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	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	CANbus, EBI/EMI, I²C, SCI, SPI
Peripherals	LCD, Motor control PWM, POR, PWM, WDT
Number of I/O	85
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	112-LQFP
Supplier Device Package	112-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s12hz256val

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
RESERVED2	R	0	0	0	0	0	0	0	0
	W								
RESERVED3	R	0	0	0	0	0	0	0	0
	W								
RESERVED4	R	0	0	0	0	0	0	0	0
	W								

= Unimplemented or Reserved

Figure 2-3. FTS256K2 Register Summary (continued)

2.3.2.1 Flash Clock Divider Register (FCLKDIV)

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

	7	6	5	4	3	2	1	0
R	FDIVLD	PRDIV8	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 2-4. 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 2-4. FCLKDIV Field Descriptions

Field	Description
7 FDIVLD	**Clock Divider Loaded.** 0 Register has not been written. 1 Register has been written to since the last reset.
6 PRDIV8	**Enable Prescalar by 8.** 0 The oscillator clock is directly fed into the clock divider. 1 The oscillator clock is divided by 8 before feeding into the clock divider.
5-0 FDIV[5:0]	**Clock Divider Bits** — The combination of PRDIV8 and FDIV[5:0] must divide the oscillator clock down to a frequency of 150 kHz–200 kHz. The maximum divide ratio is 512. Please refer to Section 2.4.1.1, “Writing the FCLKDIV Register” for more information.

2.3.2.2 Flash Security Register (FSEC)

The unbanked FSEC register holds all bits associated with the security of the MCU and Flash module.

Table 3-9. Valid EEPROM Command List

Command	Meaning
0x05	Erase verify
0x20	Word program
0x40	Sector erase
0x41	Mass erase
0x60	Sector modify

3.3.2.8 RESERVED3

This register is reserved for factory testing and is not accessible to the user.

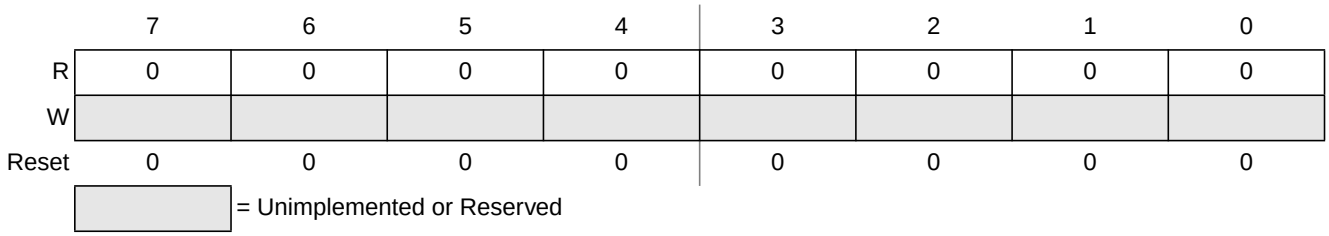

Figure 3-12. RESERVED3

All bits read 0 and are not writable.

3.3.2.9 EEPROM Address Register (EADDR)

EADDRHI and EADDRLO are the EEPROM address registers.

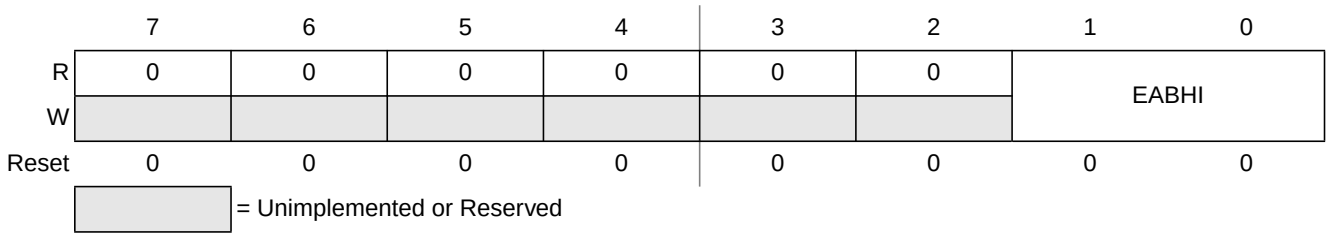

Figure 3-13. EEPROM Address High Register (EADDRHI)

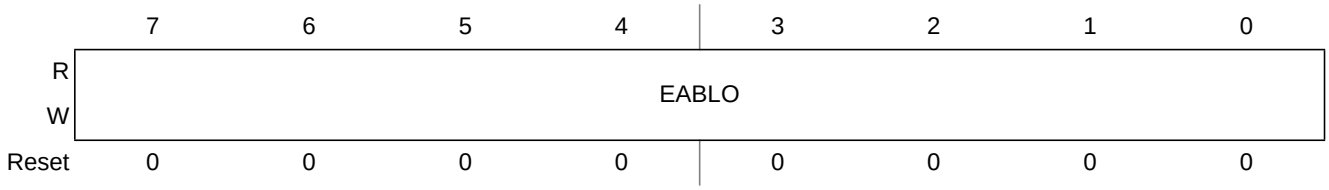

Figure 3-14. EEPROM Address Low Register (EADDRLO)

In normal modes, all EADDRHI and EADDRLO bits read 0 and are not writable.

