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	HCS12X
Core Size	16-Bit
Speed	40MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	59
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12xs256j0vaar

- INT (interrupt module)
  - Seven levels of nested interrupts
  - Flexible assignment of interrupt sources to each interrupt level.
  - External non-maskable high priority interrupt (XIRQ)
  - The following inputs can act as Wake-up Interrupts
    - IRQ and non-maskable XIRQ
    - CAN receive pins
    - SCI receive pins
    - Depending on the package option up to 20 pins on ports J, H and P configurable as rising or falling edge sensitive
- MMC (module mapping control)
- DBG (debug module)
  - Monitoring of CPU bus with tag-type or force-type breakpoint requests
  - 64 x 64-bit circular trace buffer captures change-of-flow or memory access information
- BDM (background debug mode)
- OSC_LCP (oscillator)
  - Low power loop control Pierce oscillator utilizing a 4MHz to 16MHz crystal
  - Good noise immunity
  - Full-swing Pierce option utilizing a 2MHz to 40MHz crystal
  - Transconductance sized for optimum start-up margin for typical crystals
- IPLL (Internally filtered, frequency modulated phase-locked-loop clock generation)
  - No external components required
  - Configurable option to spread spectrum for reduced EMC radiation (frequency modulation)
- CRG (clock and reset generation)
  - COP watchdog
  - Real time interrupt
  - Clock monitor
  - Fast wake up from STOP in self clock mode
- Memory Options
  - 64, 128 and 256 Kbyte Flash
  - Flash General Features
    - 64 data bits plus 8 syndrome ECC (Error Correction Code) bits allow single bit failure correction and double fault detection
    - Erase sector size 1024 bytes
    - Automated program and erase algorithm
    - Protection scheme to prevent accidental program or erase
    - Security option to prevent unauthorized access
    - Sense-amp margin level setting for reads
  - 4 and 8 Kbyte Data Flash space

Table 2-1. Pin Functions and Priorities (continued)

Port	Pin Name	Pin Function & Priority[1]	I/O	Description	Pin Function after Reset
P	PP7	PWM7	I/O	Pulse Width Modulator channel 7; emergency shut-down	GPIO
		GPIO/KWP7	I/O	General purpose; with interrupt	
	PP[6:3]	PWM[6:3]	O	Pulse Width Modulator channel 6 - 3	
		GPIO/KWP[6:3]	I/O	General purpose; with interrupt	
	PP2	PWM2	O	Pulse Width Modulator channel 2	
		(IOC2)	I/O	Timer Channel 2	
		(TXD1)	O	Serial Communication Interface 1 transmit pin	
		GPIO/KWP2	I/O	General purpose; with interrupt	
	PP1	PWM1	O	Pulse Width Modulator channel 1	
		(IOC1)	I/O	Timer Channel 1	
		GPIO/KWP1	I/O	General purpose; with interrupt	
	PP0	PWM0	O	Pulse Width Modulator channel 0	
		(IOC0)	I/O	Timer Channel 0	
		(RXD1)	I	Serial Communication Interface 1 receive pin	
		GPIO/KWP0	I/O	General purpose; with interrupt	
H	PH[7:0]	GPIO/KWH[7:0]	I/O	General purpose; with interrupt	GPIO
J	PJ[7:6]	GPIO/KWJ[7:6]	I/O	General purpose; with interrupt	GPIO
	PJ[1:0]	GPIO/KWJ[1:0]	I/O	General purpose; with interrupt	
AD	PAD[15:0]	GPIO	I/O	General purpose	GPIO
		AN[15:0]	I	ATD analog	

1 Signals in brackets denote alternative module routing pins.

2 Function active when $\overline{\text{RESET}}$ asserted.

2.3 Memory Map and Register Definition

This section provides a detailed description of all Port Integration Module registers.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0265 PPSH	R	PPSH7	PPSH6	PPSH5	PPSH4	PPSH3	PPSH2	PPSH1	PPSH0
	W								
0x0266 PIEH	R	PIEH7	PIEH6	PIEH5	PIEH4	PIEH3	PIEH2	PIEH1	PIEH0
	W								
0x0267 PIFH	R	PIFH7	PIFH6	PIFH5	PIFH4	PIFH3	PIFH2	PIFH1	PIFH0
	W								
0x0268 PTJ	R	PTJ7	PTJ6	0	0	0	0	PTJ1	PTJ0
	W								
0x0269 PTIJ	R	PTIJ7	PTIJ6	0	0	0	0	PTIJ1	PTIJ0
	W								
0x026A DDRJ	R	DDRJ7	DDRJ6	0	0	0	0	DDRJ1	DDRJ0
	W								
0x026B RDRJ	R	RDRJ7	RDRJ6	0	0	0	0	RDRJ1	RDRJ0
	W								
0x026C PERJ	R	PERJ7	PERJ6	0	0	0	0	PERJ1	PERJ0
	W								
0x026D PPSJ	R	PPSJ7	PPSJ6	0	0	0	0	PPSJ1	PPSJ0
	W								
0x026E PIEJ	R	PIEJ7	PIEJ6	0	0	0	0	PIEJ1	PIEJ0
	W								
0x026F PIFJ	R	PIFJ7	PIFJ6	0	0	0	0	PIFJ1	PIFJ0
	W								
0x0270 PT0AD0	R	PT0AD07	PT0AD06	PT0AD05	PT0AD04	PT0AD03	PT0AD02	PT0AD01	PT0AD00
	W								
0x0271 PT1AD0	R	PT1AD07	PT1AD06	PT1AD05	PT1AD04	PT1AD03	PT1AD02	PT1AD01	PT1AD00
	W								
0x0272 DDR0AD0	R	DDR0AD07	DDR0AD06	DDR0AD05	DDR0AD04	DDR0AD03	DDR0AD02	DDR0AD01	DDR0AD00
	W								
0x0273 DDR1AD0	R	DDR1AD07	DDR1AD06	DDR1AD05	DDR1AD04	DDR1AD03	DDR1AD02	DDR1AD01	DDR1AD00
	W								
0x0274 RDR0AD0	R	RDR0AD07	RDR0AD06	RDR0AD05	RDR0AD04	RDR0AD03	RDR0AD02	RDR0AD01	RDR0AD00
	W								

= Unimplemented or Reserved

Port P pins PP[7:3] can be used for either general purpose I/O with pin interrupt capability, or with the PWM or with the channels of the standard Timer.subsystem.

