off
14	10	10	VSSA	—	—	—	—	—	—	—	—	—
15	11	11	VDDA	—	—	—	—	—	—	V_{DDA}	—	—
16	12	12	PAD0	KWAD0	AN0	VRH	ACMP_0[(2)]	—	—	V_{DDA}	PERADL/ PPSADL	Off
17	—	—	PT6	IOC1_0	—	—	—	—	—	V_{DDX}	PERT/ PPST	Off
18	—	—	PP6	KWP[6]	ETRIG0	PWM6	—	—	—	V_{DDX}	PERP/ PPSP	Off
19	—	—	PT4	IOC0_4	RXD1	—	—	—	—	V_{DDX}	PERT/ PPST	Off
20	—	—	PT5	IOC0_5	TXD1	—	—	—	—	V_{DDX}	PERT/ PPST	Off
21	13	13	TEST	—	—	—	—	—	—	—	$\overline{\text{RESET}}$	Down
22	14	14	PT0	IOC0_0	SDA0	RXD1	PWM2	LPTXD0	—	V_{DDX}	PERT/ PPST	Off

Table 1-6. Pin Summary

LQFP / QFN[2]		QFN[1]		Function						Power Supply	Internal Pull Resistor	
48	32	32	Pin	1st Func.	2nd Func.	3rd Func.	4th Func.	5th Func.	6th Func.		CTRL	Reset State
23	15	15	BCTL	—	—	—	—	—	—	—	—	—
24	16	16	LIN	—	—	—	—	—	—	—	—	—
25	17	17	LGND	—	—	—	—	—	—	LGND	—	—
26	18	18	BKGD	MODC	—	—	—	—	—	V_{DDX}		Up
27	19	19	PT1	IOC0_1	SCL0	TXD1	PWM0	LPRXD0	—	V_{DDX}	PERT/ PPST	Off
28	20	20	PS0	KWS0	MISO0	PWM4	RXD0	IOC0_2	RXCAN0[4]	V_{DDX}	PERS/ PPSS	Up
29	—	—	PT7	IOC1_1	—	—	—	—	—	V_{DDX}	PERT/ PPST	Off
30	—	—	PP0	KWP0	PWM0	—	—	—	—	V_{DDX}	PERP/ PPSP	Off
31	—	—	PP2	KWP2	PWM2	—	—	—	—	V_{DDX}	PERP/ PPSP	Off
32	—	—	PP4	KWP4	PWM4	—	—	—	—	V_{DDX}	PERP/ PPSP	Off
33	21	21	PS1	KWS1	MOSI0	PWM6	TXD0 LPDC0	IOC0_3	—	V_{DDX}	PERS/ PPSS	Up
34	22	22	PS2	KWS2	SCK0	IOC0_4	DBGEEV	TXCAN0[4]	—	V_{DDX}	PERS/ PPSS	Up
35	23	23	PS3	KWS3	$\overline{SS0}$	IOC0_5	ECLK	—	—	V_{DDX}	PERS/ PPSS	Up
36	24	24	$\overline{RESET}$	—	—	—	—	—	—	V_{DDX}	TEST pin	Up
37	25	25	PP3[5]	$\overline{IRQ}$	KWP3	PWM3	—	—	—	V_{DDX}	PERP/ PPSP	Off
38	—	26	VSSX2	—	—	—	—	—	—	V_{DDX}	—	—
39	26	27	PP5[5]	$\overline{XIRQ}$	KWP5	PWM5	—	—	—	V_{DDX}	PERP/ PPSP	Off
40	27	—	PT2	IOC0_2	ACMPO[3]	—	—	—	—	V_{DDX}	PERT/ PPST	Off
41	—	—	PJ0	SDA0	PWM5	RXCAN0[4]	—	—	—	V_{DDX}	PERTJ/ PPSJ	Up
42	—	—	PJ1	SCL0	PWM7	TXCAN0[4]	—	—	—	V_{DDX}	PERJ/ PPSJ	Up
43	—	—	PT3	IOC0_3	—	—	—	—	——	V_{DDX}	PERT/ PPST	Off
44	28	28	PP7[6]	KWP7	PWM7	IOC1_0	—	—	—	V_{DDX}	PERP/ PPSP	Off
45	29	29	VDDX	—	—	—	—	—	—	V_{DDX}	—	—
46	30	30	VSSX1	—	—	—	—	—	—	V_{SSX}	—	—

NOTE

The term stop mode (STOP) is limited to voltage regulator operating in reduced performance mode (RPM). Refer to “Low Power Modes” section in device overview.

