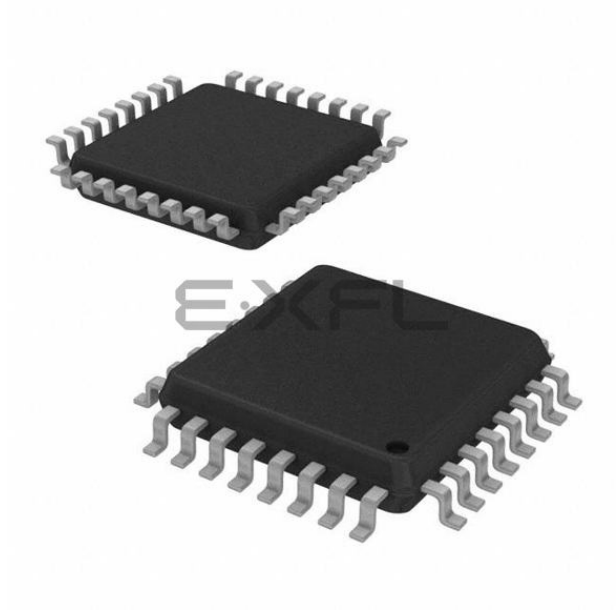


**Welcome to E-XFL.COM**

## 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 | Obsolete |
| Core Processor | 12V1 |
| Core Size | 16-Bit |
| Speed | 25MHz |
| Connectivity | IrDA, LINbus, SCI, SPI |
| Peripherals | LVD, POR, PWM, WDT |
| Number of I/O | 16 |
| Program Memory Size | 64KB (64K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | 512 x 8 |
| RAM Size | 2K x 8 |
| Voltage - Supply (Vcc/Vdd) | 3.13V ~ 5.5V |
| Data Converters | A/D 2x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 105°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 32-LQFP |
| Supplier Device Package | 32-LQFP (7x7) |
| Purchase URL | https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12vr64f2vlc |

**Table 2-16. PTS Register Field Descriptions (continued)**

| Field | Description |
|---|---|
| 1<br>PTS | **PorT data register port S** — General-purpose input/output data, SCI1, routed SCI0 or LPDR[LPDR1]<br>When not used with the alternative function, the associated pin can be used as general-purpose I/O. In general-purpose output mode the register bit value is driven to the pin.<br>If the associated data direction bit is set to 1, a read returns the value of the port register bit, otherwise the synchronized pin input state is read.<br><br>• The SCI1 function takes precedence over the routed SCI0 or LPDR[LPDR1] function and the general-purpose I/O function if enabled.<br>• The routed SCI0 or LPDR[LPDR1] function takes precedence over the general-purpose I/O function if enabled. |
| 0<br>PTS | **PorT data register port S** — General-purpose input/output data, SCI1, routed SCI0<br>When not used with the alternative function, the associated pin can be used as general-purpose I/O. In general-purpose output mode the register bit value is driven to the pin.<br>If the associated data direction bit is set to 1, a read returns the value of the port register bit, otherwise the synchronized pin input state is read.<br><br>• The SCI1 function takes precedence over the routed SCI0 function and the general-purpose I/O function if enabled.<br>• The routed SCI0 function takes precedence over the general-purpose I/O function if enabled. |

## 2.3.18 Port S Input Register (PTIS)

Address 0x0249 Access: User read only[1]

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | PTIS5 | PTIS4 | PTIS3 | PTIS2 | PTIS1 | PTIS0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 2-16. Port S Input Register (PTIS)**

[1] Read: Anytime
Write:Never

**Table 2-17. PTIS Register Field Descriptions**

| Field | Description |
|---|---|
| 5-0<br>PTIS | **PorT Input data register port S** —<br>A read always returns the synchronized input state of the associated pin. It can be used to detect overload or short circuit conditions on output pins. |

## 2.3.41 Port AD Data Direction Register (DDR1AD)

Address 0x0275 Access: User read/write[1]