Port P pins PP[2,0] can be used for either general purpose I/O, or with the PWM or with the TIM or with the SCI1 subsystem.

Port P pin PP[1] can be used for either general purpose I/O, or with the PWM or with the TIM subsystem.

2.4.3.9 Port H

Port H pins PH[7:0] can be used for general purpose I/O with pin interrupt capability.

2.4.3.10 Port J

Port J pins PJ[7,6,1,0] can be used for general purpose I/O with pin-interrupt capability.

2.4.3.11 Port AD

This port is associated with the ATD.

Port AD pins PAD[15:0] can be used for either general purpose I/O, or with the ATD0 subsystem.

2.4.4 Pin interrupts

Ports P, H and J offer pin interrupt capability. The interrupt enable as well as the sensitivity to rising or falling edges can be individually configured on a per-pin basis. All bits/pins in a port share the same interrupt vector. Interrupts can be used with the pins configured as inputs or outputs.

An interrupt is generated when a bit in the port interrupt flag register and its corresponding port interrupt enable bit are both set. The pin interrupt feature is also capable to wake up the CPU when it is in STOP or WAIT mode.

A digital filter on each pin prevents pulses (Figure 2-75) shorter than a specified time from generating an interrupt. The minimum time varies over process conditions, temperature and voltage (Figure 2-74 and Table 2-72).

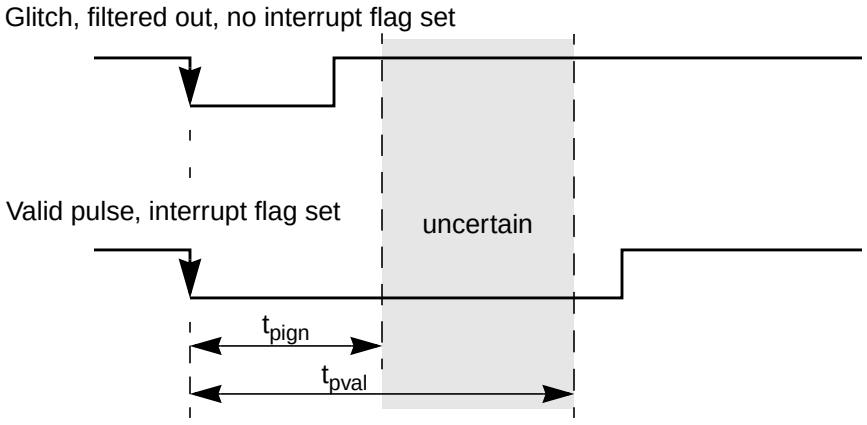


Figure 2-74. Interrupt Glitch Filter on Port P, H and J (PPS=0)

within a few percent of the actual target speed and the communication protocol can easily tolerate speed errors of several percent.

As soon as the SYNC request is detected by the target, any partially received command or bit retrieved is discarded. This is referred to as a soft-reset, equivalent to a time-out in the serial communication. After the SYNC response, the target will consider the next negative edge (issued by the host) as the start of a new BDM command or the start of new SYNC request.

Another use of the SYNC command pulse is to abort a pending ACK pulse. The behavior is exactly the same as in a regular SYNC command. Note that one of the possible causes for a command to not be acknowledged by the target is a host-target synchronization problem. In this case, the command may not have been understood by the target and so an ACK response pulse will not be issued.

5.4.10 Instruction Tracing

When a TRACE1 command is issued to the BDM in active BDM, the CPU exits the standard BDM firmware and executes a single instruction in the user code. Once this has occurred, the CPU is forced to return to the standard BDM firmware and the BDM is active and ready to receive a new command. If the TRACE1 command is issued again, the next user instruction will be executed. This facilitates stepping or tracing through the user code one instruction at a time.

If an interrupt is pending when a TRACE1 command is issued, the interrupt stacking operation occurs but no user instruction is executed. Once back in standard BDM firmware execution, the program counter points to the first instruction in the interrupt service routine.

Be aware when tracing through the user code that the execution of the user code is done step by step but all peripherals are free running. Hence possible timing relations between CPU code execution and occurrence of events of other peripherals no longer exist.

Do not trace the CPU instruction BGND used for soft breakpoints. Tracing the BGND instruction will result in a return address pointing to BDM firmware address space.

When tracing through user code which contains stop or wait instructions the following will happen when the stop or wait instruction is traced:

The CPU enters stop or wait mode and the TRACE1 command can not be finished before leaving the low power mode. This is the case because BDM active mode can not be entered after CPU executed the stop instruction. However all BDM hardware commands except the BACKGROUND command are operational after tracing a stop or wait instruction and still being in stop or wait mode. If system stop mode is entered (all bus masters are in stop mode) no BDM command is operational.

As soon as stop or wait mode is exited the CPU enters BDM active mode and the saved PC value points to the entry of the corresponding interrupt service routine.