In special modes, all EADDRHI and EADDRLO bits are readable and writable except EADDRHI[7:2] which are not writable and always read 0.

4.3 Memory Map and Register Definition

This section provides a detailed description of all registers. Table 4-2 is a standard memory map of port integration module.

Table 4-2. PIM9HZ256 Memory Map

Address Offset	Use	Access
0x0000	Port T I/O Register (PTT)	R/W
0x0001	Port T Input Register (PTIT)	R
0x0002	Port T Data Direction Register (DDRT)	R/W
0x0003	Port T Reduced Drive Register (RDRT)	R/W
0x0004	Port T Pull Device Enable Register (PERT)	R/W
0x0005	Port T Polarity Select Register (PPST)	R/W
0x0006 - 0x0007	Reserved	—
0x0008	Port S I/O Register (PTS)	R/W
0x0009	Port S Input Register (PTIS)	R
0x000A	Port S Data Direction Register (DDRS)	R/W
0x000B	Port S Reduced Drive Register (RDRS)	R/W
0x000C	Port S Pull Device Enable Register (PERS)	R/W
0x000D	Port S Polarity Select Register (PPSS)	R/W
0x000E	Port S Wired-OR Mode Register (WOMS)	R/W
0x000F	Reserved	—
0x0010	Port M I/O Register (PTM)	R/W
0x0011	Port M Input Register (PTIM)	R
0x0012	Port M Data Direction Register (DDRM)	R/W
0x0013	Port M Reduced Drive Register (RDRM)	R/W
0x0014	Port M Pull Device Enable Register (PERM)	R/W
0x0015	Port M Polarity Select Register (PPSM)	R/W
0x0016	Port M Wired-OR Mode Register (WOMM)	R/W
0x0017	Reserved	—
0x0018	Port P I/O Register (PTP)	R/W
0x0019	Port P Input Register (PTIP)	R
0x001A	Port P Data Direction Register (DDRP)	R/W
0x001B	Port P Reduced Drive Register (RDRP)	R/W
0x001C	Port P Pull Device Enable Register (PERP)	R/W
0x001D	Port P Polarity Select Register (PPSP)	R/W
0x001E	Port P Wired-OR Mode Register (WOMP)	R/W
0x001F - 0x002F	Reserved	—

5.3.2.12 CRG COP Timer Arm/Reset Register (ARMCOP)

This register is used to restart the COP time-out period.

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset	0	0	0	0	0	0	0	0

Figure 5-15. ARMCOP Register Diagram

Read: always reads 0x0000

Write: anytime

When the COP is disabled (CR[2:0] = “000”) writing to this register has no effect.

When the COP is enabled by setting CR[2:0] nonzero, the following applies:

Writing any value other than 0x0055 or 0x00AA causes a COP reset. To restart the COP time-out period you must write 0x0055 followed by a write of 0x00AA. Other instructions may be executed between these writes but the sequence (0x0055, 0x00AA) must be completed prior to COP end of time-out period to avoid a COP reset. Sequences of 0x0055 writes or sequences of 0x00AA writes are allowed. When the WCOP bit is set, 0x0055 and 0x00AA writes must be done in the last 25% of the selected time-out period; writing any value in the first 75% of the selected period will cause a COP reset.

5.4 Functional Description

This section gives detailed informations on the internal operation of the design.

5.4.1 Phase Locked Loop (PLL)

The PLL is used to run the MCU from a different time base than the incoming OSCCLK. For increased flexibility, OSCCLK can be divided in a range of 1 to 16 to generate the reference frequency. This offers a finer multiplication granularity. The PLL can multiply this reference clock by a multiple of 2, 4, 6,... 126,128 based on the SYNR register.

$$PLLCLK = 2 \times OSCCLK \times \frac{[SYNR + 1]}{[REFDV + 1]}$$

Although it is possible to set the two dividers to command a very high clock frequency, do not exceed the specified bus frequency limit for the MCU.
If (PLLSEL = 1), Bus Clock = PLLCLK / 2

The PLL is a frequency generator that operates in either acquisition mode or tracking mode, depending on the difference between the output frequency and the target frequency. The PLL can change between acquisition and tracking modes either automatically or manually.