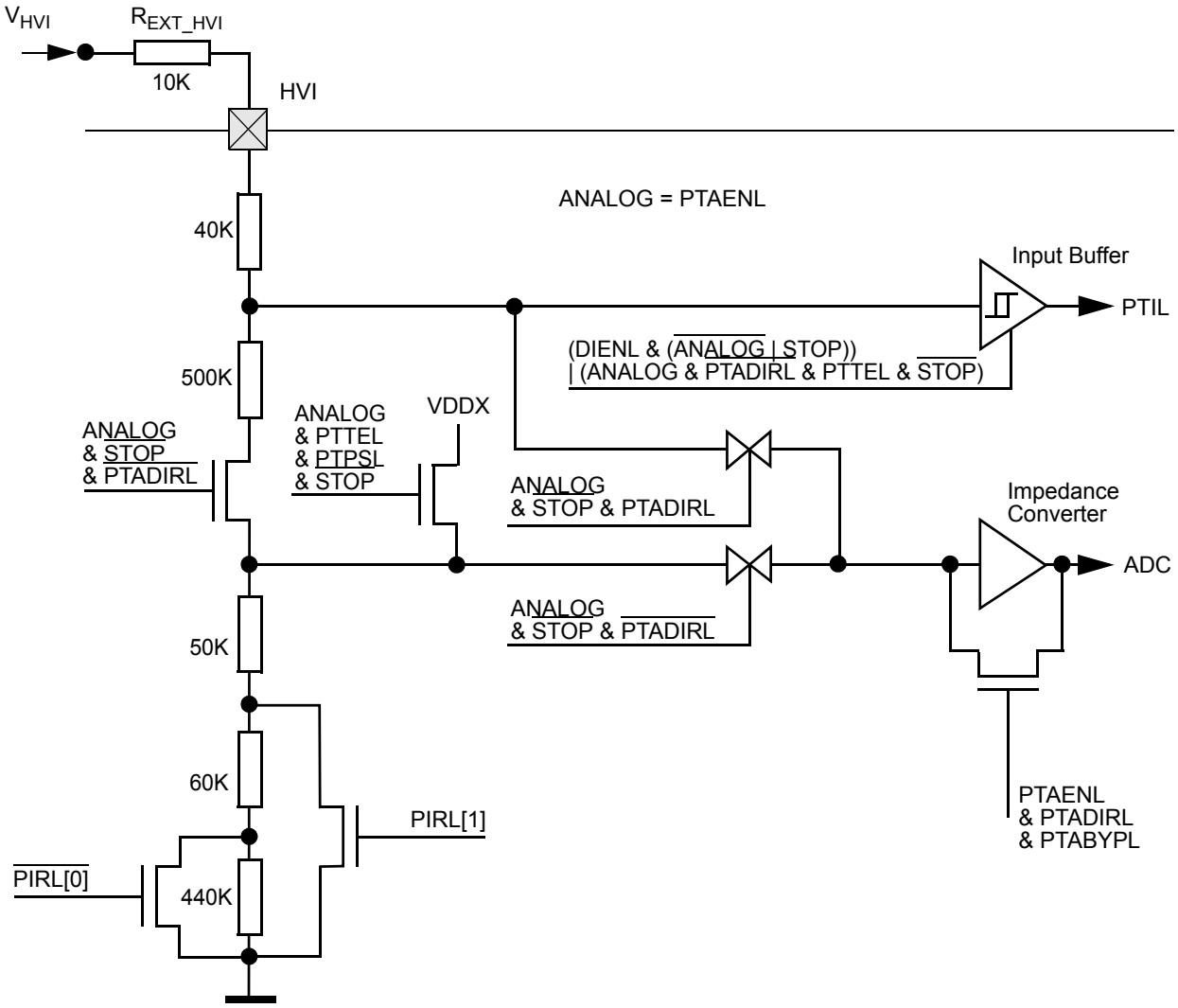

Figure 2-32. HVI Block Diagram

Voltages up to V_{HVI} can be applied to the HVI pin. Internal voltage dividers scale the input signals down to logic level. There are two modes, digital and analog, where these signals can be processed.

2.4.5.1 Digital Mode Operation

In digital mode (PTAENL=0) the input buffer is enabled if DIENL=1. The synchronized pin input state determined at threshold level V_{TH_HVI} can be read in register PTIL. Interrupt flag (PIFL) is set on input

4.3.2.2 Error Code Register (MMCECH, MMCECL)

Address: 0x0080 (MMCECH)

	7	6	5	4	3	2	1	0
R W	ITR[3:0]				TGT[3:0]			
Reset	0	0	0	0	0	0	0	0

Address: 0x0081 (MMCECL)

	7	6	5	4	3	2	1	0
R W	ACC[3:0]				ERR[3:0]			
Reset	0	0	0	0	0	0	0	0

Figure 4-5. Error Code Register (MMCEC)

Read: Anytime

Write: Write of 0xFFFF to MMCECH:MMCECL resets both registers to 0x0000

Table 4-5. MMCECH and MMCECL Field Descriptions

Field	Description
7-4 (MMCECH) ITR[3:0]	**Initiator Field** — The ITR[3:0] bits capture the initiator which caused the access violation. The initiator is captured in form of a 4 bit value which is assigned as follows: 0:none (no error condition detected) 1:S12ZCPU 2:reserved 3:ADC 4-15: reserved
3-0 (MMCECH) TGT[3:0]	**Target Field** — The TGT[3:0] bits capture the target of the faulty access. The target is captured in form of a 4 bit value which is assigned as follows: 0:none 1:register space 2:RAM 3:EEPROM 4:program flash 5:IFR 6-15: reserved

Similarly the STEP1 command issued from a WAI instruction cannot be completed by the CPU until the CPU leaves wait mode due to an interrupt. The first STEP1 into wait mode sets the BDCCSR WAIT bit.

If the part is still in Wait mode and a further STEP1 is carried out then the NORESP and ILLCMD bits are set because the device is no longer in active BDM for the duration of WAI execution.

5.1.4 Block Diagram

A block diagram of the BDC is shown in Figure 5-1.