In case the handshake feature is enabled the corresponding ACK pulse of the TRACE1 command will be discarded when tracing a stop or wait instruction. Hence there is no ACK pulse when BDM active mode is entered as part of the TRACE1 command after CPU exited from stop or wait mode. All valid commands sent during CPU being in stop or wait mode or after CPU exited from stop or wait mode will have an ACK pulse. The handshake feature becomes disabled only when system stop mode has been reached. Hence

6.1.5 Block Diagram

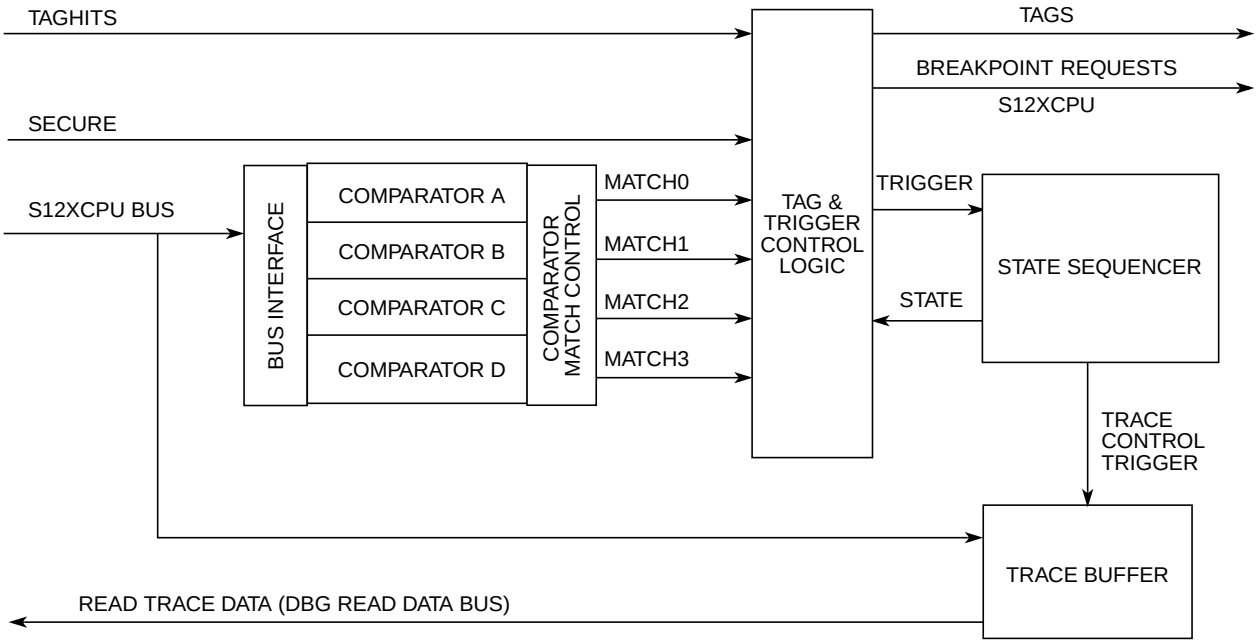


Figure 6-1. Debug Module Block Diagram

6.2 External Signal Description

The S12XDBG sub-module features no external signals.

6.3 Memory Map and Registers

6.3.1 Module Memory Map

A summary of the registers associated with the S12XDBG sub-block is shown in Table 6-2. Detailed descriptions of the registers and bits are given in the subsections that follow.

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0020	DBGC1	R	ARM	0	reserved	BDM	DBGBRK	reserved	COMRV	
		W		TRIG						
0x0021	DBGSR	R	TBF	0	0	0	0	SSF2	SSF1	SSF0
		W								
0x0022	DBGTCR	R	reserved	TSOURCE	TRANGE		TRCMOD		TALIGN	
		W								
0x0023	DBGC2	R	0	0	0	0	CDCM		ABCM	
		W								

Figure 6-2. Quick Reference to S12XDBG Registers

from the EXTAL input frequency. In full stop mode (PSTP = 0), the EXTAL pin is pulled down by an internal resistor of typical 200 kΩ.

NOTE

Freescale recommends an evaluation of the application board and chosen resonator or crystal by the resonator or crystal supplier.

Loop controlled circuit is not suited for overtone resonators and crystals.

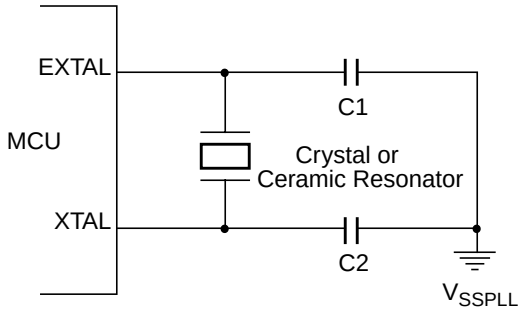
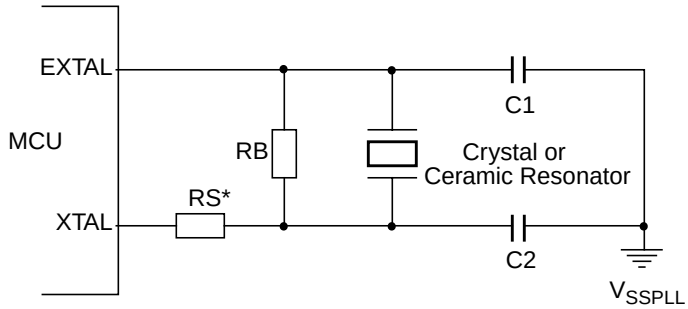


Figure 9-2. Loop Controlled Pierce Oscillator Connections (LCP mode selected)

NOTE

Full swing Pierce circuit is not suited for overtone resonators and crystals without a careful component selection.



* R_S can be zero (shorted) when use with higher frequency crystals. Refer to manufacturer's data.

Figure 9-3. Full Swing Pierce Oscillator Connections (FSP mode selected)

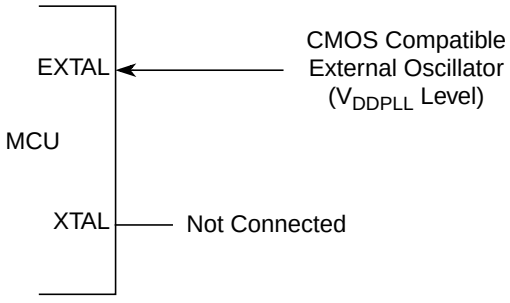


Figure 9-4. External Clock Connections (FSP mode selected)

11.4 Functional Description

11.4.1 General

This section provides a complete functional description of the MSCAN.