There are five different scenarios for the CRG to restart the MCU from wait mode:

- External reset
- Clock monitor reset
- COP reset
- Self-clock mode interrupt
- Real-time interrupt (RTI)

If the MCU gets an external reset during wait mode active, the CRG asynchronously restores all configuration bits in the register space to its default settings and starts the reset generator. After completing the reset sequence processing begins by fetching the normal reset vector. Wait mode is exited and the MCU is in run mode again.

If the clock monitor is enabled (CME=1) the MCU is able to leave wait mode when loss of oscillator/external clock is detected by a clock monitor fail. If the SCME bit is not asserted the CRG generates a clock monitor fail reset (CMRESET). The CRG's behavior for CMRESET is the same compared to external reset, but another reset vector is fetched after completion of the reset sequence. If the SCME bit is asserted the CRG generates a SCM interrupt if enabled (SCMIE=1). After generating the interrupt the CRG enters self-clock mode and starts the clock quality checker (see Section 5.4.4, "Clock Quality Checker"). Then the MCU continues with normal operation.If the SCM interrupt is blocked by SCMIE = 0, the SCMIF flag will be asserted and clock quality checks will be performed but the MCU will not wake-up from wait mode.

If any other interrupt source (e.g. RTI) triggers exit from wait mode the MCU immediately continues with normal operation. If the PLL has been powered-down during wait mode the PLLSEL bit is cleared and the MCU runs on OSCCLK after leaving wait mode. The software must manually set the PLLSEL bit again, in order to switch system and core clocks to the PLLCLK.

If wait mode is entered from self-clock mode, the CRG will continue to check the clock quality until clock check is successful. The PLL and voltage regulator (VREG) will remain enabled.

Table 5-11 summarizes the outcome of a clock loss while in wait mode.