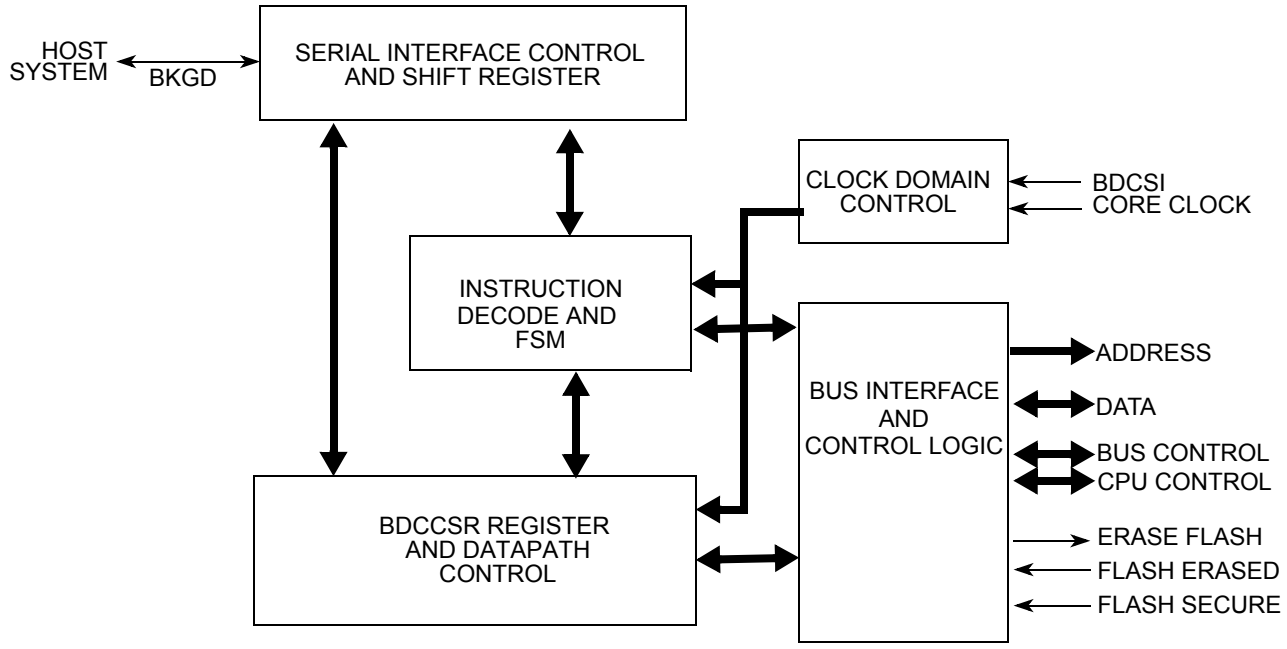

Figure 5-1. BDC Block Diagram

5.2 External Signal Description

A single-wire interface pin (BKGD) is used to communicate with the BDC system. During reset, this pin is a device mode select input. After reset, this pin becomes the dedicated serial interface pin for the BDC.

BKGD is a pseudo-open-drain pin with an on-chip pull-up. Unlike typical open-drain pins, the external RC time constant on this pin due to external capacitance, plays almost no role in signal rise time. The custom protocol provides for brief, actively driven speed-up pulses to force rapid rise times on this pin without risking harmful drive level conflicts. Refer to Section 5.4.6, “BDC Serial Interface” for more details.

The ACK handshake protocol does not support nested ACK pulses. If a BDC command is not acknowledged by an ACK pulse, the host needs to abort the pending command first in order to be able to issue a new BDC command. The host can 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.

Commands With-Status do not generate an ACK, thus if ACK is enabled and a With-Status command is issued, the host must use the 512 cycle timeout to calculate when the data is ready for retrieval.

5.4.7.1 Long-ACK Hardware Handshake Protocol

If a command results in an error condition, whereby a BDCCSRL flag is set, then the target generates a “Long-ACK” low pulse of 64 BDCSI clock cycles, followed by a brief speed pulse. This indicates to the host that an error has occurred. The host can subsequently read BDCCSR to determine the type of error. Whether normal ACK or Long-ACK, the ACK pulse is not issued earlier than 32 BDCSI clock cycles after the BDC command was issued. The end of the BDC command is assumed to be the 16th BDCSI clock cycle of the last bit. The 32 cycle minimum delay differs from the 16 cycle delay time with ACK disabled.

If a BDC access request does not gain access within 512 core clock cycles, the request is aborted, the NORESP flag is set and a Long-ACK pulse is transmitted to indicate an error case.

Following a STOP or WAI instruction, if the BDC is enabled, the first ACK, following stop or wait mode entry is a long ACK to indicate an exception.

5.4.8 Hardware Handshake Abort Procedure

The abort procedure is based on the SYNC command. To abort a command that has not responded with an ACK pulse, the host controller generates a sync request (by driving BKGD low for at least 128 BDCSI clock cycles and then driving it high for one BDCSI clock cycle as a speedup pulse). By detecting this long low pulse in the BKGD pin, the target executes the SYNC protocol, see Section 5.4.4.1, “SYNC”, and assumes that the pending command and therefore the related ACK pulse are being aborted. After the SYNC protocol has been completed the host is free to issue new BDC commands.

The host can issue a SYNC close to the 128 clock cycles length, providing a small overhead on the pulse length to assure the sync pulse is not misinterpreted by the target. See Section 5.4.4.1, “SYNC”.

Figure 5-11 shows a SYNC command being issued after a READ_MEM, which aborts the READ_MEM command. Note that, after the command is aborted a new command is issued by the host.

Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x00001B	INT_CFDATA3	R	0	0	0	0	0	PRIOLVL[2:0]		
		W								
0x00001C	INT_CFDATA4	R	0	0	0	0	0	PRIOLVL[2:0]		
		W								
0x00001D	INT_CFDATA5	R	0	0	0	0	0	PRIOLVL[2:0]		
		W								
0x00001E	INT_CFDATA6	R	0	0	0	0	0	PRIOLVL[2:0]		
		W								
0x00001F	INT_CFDATA7	R	0	0	0	0	0	PRIOLVL[2:0]		
		W								

= Unimplemented or Reserved

Figure 6-2. S12ZINTV0 Register Summary

6.3.2.1 Interrupt Vector Base Register (IVBR)

Address: 0x000010

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IVB_ADDR[15:1]															0
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Figure 6-3. Interrupt Vector Base Register (IVBR)

Read: Anytime

Write: Anytime

Table 6-4. IVBR Field Descriptions

Field	Description
15–1 IVB_ADDR [15:1]	**Interrupt Vector Base Address Bits** — These bits represent the upper 15 bits of all vector addresses. Out of reset these bits are set to 0xFFFE (i.e., vectors are located at 0xFFFE00–0xFFFFFF). **Note:** A system reset will initialize the interrupt vector base register with “0xFFFE” before it is used to determine the reset vector address. Therefore, changing the IVBR has no effect on the location of the reset vector (0xFFFFFC–0xFFFFFF).

8.2.2.7 ECC Debug Command (ECCDCMD)

Module Base + 0x000F — Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	ECCDRR	0	0	0	0	0	ECCDW	ECCDR
W								
Reset	0	0	0	0	0	0	0	0

[1] Read: Anytime
Write: Anytime, in special mode only

Figure 8-8. ECC Debug Command (ECCDCMD)

Table 8-8. ECCDCMD Field Description

Field	Description
7 ECCDRR	**ECC Disable Read Repair Function**— Writing one to this register bit will disable the automatic single bit ECC error repair function during read access; see also chapter 8.3.7, “ECC Debug Behavior”. 0 Automatic single ECC error repair function is enabled 1 Automatic single ECC error repair function is disabled
1 ECCDW	**ECC Debug Write Command** — Writing one to this register bit will perform a debug write access, to the system memory. During this access the debug data word (DDATA) and the debug ECC value (DECC) will be written to the system memory address defined by DPTR. If the debug write access is done, this bit is cleared. Writing 0 has no effect**.** It is not possible to set this bit if the previous debug access is ongoing (ECCDW or ECCDR bit set).
0 ECCDR	**ECC Debug Read Command** — Writing one to this register bit will perform a debug read access from the system memory address defined by DPTR. If the debug read access is done, this bit is cleared and the raw memory read data are available in register DDATA and the raw ECC value is available in register DECC. Writing 0 has no effect. If the ECCDW and ECCDR bit are set at the same time, then only the ECCDW bit is set and the Debug Write Command is performed. It is not possible to set this bit if the previous debug access is ongoing (ECCDW or ECCDR bit set).

8.3 Functional Description

Depending on the system integration the max memory access width can be 4 byte, but the ECC value is generated based on an aligned 2 byte data word. Depending on the access type, the access is separated into different access cycles. Table 8-9 shows the different access types with the expected number of access cycles and the performed internal operations.

Table 8-9. Memory access cycles

Access type	ECC error	access cycle	Internal operation	Memory content	Error indication
Aligned write	—	1	write to memory	new data	—

9.3.2.17 Low Voltage Control Register (CPMULVCTL)

The CPMULVCTL register allows the configuration of the low-voltage detect features.

Module Base + 0x0011

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	LVDS	LVIE	LVIF
W								
Reset	0	0	0	0	0	U	0	U

The Reset state of LVDS and LVIF depends on the external supplied VDDA level

= Unimplemented or Reserved

Figure 9-21. Low Voltage Control Register (CPMULVCTL)

Read: Anytime

Write: LVIE and LVIF are write anytime, LVDS is read only

Table 9-18. CPMULVCTL Field Descriptions

Field	Description
2 LVDS	**Low-Voltage Detect Status Bit** — This read-only status bit reflects the voltage level on VDDA. Writes have no effect. 0 Input voltage VDDA is above level V_{LVID} or RPM. 1 Input voltage VDDA is below level V_{LVIA} and FPM.
1 LVIE	**Low-Voltage Interrupt Enable Bit** 0 Interrupt request is disabled. 1 Interrupt will be requested whenever LVIF is set.
0 LVIF	**Low-Voltage Interrupt Flag** — LVIF is set to 1 when LVDS status bit changes. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (LVIE = 1), LVIF causes an interrupt request. 0 No change in LVDS bit. 1 LVDS bit has changed.

9.3.2.18 Autonomous Periodical Interrupt Control Register (CPMUAPICTL)

The CPMUAPICTL register allows the configuration of the autonomous periodical interrupt features.