11.4.2 Message Storage

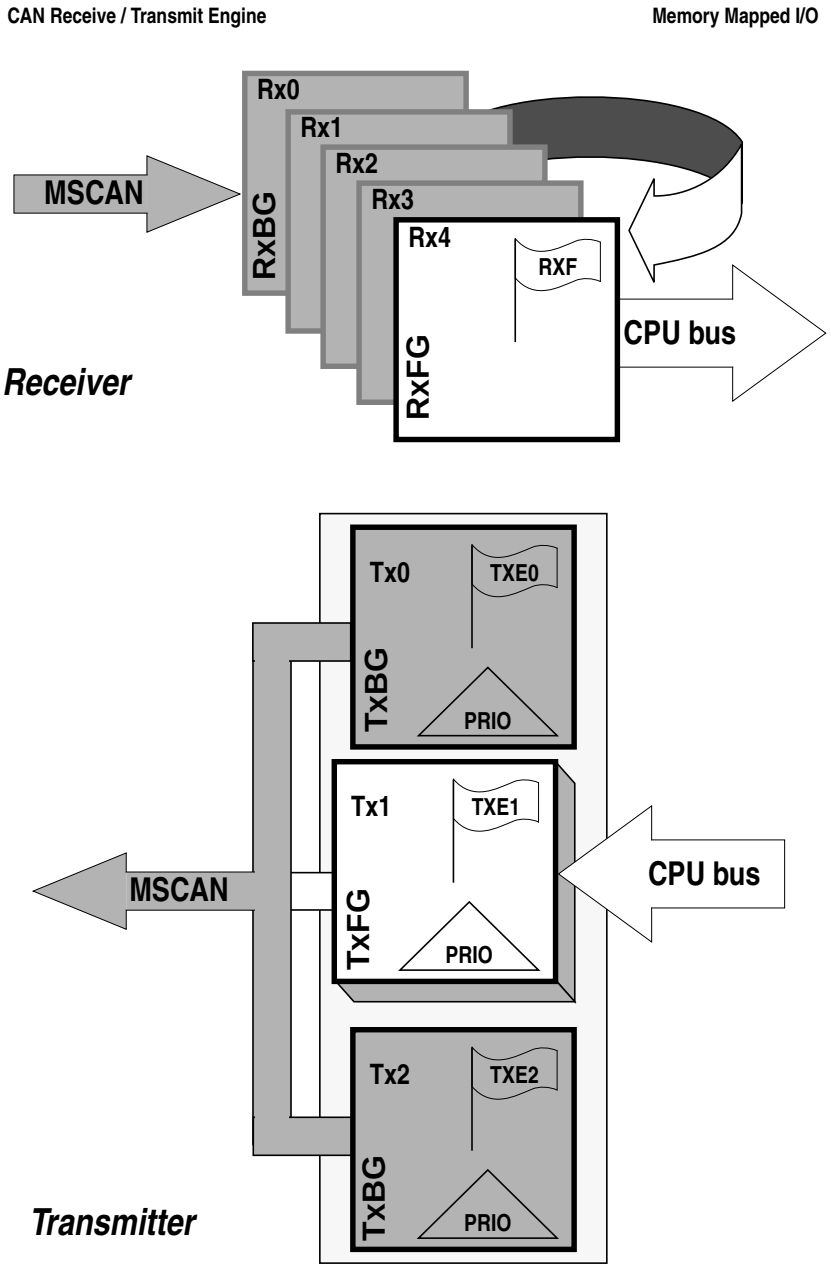


Figure 11-39. User Model for Message Buffer Organization

12.4 Functional Description

Figure 12-19 shows a detailed block diagram of the PIT module. The main parts of the PIT are status, control and data registers, two 8-bit down-counters, four 16-bit down-counters and an interrupt/trigger interface.