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | DDR1AD5 | DDR1AD4 | DDR1AD3 | DDR1AD2 | DDR1AD1 | DDR1AD0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 2-40. Port AD Data Direction Register (DDR1AD)**

[1] Read: Anytime
Write: Anytime

**Table 2-42. DDR1AD Register Field Descriptions**

| Field | Description |
|---|---|
| 5-0<br>DDR1AD | **Data Direction Register 1 port AD** —<br>This bit determines whether the associated pin is an input or output.<br>To use the digital input function the ADC Digital Input Enable Register (ATDDIEN) has to be set to logic level “1”.<br><br>1 Associated pin is configured as output<br>0 Associated pin is configured as input |

# 2.4 Functional Description

## 2.4.1 General

Each pin except BKGD and port L pins can act as general-purpose I/O. In addition each pin can act as an output or input of a peripheral module.

## 2.4.2 Registers

Table 2-47 lists the configuration registers which are available on each port. These registers except the pin input and routing registers can be written at any time, however a specific configuration might not become active.

For example selecting a pullup device: This device does not become active while the port is used as a push-pull output.

| Port | Data | Input | Data Direction | Reduced Drive | Pull Enable | Polarity Select | Wired-Or Mode | Interrupt Enable | Interrupt Flag | Routing |
|---|---|---|---|---|---|---|---|---|---|---|
| E | yes | - | yes | - | yes | - | - | - | - | - |
| T | yes | yes | yes | - | yes | yes | - | - | - | yes |
| S | yes | yes | yes | - | yes | yes | yes | - | - | yes |
| P | yes | yes | yes | yes | yes | yes | - | yes | yes | - |
| L | - | yes | yes[1] | - | - | yes | - | yes | yes | - |
| AD | yes | yes | yes | - | yes | yes | - | yes | yes | - |

**Table 2-47. Register availability per port (each cell represents one register with individual configuration bit)**

[1] Input buffer control only

### 2.4.2.1 Data register (PTx)

This register holds the value driven out to the pin if the pin is used as a general-purpose I/O.

Writing to this register has only an effect on the pin if the pin is used as general-purpose output. When reading this address, the synchronized state of the pin is returned if the associated data direction register bit is set to “0”.

If the data direction register bits are set to logic level “1”, the contents of the data register is returned. This is independent of any other configuration (Figure 2-45).

### 2.4.2.2 Input register (PTIx)

This register is read-only and always returns the synchronized state of the pin (Figure 2-45).

### 2.4.2.3 Data direction register (DDRx)

This register defines whether the pin is used as an general-purpose input or an output.

If a peripheral module controls the pin the contents of the data direction register is ignored (Figure 2-45).