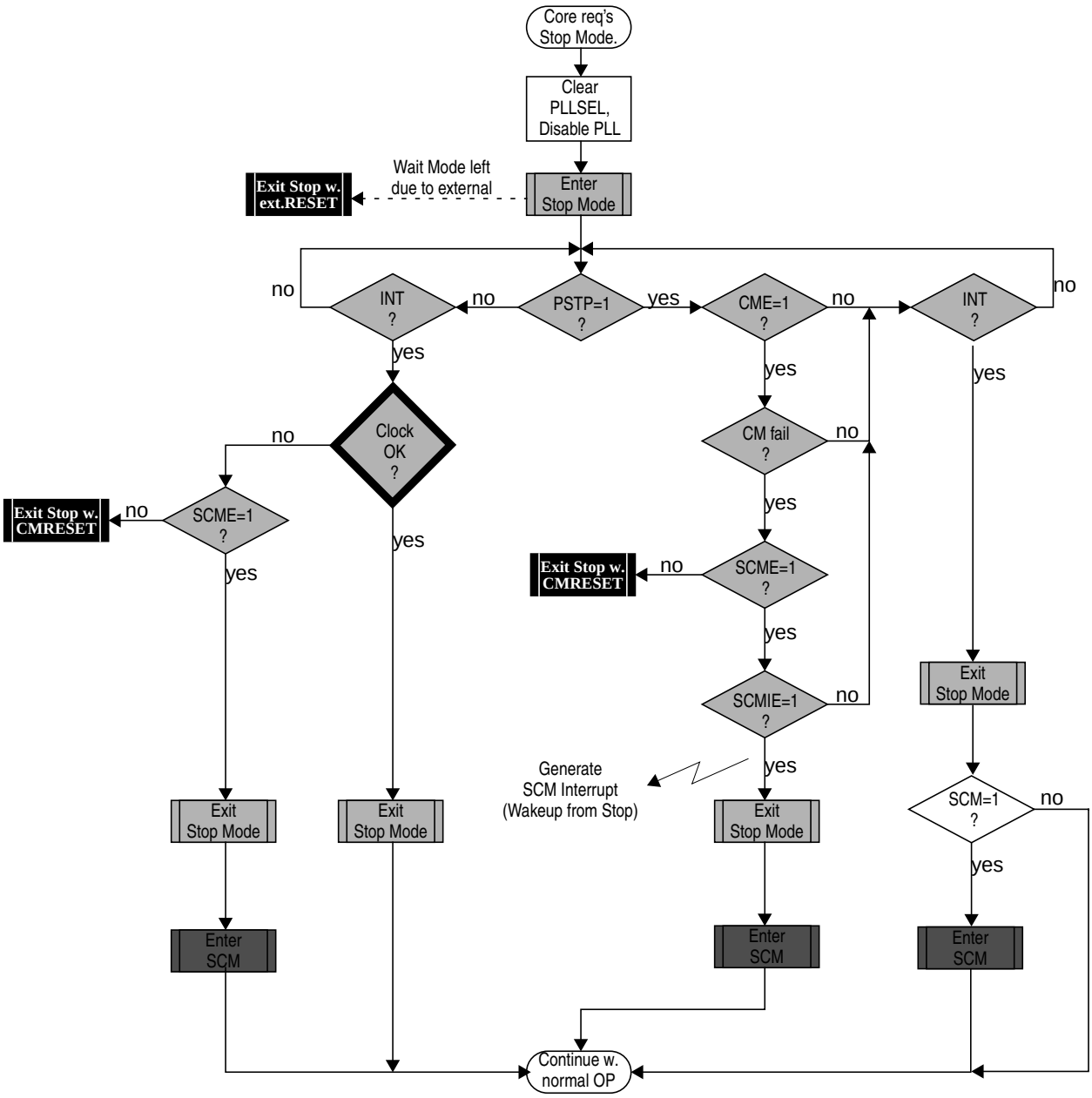

Figure 5-24. Stop Mode Entry/Exit Sequence

5.4.10.1 Wake-Up from Pseudo-Stop (PSTP=1)

Wake-up from pseudo-stop is the same as wake-up from wait mode. There are also three different scenarios for the CRG to restart the MCU from pseudo-stop mode:

- External reset
- Clock monitor fail
- Wake-up interrupt

8.3.2 Register Descriptions

This section consists of register descriptions. Each description includes a standard register diagram. Details of register bit and field function follow the register diagrams, in bit order.

8.3.2.1 LCD Control Register 0 (LCDCR0)

	7	6	5	4	3	2	1	0
R	LCDEN	0	LCLK2	LCLK1	LCLK0	BIAS	DUTY1	DUTY0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 8-2. LCD Control Register 0 (LCDCR0)

Read: anytime

Write: LCDEN anytime. To avoid segment flicker the clock prescaler bits, the bias select bit and the duty select bits must not be changed when the LCD is enabled.

Table 8-3. LCDCR0 Field Descriptions

Field	Description
7 LCDEN	**LCD32F4B Driver System Enable** — The LCDEN bit starts the LCD waveform generator. 0 All frontplane and backplane pins are disabled. In addition, the LCD32F4B system is disabled and all LCD waveform generation clocks are stopped. 1 LCD driver system is enabled. All FP[31:0] pins with FP[31:0]EN set, will output an LCD driver waveform The BP[3:0] pins will output an LCD32F4B driver waveform based on the settings of DUTY0 and DUTY1.
5:3 LCLK[2:0]	**LCD Clock Prescaler** — The LCD clock prescaler bits determine the OSCCLK divider value to produce the LCD clock frequency. For detailed description of the correlation between LCD clock prescaler bits and the divider value please refer to Table 8-7.
2 BIAS	**BIAS Voltage Level Select** — This bit selects the bias voltage levels during various LCD operating modes, as shown in Table 8-8.
1:0 DUTY[1:0]	**LCD Duty Select** — The DUTY1 and DUTY0 bits select the duty (multiplex mode) of the LCD32F4B driver system, as shown in Table 8-8.

11.5 Resets

The reset state of each individual bit is listed in Section 11.3, “Memory Map and Register Definition,” which details the registers and their bit-fields.

11.6 Interrupts

IIC uses only one interrupt vector.

Table 11-9. Interrupt Summary

Interrupt	Offset	Vector	Priority	Source	Description
IIC Interrupt	—	—	—	IBAL, TCF, IAAS bits in IBSR register	When either of IBAL, TCF or IAAS bits is set may cause an interrupt based on arbitration lost, transfer complete or address detect conditions

Internally there are three types of interrupts in IIC. The interrupt service routine can determine the interrupt type by reading the status register.

IIC Interrupt can be generated on

1. Arbitration lost condition (IBAL bit set)
2. Byte transfer condition (TCF bit set)
3. Address detect condition (IAAS bit set)

The IIC interrupt is enabled by the IBIE bit in the IIC control register. It must be cleared by writing 0 to the IBF bit in the interrupt service routine.

11.7 Initialization/Application Information

11.7.1 IIC Programming Examples

11.7.1.1 Initialization Sequence

Reset will put the IIC bus control register to its default status. Before the interface can be used to transfer serial data, an initialization procedure must be carried out, as follows:

1. Update the frequency divider register (IBFD) and select the required division ratio to obtain SCL frequency from system clock.
2. Update the IIC bus address register (IBAD) to define its slave address.
3. Set the IBEN bit of the IIC bus control register (IBCR) to enable the IIC interface system.
4. Modify the bits of the IIC bus control register (IBCR) to select master/slave mode, transmit/receive mode and interrupt enable or not.