Module Base + 0x0012

	7	6	5	4	3	2	1	0
R	APICLK	0	0	APIES	APIEA	APIFE	APIE	APIF
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 9-22. Autonomous Periodical Interrupt Control Register (CPMUAPICTL)

Read: Anytime

Write: Anytime

Table 9-19. CPMUAPICTL Field Descriptions

Field	Description
7 APICLK	**Autonomous Periodical Interrupt Clock Select Bit** — Selects the clock source for the API. Writable only if APIFE = 0. APICLK cannot be changed if APIFE is set by the same write operation. 0 Autonomous Clock (ACLK) used as source. 1 Bus Clock used as source.
4 APIES	**Autonomous Periodical Interrupt External Select Bit** — Selects the waveform at the external pin API_EXTCLK as shown in Figure 9-23. See device level specification for connectivity of API_EXTCLK pin. 0 If APIEA and APIFE are set, at the external pin API_EXTCLK periodic high pulses are visible at the end of every selected period with the size of half of the minimum period (APIR=0x0000 in Table 9-23). 1 If APIEA and APIFE are set, at the external pin API_EXTCLK a clock is visible with 2 times the selected API Period.
3 APIEA	**Autonomous Periodical Interrupt External Access Enable Bit** — If set, the waveform selected by bit APIES can be accessed externally. See device level specification for connectivity. 0 Waveform selected by APIES can not be accessed externally. 1 Waveform selected by APIES can be accessed externally, if APIFE is set.
2 APIFE	**Autonomous Periodical Interrupt Feature Enable Bit** — Enables the API feature and starts the API timer when set. 0 Autonomous periodical interrupt is disabled. 1 Autonomous periodical interrupt is enabled and timer starts running.
1 APIE	**Autonomous Periodical Interrupt Enable Bit** 0 API interrupt request is disabled. 1 API interrupt will be requested whenever APIF is set.
0 APIF	**Autonomous Periodical Interrupt Flag** — APIF is set to 1 when the in the API configured time has elapsed. This flag can only be cleared by writing a 1.Writing a 0 has no effect. If enabled (APIE = 1), APIF causes an interrupt request. 0 API time-out has not yet occurred. 1 API time-out has occurred.

10.5.2.6 ADC Conversion Flow Control Register (ADCFLWCTL)

Bit set and bit clear instructions should not be used to access this register.

When the ADC is enabled the bits of ADCFLWCTL register can be modified after a latency time of three Bus Clock cycles.
All bits are cleared if bit ADC_EN is clear or via ADC soft-reset.

Module Base + 0x0005

	7	6	5	4	3	2	1	0
R	SEQA	TRIG	RSTA	LDOK	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 10-9. ADC Conversion Flow Control Register (ADCFLWCTL)

Read: Anytime

Write:

- Bits SEQA, TRIG, RSTA, LDOK can only be set if bit ADC_EN is set.
- Writing 1’b0 to any of these bits does not have an effect

Timing considerations (Trigger Event - channel sample start) depending on ADC mode configuration:

- Restart Mode
  When the Restart Event has been processed (initial command of current CSL is loaded) it takes two Bus Clock cycles plus two ADC conversion clock cycles (pump phase) from the Trigger Event (bit TRIG set) until the select channel starts to sample.
  During a conversion sequence (back to back conversions) it takes five Bus Clock cycles plus two ADC conversion clock cycles (pump phase) from current conversion period end until the newly selected channel is sampled in the following conversion period.
- Trigger Mode
  When a Restart Event occurs a Trigger Event is issued simultaneously. The time required to process the Restart Event is mainly defined by the internal read data bus availability and therefore can vary. In this mode the Trigger Event is processed immediately after the Restart Event is finished and both conversion flow control bits are cleared simultaneously. From de-assert of bit TRIG until sampling begins five Bus Clock cycles are required. Hence from occurrence of a Restart Event until channel sampling it takes five Bus Clock cycles plus an uncertainty of a few Bus Clock cycles.

For more details regarding the sample phase please refer to Section 10.6.2.2, “Sample and Hold Machine with Sample Buffer Amplifier.

negative amplifier input is open. The unbuffered analog voltage from the DAC resistor network is available on the internal connection DACI. For decoding of the control signals see Table 11-7.

11.5.7 Mode “Buffered DAC”

The “Buffered DAC” mode is selected by DACCTL.DACM[2:0] = 0x7. During this is mode the DAC resistor network and the operational amplifier are enabled. The analog output voltage from the DAC resistor network output is buffered by the operational amplifier and is available on the AMP output pin.

The DAC resistor network output is disconnected from the DACU pin. The unbuffered analog voltage from the DAC resistor network is available on the internal connection DACI. For the decoding of the control signals see Table 11-7.

11.5.8 Analog output voltage calculation

The DAC can provide an analog output voltage in two different voltage ranges:

- FVR = 0, reduced voltage range

  The DAC generates an analog output voltage inside the range from 0.1 x (VRH - VRL) + VRL to 0.9 x (VRH-VRL) + VRL with a resolution ((VRH-VRL) x 0.8) / 256, see equation below:

analog output voltage = VOLATGE[7:0] x ((VRH-VRL) x 0.8) / 256) + 0.1 x (VRH-VRL) + VRL ***Eqn. 11-1***

- FVR = 1, full voltage range

  The DAC generates an analog output voltage inside the range from VRL to VRH with a resolution (VRH-VRL) / 256, see equation below:

analog output voltage = VOLTAGE[7:0] x (VRH-VRL) / 256 +VRL ***Eqn. 11-2***

See Table 11-8 for an example for VRL = 0.0 V and VRH = 5.0 V.