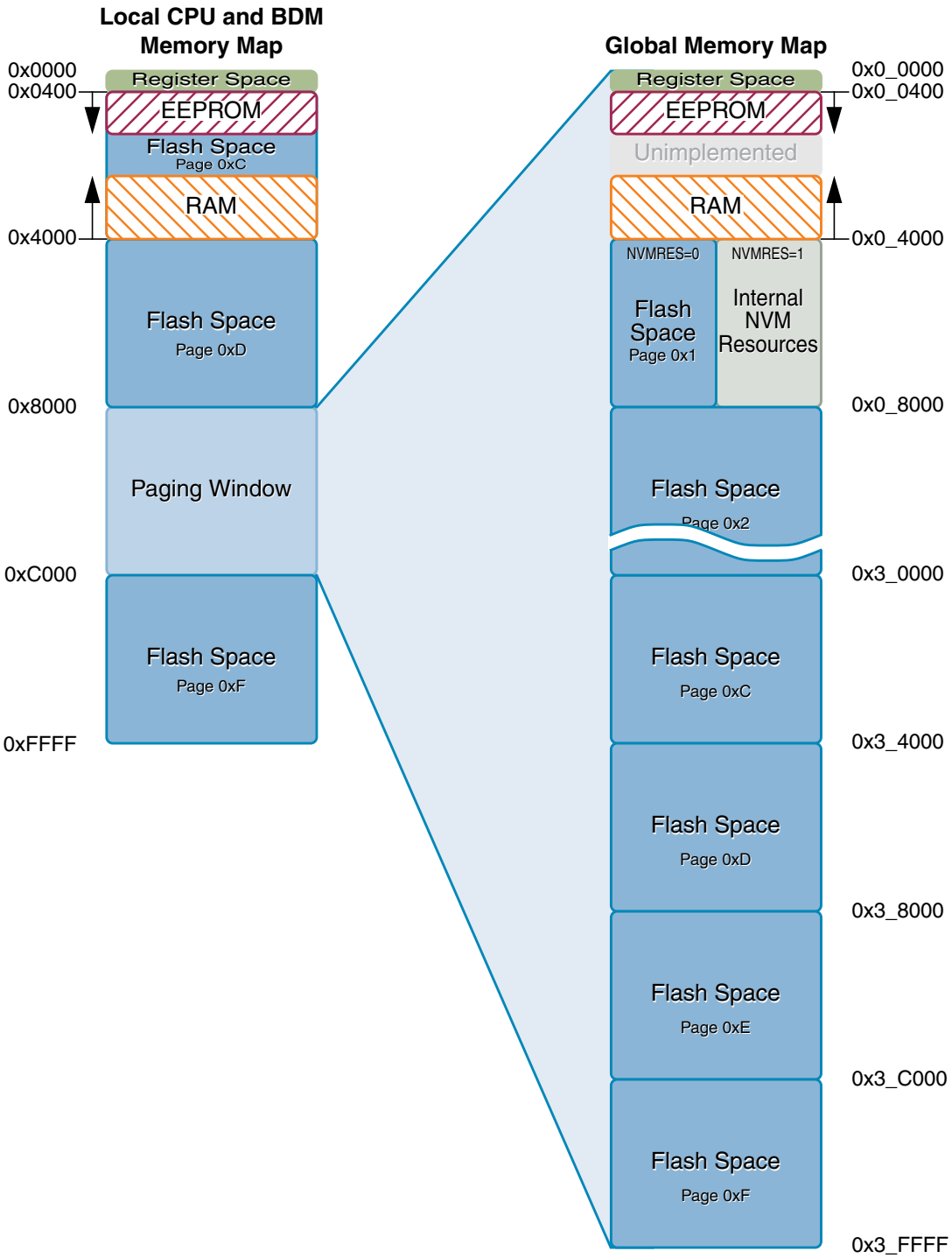



**Figure 3-11. Local to Global Address Mapping**

### 4.3.2.7 S12CPMU_UHV PLL Control Register (CPMUPLL)

This register controls the PLL functionality.

0x003A

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | FM1 | FM0 | 0 | 0 | 0 | 0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 4-10. S12CPMU_UHV PLL Control Register (CPMUPLL)**

Read: Anytime

Write: Anytime if PROT=0 (CPMUPROT register) and PLLSEL=1 (CPMUCLKS register). Else write has no effect.

**NOTE**

Write to this register clears the LOCK and UPOSC status bits.

**NOTE**

Care should be taken to ensure that the bus frequency does not exceed the specified maximum when frequency modulation is enabled.

**Table 4-7. CPMUPLL Field Descriptions**

| Field | Description |
|---|---|
| 5, 4<br>FM1, FM0 | PLL **Frequency Modulation Enable Bits** — FM1 and FM0 enable frequency modulation on the VCOCLK. This is to reduce noise emission. The modulation frequency is $f_{ref}$ divided by 16. See Table 4-8 for coding. |

**Table 4-8. FM Amplitude selection**

| FM1 | FM0 | FM Amplitude / $f_{VCO}$ Variation |
|---|---|---|
| 0 | 0 | FM off |
| 0 | 1 | ±1% |
| 1 | 0 | ±2% |
| 1 | 1 | ±4% |

### 4.3.2.9 S12CPMU_UHV COP Control Register (CPMUCOP)

This register controls the COP (Computer Operating Properly) watchdog.