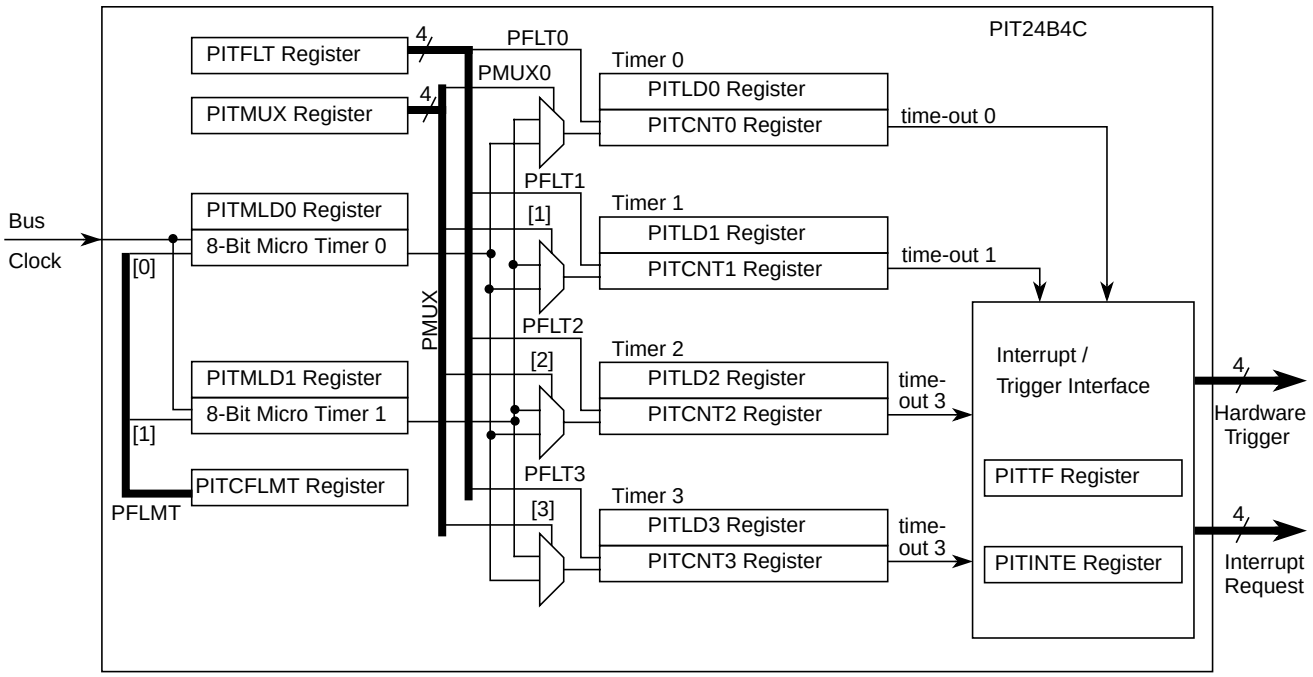


Figure 12-19. PIT24B4C Detailed Block Diagram

12.4.1 Timer

As shown in Figure 12-1 and Figure 12-19, the 24-bit timers are built in a two-stage architecture with four 16-bit modulus down-counters and two 8-bit modulus down-counters. The 16-bit timers are clocked with two selectable micro time bases which are generated with 8-bit modulus down-counters. Each 16-bit timer is connected to micro time base 0 or 1 via the PMUX[3:0] bit setting in the PIT Multiplex (PITMUX) register.

A timer channel is enabled if the module enable bit PITE in the PIT control and force load micro timer (PITCFLMT) register is set and if the corresponding PCE bit in the PIT channel enable (PITCE) register is set. Two 8-bit modulus down-counters are used to generate two micro time bases. As soon as a micro time base is selected for an enabled timer channel, the corresponding micro timer modulus down-counter will load its start value as specified in the PITMTLD0 or PITMTLD1 register and will start down-counting. Whenever the micro timer down-counter has counted to zero the PITMTLD register is reloaded and the connected 16-bit modulus down-counters count one cycle.

15.3.2.3 SPI Baud Rate Register (SPIBR)

Module Base +0x0002

	7	6	5	4	3	2	1	0
R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 15-5. SPI Baud Rate Register (SPIBR)

Read: Anytime

Write: Anytime; writes to the reserved bits have no effect

Table 15-6. SPIBR Field Descriptions

Field	Description
6–4 SPPR[2:0]	**SPI Baud Rate Preselection Bits** — These bits specify the SPI baud rates as shown in Table 15-7. In master mode, a change of these bits will abort a transmission in progress and force the SPI system into idle state.
2–0 SPR[2:0]	**SPI Baud Rate Selection Bits** — These bits specify the SPI baud rates as shown in Table 15-7. In master mode, a change of these bits will abort a transmission in progress and force the SPI system into idle state.

The baud rate divisor equation is as follows:

$$\text{BaudRateDivisor} = (\text{SPPR} + 1) \bullet 2^{(\text{SPR} + 1)} \qquad \textit{Eqn. 15-1}$$

The baud rate can be calculated with the following equation:

$$\text{Baud Rate} = \text{BusClock} / \text{BaudRateDivisor} \qquad \textit{Eqn. 15-2}$$

NOTE

For maximum allowed baud rates, please refer to the SPI Electrical Specification in the Electricals chapter of this data sheet.

Table 15-7. Example SPI Baud Rate Selection (25 MHz Bus Clock) (Sheet 1 of 3)

SPPR2	SPPR1	SPPR0	SPR2	SPR1	SPR0	Baud Rate Divisor	Baud Rate
0	0	0	0	0	0	2	12.5 Mbit/s
0	0	0	0	0	1	4	6.25 Mbit/s
0	0	0	0	1	0	8	3.125 Mbit/s
0	0	0	0	1	1	16	1.5625 Mbit/s
0	0	0	1	0	0	32	781.25 kbit/s
0	0	0	1	0	1	64	390.63 kbit/s
0	0	0	1	1	0	128	195.31 kbit/s
0	0	0	1	1	1	256	97.66 kbit/s
0	0	1	0	0	0	4	6.25 Mbit/s
0	0	1	0	0	1	8	3.125 Mbit/s
0	0	1	0	1	0	16	1.5625 Mbit/s
0	0	1	0	1	1	32	781.25 kbit/s

Table 18-23. D-Flash Protection Address Range

DPS[4:0]	Global Address Range	Protected Size
1_0110	0x10_0000 – 0x10_16FF	5888 bytes
1_0111	0x10_0000 – 0x10_17FF	6144 bytes
1_1000	0x10_0000 – 0x10_18FF	6400 bytes
1_1001	0x10_0000 – 0x10_19FF	6656 bytes
1_1010	0x10_0000 – 0x10_1AFF	6912 bytes
1_1011	0x10_0000 – 0x10_1BFF	7168 bytes
1_1100	0x10_0000 – 0x10_1CFF	7424 bytes
1_1101	0x10_0000 – 0x10_1DFF	7680 bytes
1_1110	0x10_0000 – 0x10_1EFF	7936 bytes
1_1111	0x10_0000 – 0x10_1FFF	8192 bytes

18.3.2.11 Flash Common Command Object Register (FCCOB)

The FCCOB is an array of six words addressed via the CCOBIX index found in the FCCOBIX register. Byte wide reads and writes are allowed to the FCCOB register.