Table 11-8. Analog output voltage calculation

FVR	min. voltage	max. voltage	Resolution	Equation
0	0.5V	4.484V	15.625mV	**VOLTAGE[7:0] x (4.0V) / 256) + 0.5V**
1	0.0V	4.980V	19.531mV	**VOLTAGE[7:0] x (5.0V) / 256**

Table 13-3. CANCTL0 Register Field Descriptions (continued)

Field	Description
1 SLPRQ[4]	**Sleep Mode Request** — This bit requests the MSCAN to enter sleep mode, which is an internal power saving mode (see Section 13.4.5.5, “MSCAN Sleep Mode”). The sleep mode request is serviced when the CAN bus is idle, i.e., the module is not receiving a message and all transmit buffers are empty. The module indicates entry to sleep mode by setting SLPAK = 1 (see Section 13.3.2.2, “MSCAN Control Register 1 (CANCTL1)”). SLPRQ cannot be set while the WUPIF flag is set (see Section 13.3.2.5, “MSCAN Receiver Flag Register (CANRFLG)”). Sleep mode will be active until SLPRQ is cleared by the CPU or, depending on the setting of WUPE, the MSCAN detects activity on the CAN bus and clears SLPRQ itself. 0 Running — The MSCAN functions normally 1 Sleep mode request — The MSCAN enters sleep mode when CAN bus idle
0 INITRQ[5,6]	**Initialization Mode Request** — When this bit is set by the CPU, the MSCAN skips to initialization mode (see Section 13.4.4.5, “MSCAN Initialization Mode”). Any ongoing transmission or reception is aborted and synchronization to the CAN bus is lost. The module indicates entry to initialization mode by setting INITAK = 1 (Section 13.3.2.2, “MSCAN Control Register 1 (CANCTL1)”). The following registers enter their hard reset state and restore their default values: CANCTL0[7], CANRFLG[8], CANRIER[9], CANTFLG, CANTIER, CANTARQ, CANTAAK, and CANTBSEL. The registers CANCTL1, CANBTR0, CANBTR1, CANIDAC, CANIDAR0-7, and CANIDMR0-7 can only be written by the CPU when the MSCAN is in initialization mode (INITRQ = 1 and INITAK = 1). The values of the error counters are not affected by initialization mode. When this bit is cleared by the CPU, the MSCAN restarts and then tries to synchronize to the CAN bus. If the MSCAN is not in bus-off state, it synchronizes after 11 consecutive recessive bits on the CAN bus; if the MSCAN is in bus-off state, it continues to wait for 128 occurrences of 11 consecutive recessive bits. Writing to other bits in CANCTL0, CANRFLG, CANRIER, CANTFLG, or CANTIER must be done only after initialization mode is exited, which is INITRQ = 0 and INITAK = 0. 0 Normal operation 1 MSCAN in initialization mode

1 See the Bosch CAN 2.0A/B specification for a detailed definition of transmitter and receiver states.

2 In order to protect from accidentally violating the CAN protocol, TXCAN is immediately forced to a recessive state when the CPU enters wait (CSWAI = 1) or stop mode (see Section 13.4.5.2, “Operation in Wait Mode” and Section 13.4.5.3, “Operation in Stop Mode”).

3 The CPU has to make sure that the WUPE register and the WUPIE wake-up interrupt enable register (see Section 13.3.2.6, “MSCAN Receiver Interrupt Enable Register (CANRIER)) is enabled, if the recovery mechanism from stop or wait is required.

4 The CPU cannot clear SLPRQ before the MSCAN has entered sleep mode (SLPRQ = 1 and SLPAK = 1).

5 The CPU cannot clear INITRQ before the MSCAN has entered initialization mode (INITRQ = 1 and INITAK = 1).

6 In order to protect from accidentally violating the CAN protocol, TXCAN is immediately forced to a recessive state when the initialization mode is requested by the CPU. Thus, the recommended procedure is to bring the MSCAN into sleep mode (SLPRQ = 1 and SLPAK = 1) before requesting initialization mode.

7 Not including WUPE, INITRQ, and SLPRQ.

8 TSTAT1 and TSTAT0 are not affected by initialization mode.

9 RSTAT1 and RSTAT0 are not affected by initialization mode.

13.3.2.2 MSCAN Control Register 1 (CANCTL1)

The CANCTL1 register provides various control bits and handshake status information of the MSCAN module as described below.

The comparator outputs BVLC and BVHC are forced to zero if the comparator is disabled (configuration bit BSUSE is cleared). If the software disables the comparator during a high or low Voltage condition (BVHC or BVLC active), then an additional interrupt is generated. To avoid this behavior the software must disable the interrupt generation before disabling the comparator.

The BATS interrupt vector is named in Table 14-6. Vector addresses and interrupt priorities are defined at MCU level.

The module internal interrupt sources are combined into one module interrupt signal.

Table 14-6. BATSV3 Interrupt Sources

Module Interrupt Source	Module Internal Interrupt Source	Local Enable
BATSV3 Interrupt (BATI)	BATS Voltage Low Condition Interrupt (BVLI)	BVLIE = 1
	BATS Voltage High Condition Interrupt (BVHI)	BVHIE = 1

14.4.2.1 BATS Voltage Low Condition Interrupt (BVLI)

To use the Voltage Low Interrupt the Level Sensing must be enabled (BSUSE =1).