The clock source for the COP is either ACLK, IRCCLK or OSCCLK depending on the setting of the COPOSCSEL0 and COPOSCSEL1 bit (see also Table 4-6).
In Stop Mode with PSTP=1 (Pseudo Stop Mode), COPOSCSEL0=1 and COPOSCEL1=0 and PCE=1 the COP continues to run, else the COP counter halts in Stop Mode with COPOSCSEL1 =0.
In Full Stop Mode and Pseudo Stop Mode with COPOSCSEL1=1 the COP continues to run.

0x003C

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | WCOP | RSBCK | 0 | 0 | 0 | CR2 | CR1 | CR0 |
| W | | | WRTMASK | | | | | |
| Reset | F | 0 | 0 | 0 | 0 | F | F | F |

After de-assert of System Reset the values are automatically loaded from the Flash memory. See Device specification for details.

= Unimplemented or Reserved

**Figure 4-12. S12CPMU_UHV COP Control Register (CPMUCOP)**

Read: Anytime

Write:

1. RSBCK: Anytime in Special Mode; write to “1” but not to “0” in Normal Mode
2. WCOP, CR2, CR1, CR0:
   — Anytime in Special Mode, when WRTMASK is 0, otherwise it has no effect
   — Write once in Normal Mode, when WRTMASK is 0, otherwise it has no effect.
   – Writing CR[2:0] to “000” has no effect, but counts for the “write once” condition.
   – Writing WCOP to “0” has no effect, but counts for the “write once” condition.

When a non-zero value is loaded from Flash to CR[2:0] the COP time-out period is started.

A change of the COPOSCSEL0 or COPOSCSEL1 bit (writing a different value) or loosing UPOSC status while COPOSCSEL1 is clear and COPOSCSEL0 is set, re-starts the COP time-out period.

In Normal Mode the COP time-out period is restarted if either of these conditions is true:

1. Writing a non-zero value to CR[2:0] (anytime in special mode, once in normal mode) with WRTMASK = 0.
2. Writing WCOP bit (anytime in Special Mode, once in Normal Mode) with WRTMASK = 0.
3. Changing RSBCK bit from “0” to “1”.

In Special Mode, any write access to CPMUCOP register restarts the COP time-out period.

#### 6.4.3.4 Channel Priorities

In case of simultaneous matches the priority is resolved according to Table 6-36. The lower priority is suppressed. It is thus possible to miss a lower priority match if it occurs simultaneously with a higher priority. The priorities described in Table 6-36 dictate that in the case of simultaneous matches, the match pointing to final state has highest priority followed by the lower channel number (0,1,2).

**Table 6-36. Channel Priorities**

| Priority | Source | Action |
|---|---|---|
| **Highest** | **TRIG** | Enter Final State |
| | **Channel pointing to Final State** | Transition to next state as defined by state control registers |
| | **Match0 (force or tag hit)** | Transition to next state as defined by state control registers |
| | **Match1 (force or tag hit)** | Transition to next state as defined by state control registers |
| **Lowest** | **Match2 (force or tag hit)** | Transition to next state as defined by state control registers |