Offset Module Base + 0x000A

	7	6	5	4	3	2	1	0
R W	CCOB[15:8]							
Reset	0	0	0	0	0	0	0	0

Figure 18-16. Flash Common Command Object High Register (FCCOBHI)

Offset Module Base + 0x000B

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

Figure 18-17. Flash Common Command Object Low Register (FCCOBLO)

18.3.2.11.1 FCCOB - NVM Command Mode

NVM command mode uses the indexed FCCOB register to provide a command code and its relevant parameters to the Memory Controller. The user first sets up all required FCCOB fields and then initiates the command's execution by writing a 1 to the CCIF bit in the FSTAT register (a 1 written by the user clears the CCIF command completion flag to 0). When the user clears the CCIF bit in the FSTAT register all FCCOB parameter fields are locked and cannot be changed by the user until the command completes

- Automated program and erase algorithm with verify and generation of ECC parity bits
- Fast sector erase and phrase program operation
- Flexible protection scheme to prevent accidental program or erase of P-Flash memory

19.1.2.2 D-Flash Features

- 8 Kbytes of D-Flash memory composed of one 8 Kbyte Flash block divided into 32 sectors of 256 bytes
- Single bit fault correction and double bit fault detection within a word during read operations
- Automated program and erase algorithm with verify and generation of ECC parity bits
- Fast sector erase and word program operation
- Protection scheme to prevent accidental program or erase of D-Flash memory
- Ability to program up to four words in a burst sequence

19.1.2.3 Other Flash Module Features

- No external high-voltage power supply required for Flash memory program and erase operations
- Interrupt generation on Flash command completion and Flash error detection
- Security mechanism to prevent unauthorized access to the Flash memory

19.1.3 Block Diagram

The block diagram of the Flash module is shown in Figure 19-1.

Table 19-4. Program IFR Fields

Global Address (PGMIFRON)	Size (Bytes)	Field Description
0x40_0000 – 0x40_0007	8	Device ID
0x40_0008 – 0x40_00E7	224	Reserved
0x40_00E8 – 0x40_00E9	2	Version ID
0x40_00EA – 0x40_00FF	22	Reserved
0x40_0100 – 0x40_013F	64	Program Once Field Refer to Section 19.4.2.6, “Program Once Command”
0x40_0140 – 0x40_01FF	192	Reserved

Table 19-5. D-Flash and Memory Controller Resource Fields

Global Address	Size (Bytes)	Description
0x10_0000 – 0x10_1FFF	8,192	D-Flash Memory
0x10_2000 – 0x11_FFFF	122,880	Reserved
0x12_0000 – 0x12_007F	128	D-Flash Nonvolatile Information Register (DFIFRON[1] = 1)
0x12_0080 – 0x12_0FFF	3,968	Reserved
0x12_1000 – 0x12_1FFF	4,096	Reserved
0x12_2000 – 0x12_3CFF	7,242	Reserved
0x12_3D00 – 0x12_3FFF	768	Memory Controller Scratch RAM (MGRAMON[1] = 1)
0x12_4000 – 0x12_E7FF	43,008	Reserved
0x12_E800 – 0x12_FFFF	6,144	Reserved
0x13_0000 – 0x13_FFFF	65,536	Reserved

1 MMCCTL1 register bit

Address & Name		7	6	5	4	3	2	1	0
0x0010 FOPT	R	NV7	NV6	NV5	NV4	NV3	NV2	NV1	NV0
	W								
0x0011 FRSV2	R	0	0	0	0	0	0	0	0
	W								
0x0012 FRSV3	R	0	0	0	0	0	0	0	0
	W								
0x0013 FRSV4	R	0	0	0	0	0	0	0	0
	W								

= Unimplemented or Reserved

Figure 20-4. FTMR64K1 Register Summary (continued)

20.3.2.1 Flash Clock Divider Register (FCLKDIV)

The FCLKDIV register is used to control timed events in program and erase algorithms.

Offset Module Base + 0x0000

	7	6	5	4	3	2	1	0
R	FDIVLD	FDIV[6:0]						
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 20-5. Flash Clock Divider Register (FCLKDIV)

All bits in the FCLKDIV register are readable, bits 6–0 are write once and bit 7 is not writable.

Table 20-6. FCLKDIV Field Descriptions

Field	Description
7 FDIVLD	**Clock Divider Loaded** 0 FCLKDIV register has not been written 1 FCLKDIV register has been written since the last reset
6–0 FDIV[6:0]	**Clock Divider Bits** — FDIV[6:0] must be set to effectively divide OSCCLK down to generate an internal Flash clock, FCLK, with a target frequency of 1 MHz for use by the Flash module to control timed events during program and erase algorithms. Table 20-7 shows recommended values for FDIV[6:0] based on OSCCLK frequency. Please refer to Section 20.4.1, “Flash Command Operations,” for more information.

Offset Module Base + 0x000A

	7	6	5	4	3	2	1	0
R W	CCOB[15:8]							
Reset	0	0	0	0	0	0	0	0

Figure 20-16. Flash Common Command Object High Register (FCCOBHI)

Offset Module Base + 0x000B

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

Figure 20-17. Flash Common Command Object Low Register (FCCOBLO)

20.3.2.11.1 FCCOB - NVM Command Mode

NVM command mode uses the indexed FCCOB register to provide a command code and its relevant parameters to the Memory Controller. The user first sets up all required FCCOB fields and then initiates the command's execution by writing a 1 to the CCIF bit in the FSTAT register (a 1 written by the user clears the CCIF command completion flag to 0). When the user clears the CCIF bit in the FSTAT register all FCCOB parameter fields are locked and cannot be changed by the user until the command completes (as evidenced by the Memory Controller returning CCIF to 1). Some commands return information to the FCCOB register array.

The generic format for the FCCOB parameter fields in NVM command mode is shown in Table 20-24. The return values are available for reading after the CCIF flag in the FSTAT register has been returned to 1 by the Memory Controller. Writes to the unimplemented parameter fields (CCOBIX = 110 and CCOBIX = 111) are ignored with reads from these fields returning 0x0000.

Table 20-24 shows the generic Flash command format. The high byte of the first word in the CCOB array contains the command code, followed by the parameters for this specific Flash command. For details on the FCCOB settings required by each command, see the Flash command descriptions in Section 20.4.2.

Table 20-24. FCCOB - NVM Command Mode (Typical Usage)

CCOBIX[2:0]	Byte	FCCOB Parameter Fields (NVM Command Mode)
000	HI	FCMD[7:0] defining Flash command
	LO	0, Global address [22:16]
001	HI	Global address [15:8]
	LO	Global address [7:0]
010	HI	Data 0 [15:8]
	LO	Data 0 [7:0]

Table 20-35. Erase Verify P-Flash Section Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x03	Global address [22:16] of a P-Flash block
001	Global address [15:0] of the first phrase to be verified	
010	Number of phrases to be verified	

Upon clearing CCIF to launch the Erase Verify P-Flash Section command, the Memory Controller will verify the selected section of Flash memory is erased. The CCIF flag will set after the Erase Verify P-Flash Section operation has completed.