12.1.2 Features

The basic features of the MSCAN are as follows:

- Implementation of the CAN protocol — Version 2.0A/B
  - — Standard and extended data frames
  - — Zero to eight bytes data length
  - — Programmable bit rate up to 1 Mbps[1]
  - — Support for remote frames
- Five receive buffers with FIFO storage scheme
- Three transmit buffers with internal prioritization using a "local priority" concept
- Flexible maskable identifier filter supports two full-size (32-bit) extended identifier filters, or four 16-bit filters, or eight 8-bit filters
- Programmable wakeup functionality with integrated low-pass filter
- Programmable loopback mode supports self-test operation
- Programmable listen-only mode for monitoring of CAN bus
- Separate signalling and interrupt capabilities for all CAN receiver and transmitter error states (warning, error passive, bus-off)
- Programmable MSCAN clock source either bus clock or oscillator clock
- Internal timer for time-stamping of received and transmitted messages
- Three low-power modes: sleep, power down, and MSCAN enable
- Global initialization of configuration registers

12.1.3 Modes of Operation

The following modes of operation are specific to the MSCAN. See Section 12.4, "Functional Description," for details.

- Listen-Only Mode
- MSCAN Sleep Mode
- MSCAN Initialization Mode
- MSCAN Power Down Mode

12.2 External Signal Description

The MSCAN uses two external pins:

12.2.1 RXCAN — CAN Receiver Input Pin

RXCAN is the MSCAN receiver input pin.

1. Depending on the actual bit timing and the clock jitter of the PLL.

Table 14-2. SPICR1 Field Descriptions

Field	Description
7 SPIE	**SPI Interrupt Enable Bit** — This bit enables SPI interrupt requests, if SPIF or MODF status flag is set. 0 SPI interrupts disabled. 1 SPI interrupts enabled.
6 SPE	**SPI System Enable Bit** — This bit enables the SPI system and dedicates the SPI port pins to SPI system functions. If SPE is cleared, SPI is disabled and forced into idle state, status bits in SPISR register are reset. 0 SPI disabled (lower power consumption). 1 SPI enabled, port pins are dedicated to SPI functions.
5 SPTIE	**SPI Transmit Interrupt Enable** — This bit enables SPI interrupt requests, if SPTEF flag is set. 0 SPTEF interrupt disabled. 1 SPTEF interrupt enabled.
4 MSTR	**SPI Master/Slave Mode Select Bit** — This bit selects, if the SPI operates in master or slave mode. Switching the SPI from master to slave or vice versa forces the SPI system into idle state. 0 SPI is in slave mode 1 SPI is in master mode
3 CPOL	**SPI Clock Polarity Bit** — This bit selects an inverted or non-inverted SPI clock. To transmit data between SPI modules, the SPI modules must have identical CPOL values. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 Active-high clocks selected. In idle state SCK is low. 1 Active-low clocks selected. In idle state SCK is high.
2 CPHA	**SPI Clock Phase Bit** — This bit is used to select the SPI clock format. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 Sampling of data occurs at odd edges (1,3,5,...,15) of the SCK clock 1 Sampling of data occurs at even edges (2,4,6,...,16) of the SCK clock
1 SSOE	**Slave Select Output Enable** — The $\overline{SS}$ output feature is enabled only in master mode, if MODFEN is set, by asserting the SSOE as shown in Table 14-3. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state.
0 LSBFE	**LSB-First Enable** — This bit does not affect the position of the MSB and LSB in the data register. Reads and writes of the data register always have the MSB in bit 7. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 Data is transferred most significant bit first. 1 Data is transferred least significant bit first.

Table 14-3. $\overline{SS}$ Input / Output Selection

MODFEN	SSOE	Master Mode	Slave Mode
0	0	$\overline{SS}$ not used by SPI	$\overline{SS}$ input
0	1	$\overline{SS}$ not used by SPI	$\overline{SS}$ input
1	0	$\overline{SS}$ input with MODF feature	$\overline{SS}$ input
1	1	$\overline{SS}$ is slave select output	$\overline{SS}$ input

Table 16-8. TCTL1/TCTL2 Field Descriptions

Field	Description
7:0 OMx	**Output Mode** — These eight pairs of control bits are encoded to specify the output action to be taken as a result of a successful OCx compare. When either OMx or OLx is 1, the pin associated with OCx becomes an output tied to OCx. **Note:** To enable output action by OMx bits on timer port, the corresponding bit in OC7M should be cleared.
7:0 OLx	**Output Level** — These eight pairs of control bits are encoded to specify the output action to be taken as a result of a successful OCx compare. When either OMx or OLx is 1, the pin associated with OCx becomes an output tied to OCx. **Note:** To enable output action by OLx bits on timer port, the corresponding bit in OC7M should be cleared.