### 6.4.4 State Sequence Control

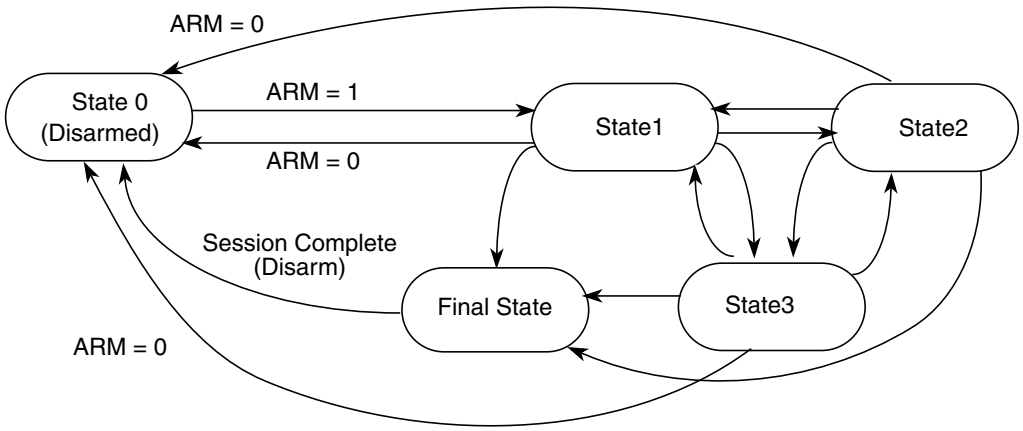



**Figure 6-24. State Sequencer Diagram**

The state sequencer allows a defined sequence of events to provide a trigger point for tracing of data in the trace buffer. Once the DBG module has been armed by setting the ARM bit in the DBGC1 register, then state1 of the state sequencer is entered. Further transitions between the states are then controlled by the state control registers and channel matches. From Final State the only permitted transition is back to the disarmed state0. Transition between any of the states 1 to 3 is not restricted. Each transition updates the SSF[2:0] flags in DBGSR accordingly to indicate the current state.

Alternatively writing to the TRIG bit in DBGSC1, provides an immediate trigger independent of comparator matches.

Independent of the state sequencer, each comparator channel can be individually configured to generate an immediate breakpoint when a match occurs through the use of the BRK bits in the DBGxCTL registers. Thus it is possible to generate an immediate breakpoint on selected channels, whilst a state sequencer transition can be initiated by a match on other channels. If a debug session is ended by a match on a channel the state sequencer transitions through Final State for a clock cycle to state0. This is independent of tracing

```
        RTI                             ;
```

The execution flow taking into account the IRQ is as follows

```
        LDX     #SUB_1
MARK1   JMP     0,X                     ;
IRQ_ISR LDAB    #$F0                    ;
        STAB    VAR_C1
        RTI                             ;
SUB_1   BRN     *
        NOP                             ;
ADDR1   DBNE    A,PART5                 ;
```

#### 6.4.5.2.2 Loop1 Mode

Loop1 Mode, similarly to Normal Mode also stores only COF address information to the trace buffer, it however allows the filtering out of redundant information.

The intent of Loop1 Mode is to prevent the Trace Buffer from being filled entirely with duplicate information from a looping construct such as delays using the DBNE instruction or polling loops using BRSET/BRCLR instructions. Immediately after address information is placed in the Trace Buffer, the DBG module writes this value into a background register. This prevents consecutive duplicate address entries in the Trace Buffer resulting from repeated branches.

Loop1 Mode only inhibits consecutive duplicate source address entries that would typically be stored in most tight looping constructs. It does not inhibit repeated entries of destination addresses or vector addresses, since repeated entries of these would most likely indicate a bug in the user's code that the DBG module is designed to help find.

#### 6.4.5.2.3 Detail Mode

In Detail Mode, address and data for all memory and register accesses is stored in the trace buffer. This mode is intended to supply additional information on indexed, indirect addressing modes where storing only the destination address would not provide all information required for a user to determine where the code is in error. This mode also features information bit storage to the trace buffer, for each address byte storage. The information bits indicate the size of access (word or byte) and the type of access (read or write).

When tracing in Detail Mode, all cycles are traced except those when the CPU is either in a free or opcode fetch cycle.

#### 6.4.5.2.4 Compressed Pure PC Mode

In Compressed Pure PC Mode, the PC addresses of all executed opcodes, including illegal opcodes are stored. A compressed storage format is used to increase the effective depth of the trace buffer. This is achieved by storing the lower order bits each time and using 2 information bits to indicate if a 64 byte boundary has been crossed, in which case the full PC is stored.