If measured when

a) V_{LBI1} selected with BVLS[1:0] = 0x0
$V_{measure} < V_{LBI1\_A}$ (falling edge) or $V_{measure} < V_{LBI1\_D}$ (rising edge)

or when

b) V_{LBI2} selected with BVLS[1:0] = 0x1 at pin VSUP
$V_{measure} < V_{LBI2\_A}$ (falling edge) or $V_{measure} < V_{LBI2\_D}$ (rising edge)

or when

c) V_{LBI3} selected with BVLS[1:0] = 0x2
$V_{measure} < V_{LBI3\_A}$ (falling edge) or $V_{measure} < V_{LBI3\_D}$ (rising edge)

or when

d) V_{LBI4} selected with BVLS[1:0] = 0x3
$V_{measure} < V_{LBI4\_A}$ (falling edge) or $V_{measure} < V_{LBI4\_D}$ (rising edge)

then BVLC is set. BVLC status bit indicates that a low voltage at pin VSUP is present. The Low Voltage Interrupt flag (BVLIF) is set to 1 when the Voltage Low Condition (BVLC) changes state . The Interrupt flag BVLIF can only be cleared by writing a 1. If the interrupt is enabled by bit BVLIE the module requests an interrupt to MCU (BATI).

14.4.2.2 BATS Voltage High Condition Interrupt (BVHI)

To use the Voltage High Interrupt the Level Sensing must be enabled (BSUSE=1).

Module Base + 0x0005

	7	6	5	4	3	2	1	0
R W	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0
Reset	0	0	0	0	0	0	0	0

Figure 16-7. Timer Count Register Low (TCNTL)

The 16-bit main timer is an up counter.

A full access for the counter register should take place in one clock cycle. A separate read/write for high byte and low byte will give a different result than accessing them as a word.

Read: Anytime

Write: Has no meaning or effect in the normal mode; only writable in special modes .

The period of the first count after a write to the TCNT registers may be a different size because the write is not synchronized with the prescaler clock.

16.3.2.4 Timer System Control Register 1 (TSCR1)

Module Base + 0x0006

	7	6	5	4	3	2	1	0
R	TEN	TSWAI	TSFRZ	TFFCA	PRNT	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 16-8. Timer System Control Register 1 (TSCR1)

Read: Anytime

Write: Anytime

Table 16-4. TSCR1 Field Descriptions

Field	Description
7 TEN	Timer Enable 0 Disables the main timer, including the counter. Can be used for reducing power consumption. 1 Allows the timer to function normally. If for any reason the timer is not active, there is no ÷64 clock for the pulse accumulator because the ÷64 is generated by the timer prescaler.
6 TSWAI	Timer Module Stops While in Wait 0 Allows the timer module to continue running during wait. 1 Disables the timer module when the MCU is in the wait mode. Timer interrupts cannot be used to get the MCU out of wait. TSWAI also affects pulse accumulator.

Module Base + 0x0001

	7	6	5	4	3	2	1	0
R W	PPOL7	PPOL6	PPOL5	PPOL4	PPOL3	PPOL2	PPOL1	PPOL0
Reset	0	0	0	0	0	0	0	0

Figure 17-4. PWM Polarity Register (PWMPOL)

Read: Anytime

Write: Anytime

NOTE

PPOLx register bits can be written anytime. If the polarity is changed while a PWM signal is being generated, a truncated or stretched pulse can occur during the transition

Table 17-3. PWMPOL 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
7–0 PPOL[7:0]	**Pulse Width Channel 7–0 Polarity Bits** 0 PWM channel 7–0 outputs are low at the beginning of the period, then go high when the duty count is reached. 1 PWM channel 7–0 outputs are high at the beginning of the period, then go low when the duty count is reached.

17.3.2.3 PWM Clock Select Register (PWMCLK)

Each PWM channel has a choice of four clocks to use as the clock source for that channel as described below.

Module Base + 0x0002

	7	6	5	4	3	2	1	0
R W	PCLK7	PCLKL6	PCLK5	PCLK4	PCLK3	PCLK2	PCLK1	PCLK0
Reset	0	0	0	0	0	0	0	0

Figure 17-5. PWM Clock Select Register (PWMCLK)

Read: Anytime

Write: Anytime

NOTE

Register bits PCLK0 to PCLK7 can be written anytime. If a clock select is changed while a PWM signal is being generated, a truncated or stretched pulse can occur during the transition.

19.2.2 MISO — Master In/Slave Out Pin

This pin is used to transmit data out of the SPI module when it is configured as a slave and receive data when it is configured as master.

19.2.3 $\overline{SS}$ — Slave Select Pin

This pin is used to output the select signal from the SPI module to another peripheral with which a data transfer is to take place when it is configured as a master and it is used as an input to receive the slave select signal when the SPI is configured as slave.

19.2.4 SCK — Serial Clock Pin

In master mode, this is the synchronous output clock. In slave mode, this is the synchronous input clock.

19.3 Memory Map and Register Definition

This section provides a detailed description of address space and registers used by the SPI.

19.3.1 Module Memory Map

The memory map for the S12SPIV5 is given in Figure 19-2. The address listed for each register is the sum of a base address and an address offset. The base address is defined at the SoC level and the address offset is defined at the module level. Reads from the reserved bits return zeros and writes to the reserved bits have no effect.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 SPICR1	R	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
	W								
0x0001 SPICR2	R	0	XFRW	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
	W								
0x0002 SPIBR	R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
	W								
0x0003 SPISR	R	SPIF	0	SPTEF	MODF	0	0	0	0
	W								
0x0004 SPIDRH	R	R15	R14	R13	R12	R11	R10	R9	R8
	W	T15	T14	T13	T12	T11	T10	T9	T8
0x0005 SPIDRL	R	R7	R6	R5	R4	R3	R2	R1	R0
	W	T7	T6	T5	T4	T3	T2	T1	T0
0x0006 Reserved	R								
	W								