Table 16-9. Compare Result Output Action

OMx	OLx	Action
0	0	Timer disconnected from output pin logic
0	1	Toggle OCx output line
1	0	Clear OCx output line to zero
1	1	Set OCx output line to one

To operate the 16-bit pulse accumulator independently of input capture or output compare 7 and 0 respectively the user must set the corresponding bits IOSx = 1, OMx = 0 and OLx = 0. OC7M7 in the OC7M register must also be cleared.

16.3.2.9 Timer Control Register 3/Timer Control Register 4 (TCTL3 and TCTL4)

	7	6	5	4	3	2	1	0
R W	EDG7B	EDG7A	EDG6B	EDG6A	EDG5B	EDG5A	EDG4B	EDG4A
Reset	0	0	0	0	0	0	0	0

Figure 16-16. Timer Control Register 3 (TCTL3)

	7	6	5	4	3	2	1	0
R W	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A	EDG0B	EDG0A
Reset	0	0	0	0	0	0	0	0

Figure 16-17. Timer Control Register 4 (TCTL4)

Read: Anytime

Write: Anytime.

Table 16-10. TCTL3/TCTL4 Field Descriptions

Field	Description
7:0 EDGnB EDGnA	**Input Capture Edge Control** — These eight pairs of control bits configure the input capture edge detector circuits.

Table 16-11. Edge Detector Circuit Configuration

EDGnB	EDGnA	Configuration
0	0	Capture disabled
0	1	Capture on rising edges only
1	0	Capture on falling edges only
1	1	Capture on any edge (rising or falling)

firmware. The standard BDM firmware watches for serial commands and executes them as they are received.

The firmware commands are shown in Table 18-6.

Table 18-6. Firmware Commands

Command[1]	Opcode (hex)	Data	Description
READ_NEXT	62	16-bit data out	Increment X by 2 (X = X + 2), then read word X points to.
READ_PC	63	16-bit data out	Read program counter.
READ_D	64	16-bit data out	Read D accumulator.
READ_X	65	16-bit data out	Read X index register.
READ_Y	66	16-bit data out	Read Y index register.
READ_SP	67	16-bit data out	Read stack pointer.
WRITE_NEXT	42	16-bit data in	Increment X by 2 (X = X + 2), then write word to location pointed to by X.
WRITE_PC	43	16-bit data in	Write program counter.
WRITE_D	44	16-bit data in	Write D accumulator.
WRITE_X	45	16-bit data in	Write X index register.
WRITE_Y	46	16-bit data in	Write Y index register.
WRITE_SP	47	16-bit data in	Write stack pointer.
GO	08	None	Go to user program. If enabled, ACK will occur when leaving active background mode.
GO_UNTIL[2]	0C	None	Go to user program. If enabled, ACK will occur upon returning to active background mode.
TRACE1	10	None	Execute one user instruction then return to active BDM. If enabled, ACK will occur upon returning to active background mode.
TAGGO	18	None	Enable tagging and go to user program. There is no ACK pulse related to this command.

1 If enabled, ACK will occur when data is ready for transmission for all BDM READ commands and will occur after the write is complete for all BDM WRITE commands.

2 Both WAIT (with clocks to the S12 CPU core disabled) and STOP disable the ACK function. The GO_UNTIL command will not get an Acknowledge if one of these two CPU instructions occurs before the “UNTIL” instruction. This can be a problem for any instruction that uses ACK, but GO_UNTIL is a lot more difficult for the development tool to time-out.

18.4.5 BDM Command Structure

Hardware and firmware BDM commands start with an 8-bit opcode followed by a 16-bit address and/or a 16-bit data word depending on the command. All the read commands return 16 bits of data despite the byte or word implication in the command name.

NOTE

8-bit reads return 16-bits of data, of which, only one byte will contain valid data. If reading an even address, the valid data will appear in the MSB. If reading an odd address, the valid data will appear in the LSB.