Table 20-36. Erase Verify P-Flash Section Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 010 at command launch
		Set if command not available in current mode (see Table 20-28)
		Set if an invalid global address [22:0] is supplied[1]
		Set if a misaligned phrase address is supplied (global address [2:0] != 000)
		Set if the requested section crosses a 128 Kbyte boundary
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read[2]
	MGSTAT0	Set if any non-correctable errors have been encountered during the read[2]

1 As defined by the memory map for FTMR128K1.

2 As found in the memory map for FTMR128K1.

20.4.2.4 Read Once Command

The Read Once command provides read access to a reserved 64 byte field (8 phrases) located in the nonvolatile information register of P-Flash block 0. The Read Once field is programmed using the Program Once command described in Section 20.4.2.6. The Read Once command must not be executed from the Flash block containing the Program Once reserved field to avoid code runaway.

Table 20-37. Read Once Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x04	Not Required
001	Read Once phrase index (0x0000 - 0x0007)	
010	Read Once word 0 value	
011	Read Once word 1 value	
100	Read Once word 2 value	
101	Read Once word 3 value	

Table A-7. 3.3-V I/O Characteristics

Conditions are 3.13 V < V_{DD35} < 3.6 V junction temperature from –40°C to +150°C, unless otherwise noted I/O Characteristics for all I/O pins except EXTAL, XTAL,TEST and supply pins.							
13	P	Port H, J, P interrupt input pulse filtered (STOP)[3]	t_{PULSE}	—	—	3	µs
14	P	Port H, J, P interrupt input pulse passed (STOP)[3]	t_{PULSE}	10	—	—	µs
15	D	Port H, J, P interrupt input pulse filtered ($\overline{STOP}$)	t_{PULSE}	—	—	3	tcyc
16	D	Port H, J, P interrupt input pulse passed ($\overline{STOP}$)	t_{PULSE}	4	—	—	tcyc
17	D	$\overline{IRQ}$ pulse width, edge-sensitive mode ($\overline{STOP}$)	PW_{IRQ}	1	—	—	tcyc
18	D	$\overline{XIRQ}$ pulse width with X-bit set (STOP)	PW_{XIRQ}	4	—	—	tosc

1 Maximum leakage current occurs at maximum operating temperature.
2 Refer to Section A.1.4, “Current Injection” for more details
3 Parameter only applies in stop or pseudo stop mode.

$$t = 350 \cdot \frac{1}{f_{NVMBUS}}$$

A.3.1.14 Erase Verify D-Flash Section (FCMD=0x10)

Erase Verify D-Flash for a given number of words N_W is given by .

$$t_{check} \approx (840 + N_W) \cdot \frac{1}{f_{NVMBUS}}$$

A.3.1.15 D-Flash Programming (FCMD=0x11)

D-Flash programming time is dependent on the number of words being programmed and their location with respect to a row boundary, because programming across a row boundary requires extra steps. The D-Flash programming time is specified for different cases (1,2,3,4 words and 4 words across a row boundary) at a 40MHz bus frequency. The typical programming time can be calculated using the following equation, whereby N_w denotes the number of words; BC=0 if no boundary is crossed and BC=1 if a boundary is crossed.

$$t_{dpgm} = \left((15 + (54 \cdot N_w) + (16 \cdot BC)) \cdot \frac{1}{f_{NVMOP}}\right) + \left((460 + (640 \cdot N_W) + (500 \cdot BC)) \cdot \frac{1}{f_{NVMBUS}}\right)$$

The maximum programming time can be calculated using the following equation

$$t_{dpgm} = \left((15 + (56 \cdot N_w) + (16 \cdot BC)) \cdot \frac{1}{f_{NVMOP}}\right) + \left((460 + (840 \cdot N_W) + (500 \cdot BC)) \cdot \frac{1}{f_{NVMBUS}}\right)$$

A.3.1.16 Erase D-Flash Sector (FCMD=0x12)

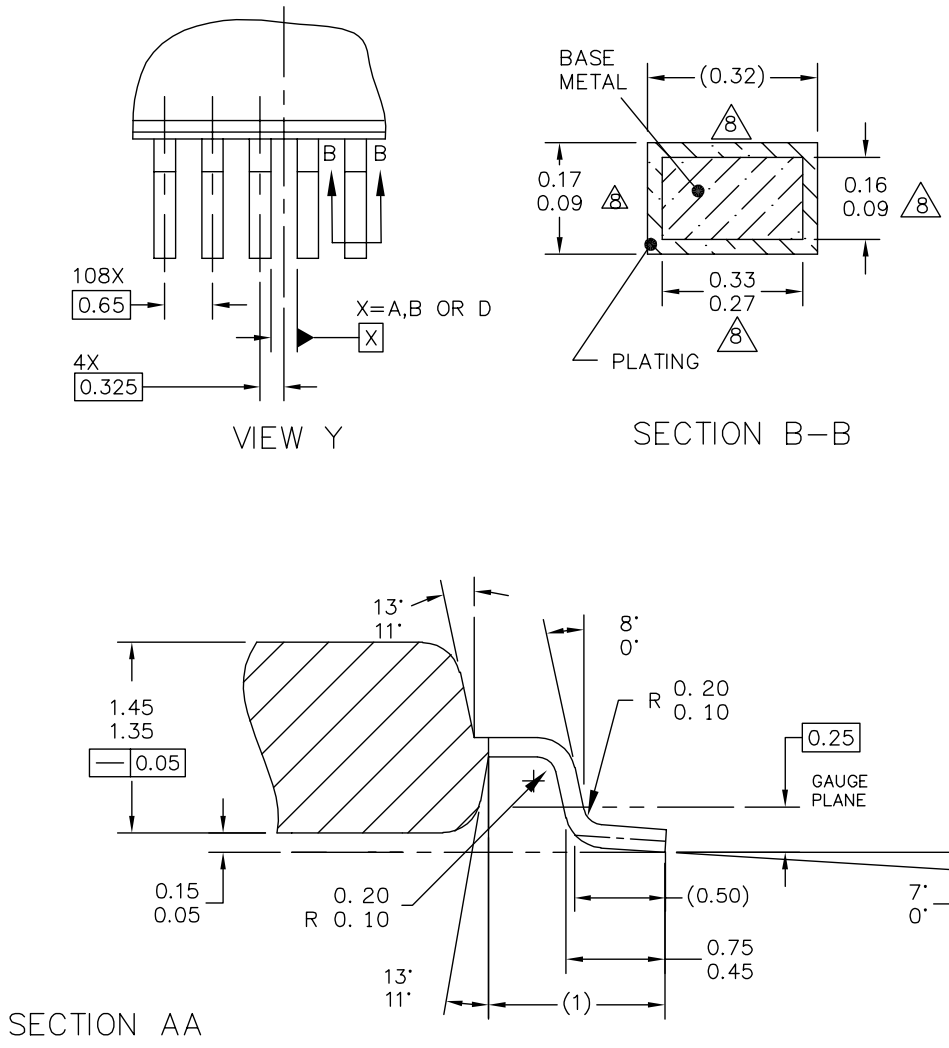
Typical D-Flash sector erase times are those expected on a new device, where no margin verify fails occur. They can be calculated using the following equation.