= Unimplemented or Reserved

Figure 19-2. SPI Register Summary

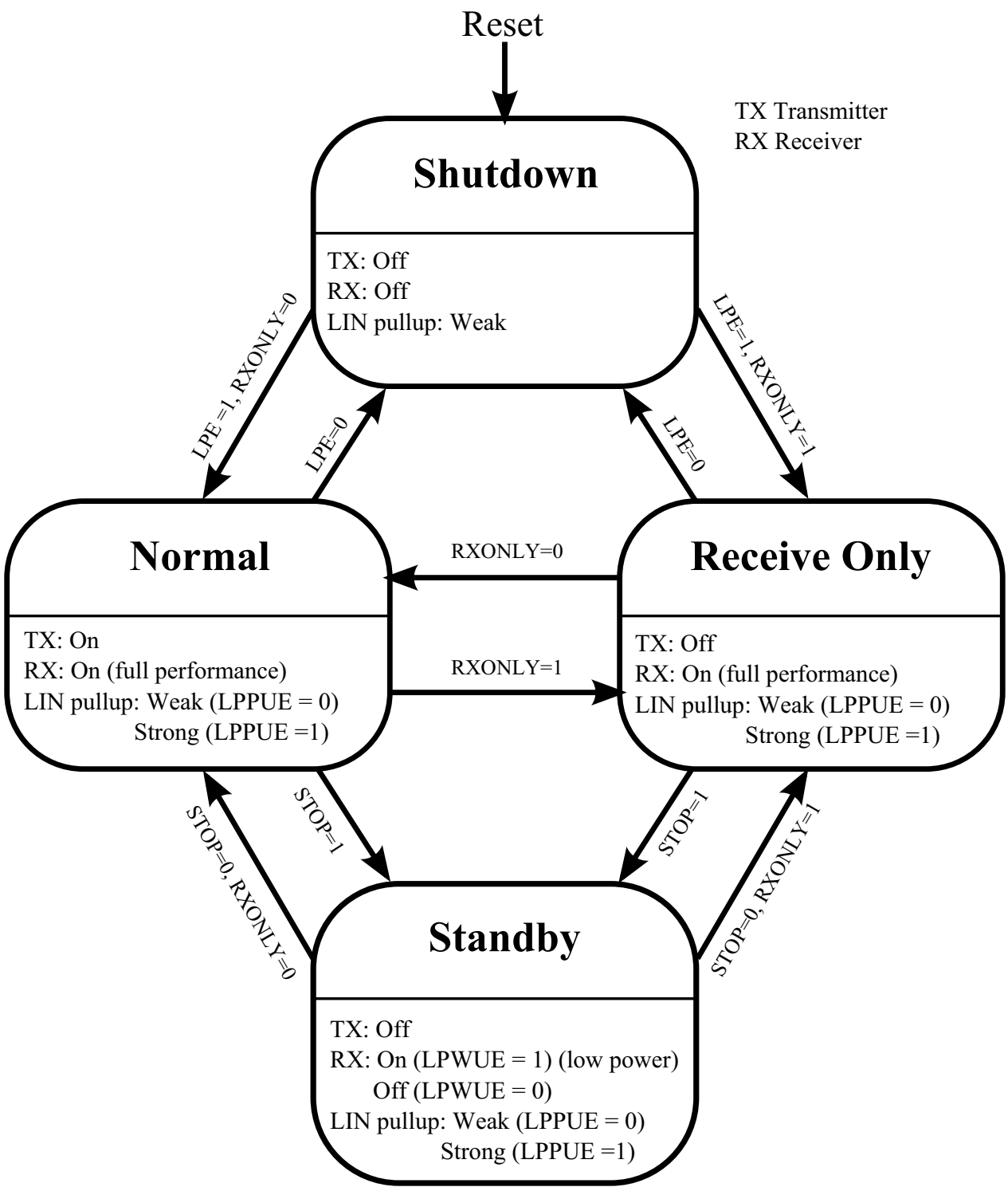

Figure 21-11. LIN Physical Layer Mode Transitions

O.5 0x0200-0x037F PIM (continued)

Global Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0210–0x025F	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0260	PTE	R	0	0	0	0	0	0	PTE1	PTE0
		W								
0x0261	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0262	PTIE	R	0	0	0	0	0	0	PTIE1	PTIE0
		W								
0x0263	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0264	DDRE	R	0	0	0	0	0	0	DDRE1	DDRE0
		W								
0x0265	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0266	PERE	R	0	0	0	0	0	0	PERE1	PERE0
		W								
0x0267	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0268	PPSE	R	0	0	0	0	0	0	PPSE1	PPSE0
		W								
0x0269–0x027F	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0280	PTADH	R	0	0	0	0	0	0	PTADH1	PTADH0
		W								
0x0281	PTADL	R	PTADL7	PTADL6	PTADL5	PTADL4	PTADL3	PTADL2	PTADL1	PTADL0
		W								
0x0282	PTIADH	R	0	0	0	0	0	0	PTIADH1	PTIADH0
		W								
0x0283	PTIADL	R	PTIADL7	PTIADL6	PTIADL5	PTIADL4	PTIADL3	PTIADL2	PTIADL1	PTIADL0
		W								
0x0284	DDRADH	R	0	0	0	0	0	0	DDRADH1	DDRADH0
		W								
0x0285	DDRADL	R	DDRADL7	DDRADL6	DDRADL5	DDRADL4	DDRADL3	DDRADL2	DDRADL1	DDRADL0
		W								