18.4.9 SYNC — Request Timed Reference Pulse

The SYNC command is unlike other BDM commands because the host does not necessarily know the correct communication speed to use for BDM communications until after it has analyzed the response to the SYNC command. To issue a SYNC command, the host should perform the following steps:

1. Drive the BKGD pin low for at least 128 cycles at the lowest possible BDM serial communication frequency (the lowest serial communication frequency is determined by the crystal oscillator or the clock chosen by CLKSW.)
2. Drive BKGD high for a brief speedup pulse to get a fast rise time (this speedup pulse is typically one cycle of the host clock.)
3. Remove all drive to the BKGD pin so it reverts to high impedance.
4. Listen to the BKGD pin for the sync response pulse.

Upon detecting the SYNC request from the host, the target performs the following steps:

1. Discards any incomplete command received or bit retrieved.
2. Waits for BKGD to return to a logic 1.
3. Delays 16 cycles to allow the host to stop driving the high speedup pulse.
4. Drives BKGD low for 128 cycles at the current BDM serial communication frequency.
5. Drives a one-cycle high speedup pulse to force a fast rise time on BKGD.
6. Removes all drive to the BKGD pin so it reverts to high impedance.

The host measures the low time of this 128 cycle SYNC response pulse and determines the correct speed for subsequent BDM communications. Typically, the host can determine the correct communication speed 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 falling 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.

18.4.10 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. As soon as 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.

This register is not in the on-chip memory map in expanded and special peripheral modes. Therefore, these accesses will be echoed externally.

NOTE

These bits have no effect when the associated pin(s) are outputs. (The pull resistors are inactive.)

Table 21-9. PUCR Field Descriptions

Field	Description
7 PUPKE	**Pull resistors Port K Enable** 0 Port K pull resistors are disabled. 1 Enable pull resistors for port K input pins.
4 PUPEE	**Pull resistors Port E Enable** 0 Port E pull resistors on bits 7, 4:0 are disabled. 1 Enable pull resistors for port E input pins bits 7, 4:0. **Note:** Pins 5 and 6 of port E have pull resistors which are only enabled during reset. This bit has no effect on these pins.
1 PUPBE	**Pull resistors Port B Enable** 0 Port B pull resistors are disabled. 1 Enable pull resistors for all port B input pins.
0 PUPAE	**Pull resistors Port A Enable** 0 Port A pull resistors are disabled. 1 Enable pull resistors for all port A input pins.

21.3.2.11 Reduced Drive Register (RDRIV)

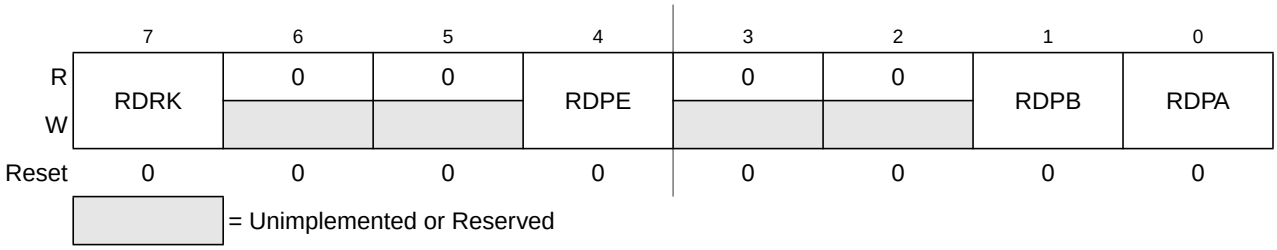

Figure 21-15. Reduced Drive Register (RDRIV)

Read: Anytime (provided this register is in the map)

Write: Anytime (provided this register is in the map)

This register is used to select reduced drive for the pins associated with the core ports. This gives reduced power consumption and reduced RFI with a slight increase in transition time (depending on loading). This feature would be used on ports which have a light loading. The reduced drive function is independent of which function is being used on a particular port.

This register is not in the on-chip memory map in expanded and special peripheral modes. Therefore, these accesses will be echoed externally.

22.3.2.3 Initialization of Internal EEPROM Position Register (INITEE)

	7	6	5	4	3	2	1	0
R	EE15	EE14	EE13	EE12	EE11	0	0	EEON
W								
Reset[1]	—	—	—	—	—	—	—	—

1. The reset state of this register is controlled at chip integration. Please refer to the device overview section to determine the actual reset state of this register.

= Unimplemented or Reserved

Figure 22-5. Initialization of Internal EEPROM Position Register (INITEE)

Read: Anytime

Write: The EEON bit can be written to any time on all devices. Bits E[11:15] are “write anytime in all modes” on most devices. On some devices, bits E[11:15] are “write once in normal and emulation modes and write anytime in special modes”. See device overview chapter to determine the actual write access rights.

NOTE

Writes to this register take one cycle to go into effect.

This register initializes the position of the internal EEPROM within the on-chip system memory map.