$$t_{eradf} \approx 5025 \cdot \frac{1}{f_{NVMOP}} + 700 \cdot \frac{1}{f_{NVMBUS}}$$

Maximum D-Fash sector erase times can be calculated using the following equation.

$$t_{eradf} \approx 20100 \cdot \frac{1}{f_{NVMOP}} + 3300 \cdot \frac{1}{f_{NVMBUS}}$$

The D-Flash sector erase time on a new device is ~5ms and can extend to 20ms as the flash is cycled.



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	**MECHANICAL OUTLINE**	PRINT VERSION NOT TO SCALE
TITLE: 112LD LQFP 20 X 20 X 1.4 0.65 PITCH	DOCUMENT NO: 98ASS23330W	REV: F
	CASE NUMBER: 987-03	15 DEC 2006
	STANDARD: JEDEC MS-026 BFA	

Figure B-2. 112-pin LQFP (case no. 987) - page 2

0x0240–0x027F Port Integration Module (PIM) Map 5 of 5

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0260	PTH	R	PTH7	PTH6	PTH5	PTH4	PTH3	PTH2	PTH1	PTH0
		W								
0x0261	PTIH	R	PTIH7	PTIH6	PTIH5	PTIH4	PTIH3	PTIH2	PTIH1	PTIH0
		W								
0x0262	DDRH	R	DDRH7	DDRH6	DDRH5	DDRH4	DDRH3	DDRH2	DDRH1	DDRH0
		W								
0x0263	RDRH	R	RDRH7	RDRH6	RDRH5	RDRH4	RDRH3	RDRH2	RDRH1	RDRH0
		W								
0x0264	PERH	R	PERH7	PERH6	PERH5	PERH4	PERH3	PERH2	PERH1	PERH0
		W								
0x0265	PPSH	R	PPSH7	PPSH6	PPSH5	PPSH4	PPSH3	PPSH2	PPSH1	PPSH0
		W								
0x0266	PIEH	R	PIEH7	PIEH6	PIEH5	PIEH4	PIEH3	PIEH2	PIEH1	PIEH0
		W								
0x0267	PIFH	R	PIFH7	PIFH6	PIFH5	PIFH4	PIFH3	PIFH2	PIFH1	PIFH0
		W								
0x0268	PTJ	R	PTJ7	PTJ6	0	0	0	0	PTJ1	PTJ0
		W								
0x0269	PTIJ	R	PTIJ7	PTIJ6	0	0	0	0	PTIJ1	PTIJ0
		W								
0x026A	DDRJ	R	DDRJ7	DDRJ6	0	0	0	0	DDRJ1	DDRJ0
		W								
0x026B	RDRJ	R	RDRJ7	RDRJ6	0	0	0	0	RDRJ1	RDRJ0
		W								
0x026C	PERJ	R	PERJ7	PERJ6	0	0	0	0	PERJ1	PERJ0
		W								
0x026D	PPSJ	R	PPSJ7	PPSJ6	0	0	0	0	PPSJ1	PPSJ0
		W								
0x026E	PIEJ	R	PIEJ7	PIEJ6	0	0	0	0	PIEJ1	PIEJ0
		W								
0x026f	PIFJ	R	PIFJ7	PIFJ6	0	0	0	0	PIFJ1	PIFJ0
		W								
0x0270	PT0AD0	R	PT0AD07	PT0AD06	PT0AD05	PT0AD04	PT0AD03	PT0AD02	PT0AD01	PT0AD00
		W								
0x0271	PT1AD0	R	PT1AD07	PT1AD06	PT1AD05	PT1AD04	PT1AD03	PT1AD02	PT1AD01	PT1AD00
		W								
0x0272	DDR0AD0	R	DDR0AD07	DDR0AD06	DDR0AD05	DDR0AD04	DDR0AD03	DDR0AD02	DDR0AD01	DDR0AD00
		W								
0x0273	DDR1AD0	R	DDR1AD07	DDR1AD06	DDR1AD05	DDR1AD04	DDR1AD03	DDR1AD02	DDR1AD01	DDR1AD00
		W								
0x0274	RDR0AD0	R	RDR0AD07	RDR0AD06	RDR0AD05	RDR0AD04	RDR0AD03	RDR0AD02	RDR0AD01	RDR0AD00
		W								
0x0275	RDR1AD0	R	RDR1AD07	RDR1AD06	RDR1AD05	RDR1AD04	RDR1AD03	RDR1AD02	RDR1AD01	RDR1AD00
		W								