Table 22-4. INITEE Field Descriptions

Field	Description
7:3 EE[15:11]	**Internal EEPROM Map Position** — These bits determine the upper five bits of the base address for the system’s internal EEPROM array.
0 EEON	**Enable EEPROM** — This bit is used to enable the EEPROM memory in the memory map. 0 Disables the EEPROM from the memory map. 1 Enables the EEPROM in the memory map at the address selected by EE[15:11].

Appendix G
Detailed Register Map

The following tables show the detailed register map of the MC9S12HZ256.

0x0000–0x000F MEBI Map 1 of 3 (HCS12 Multiplexed External Bus Interface)

Address	Name		7	6	5	4	3	2	1	0
0x0000	PORTA	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0001	PORTB	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0002	DDRA	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0003	DDRB	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0004–0x0007	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0008	PORTE	R	Bit 7	6	5	4	3	2	Bit 1	Bit 0
		W								
0x0009	DDRE	R	Bit 7	6	5	4	3	Bit 2	0	0
		W								
0x000A	PEAR	R	NOACCE	0	PIPOE	NECLK	LSTRE	RDWE	0	0
		W								
0x000B	MODE	R	MODC	MODB	MODA	0	IVIS	0	EMK	EME
		W								
0x000C	PUCR	R	PUPKE	0	0	PUPEE	0	0	PUPBE	PUPAE
		W								
0x000D	RDRIV	R	RDPK	0	0	RDPE	0	0	RDPB	RDPA
		W								
0x000E	EBICTL	R	0	0	0	0	0	0	0	ESTR
		W								
0x000F	Reserved	R	0	0	0	0	0	0	0	0
		W								

0x0010–0x0014 MMC Map 1 of 4 (HCS12 Module Mapping Control)

Address	Name		7	6	5	4	3	2	1	0
0x0010	INITRM	R	RAM15	RAM14	RAM13	RAM12	RAM11	0	0	RAMHAL
		W								
0x0011	INITRG	R	0	REG14	REG13	REG12	REG11	0	0	0
		W								
0x0012	INITEE	R	EE15	EE14	EE13	EE12	EE11	0	0	EEON
		W								
0x0013	MISC	R	0	0	0	0	EXSTR1	EXSTR0	ROMHM	ROMON
		W								
0x0014	MTST0 Test Only	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								

0x00D0–0x00D7 SCI1 (Asynchronous Serial Interface) (continued)

Address	Name		7	6	5	4	3	2	1	0
0x00D5	SCI1SR2	R	0	0	0	TXPOL	RXPOL	BRK13	TXDIR	RAF
		W								
0x00D6	SCI1DRH	R	R8	T8	0	0	0	0	0	0
		W								
0x00D7	SCI1DRL	R	R7	R6	R5	R4	R3	R2	R1	R0
		W	T7	T6	T5	T4	T3	T2	T1	T0

0x00D8–0x00DF SPI (Serial Peripheral Interface)

Address	Name		7	6	5	4	3	2	1	0
0x00D8	SPICR1	R	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
		W								
0x00D9	SPICR2	R	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
		W								
0x00DA	SPIBR	R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
		W								
0x00DB	SPISR	R	SPIF	0	SPTEF	MODF	0	0	0	0
		W								
0x00DC	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x00DD	SPIDR	R	Bit7	6	5	4	3	2	1	Bit0
		W								
0x00DE–0x00DF	Reserved	R	0	0	0	0	0	0	0	0
		W								

0x00E0–0x00FF PWM (Pulse Width Modulator 8 Bit 6 Channel) (Sheet 1 of 2)

Address	Name		7	6	5	4	3	2	1	0
0x00E0	PWME	R	0	0	PWME5	PWME4	PWME3	PWME2	PWME1	PWME0
		W								
0x00E1	PWMPOL	R	0	0	PPOL5	PPOL4	PPOL3	PPOL2	PPOL1	PPOL0
		W								
0x00E2	PWMCLK	R	0	0	PCLK5	PCLK4	PCLK3	PCLK2	PCLK1	PCLK0
		W								
0x00E3	PWMPRCLK	R	0	PCKB2	PCKB1	PCKB0	0	PCKA2	PCKA1	PCKA0
		W								
0x00E4	PWMCAE	R	0	0	CAE5	CAE4	CAE3	CAE2	CAE1	CAE0
		W								
0x00E5	PWMCTL	R	0	CON45	CON23	CON01	PSWAI	PFRZ	0	0
		W								
0x00E6	PWMTST Test Only	R	0	0	0	0	0	0	0	0
		W								
0x00E7	PWMPRSC Test Only	R	0	0	0	0	0	0	0	0
		W								
0x00E8	PWMSCLA	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								