Each Trace Buffer row consists of 2 information bits and 18 PC address bits

### 10.3.2.4 SCI Alternative Control Register 1 (SCIACR1)

Module Base + 0x0001

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | RXEDGIE | 0 | 0 | 0 | 0 | 0 | BERRIE | BKDIE |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Figure 10-7. SCI Alternative Control Register 1 (SCIACR1)**

Read: Anytime, if AMAP = 1

Write: Anytime, if AMAP = 1

**Table 10-7. SCIACR1 Field Descriptions**

| Field | Description |
|---|---|
| 7<br>RSEDGIE | **Receive Input Active Edge Interrupt Enable** — RXEDGIE enables the receive input active edge interrupt flag, RXEDGIF, to generate interrupt requests.<br>0 RXEDGIF interrupt requests disabled<br>1 RXEDGIF interrupt requests enabled |
| 1<br>BERRIE | **Bit Error Interrupt Enable** — BERRIE enables the bit error interrupt flag, BERRIF, to generate interrupt requests.<br>0 BERRIF interrupt requests disabled<br>1 BERRIF interrupt requests enabled |
| 0<br>BKDIE | **Break Detect Interrupt Enable** — BKDIE enables the break detect interrupt flag, BKDIF, to generate interrupt requests.<br>0 BKDIF interrupt requests disabled<br>1 BKDIF interrupt requests enabled |

### 10.4.1 Infrared Interface Submodule

This module provides the capability of transmitting narrow pulses to an IR LED and receiving narrow pulses and transforming them to serial bits, which are sent to the SCI. The IrDA physical layer specification defines a half-duplex infrared communication link for exchange data. The full standard includes data rates up to 16 Mbits/s. This design covers only data rates between 2.4 Kbits/s and 115.2 Kbits/s.

The infrared submodule consists of two major blocks: the transmit encoder and the receive decoder. The SCI transmits serial bits of data which are encoded by the infrared submodule to transmit a narrow pulse for every zero bit. No pulse is transmitted for every one bit. When receiving data, the IR pulses should be detected using an IR photo diode and transformed to CMOS levels by the IR receive decoder (external from the MCU). The narrow pulses are then stretched by the infrared submodule to get back to a serial bit stream to be received by the SCI.The polarity of transmitted pulses and expected receive pulses can be inverted so that a direct connection can be made to external IrDA transceiver modules that uses active low pulses.

The infrared submodule receives its clock sources from the SCI. One of these two clocks are selected in the infrared submodule in order to generate either 3/16, 1/16, 1/32 or 1/4 narrow pulses during transmission. The infrared block receives two clock sources from the SCI, R16XCLK and R32XCLK, which are configured to generate the narrow pulse width during transmission. The R16XCLK and R32XCLK are internal clocks with frequencies 16 and 32 times the baud rate respectively. Both R16XCLK and R32XCLK clocks are used for transmitting data. The receive decoder uses only the R16XCLK clock.

#### 10.4.1.1 Infrared Transmit Encoder

The infrared transmit encoder converts serial bits of data from transmit shift register to the TXD pin. A narrow pulse is transmitted for a zero bit and no pulse for a one bit. The narrow pulse is sent in the middle of the bit with a duration of 1/32, 1/16, 3/16 or 1/4 of a bit time. A narrow high pulse is transmitted for a zero bit when TXPOL is cleared, while a narrow low pulse is transmitted for a zero bit when TXPOL is set.

#### 10.4.1.2 Infrared Receive Decoder

The infrared receive block converts data from the RXD pin to the receive shift register. A narrow pulse is expected for each zero received and no pulse is expected for each one received. A narrow high pulse is expected for a zero bit when RXPOL is cleared, while a narrow low pulse is expected for a zero bit when RXPOL is set. This receive decoder meets the edge jitter requirement as defined by the IrDA serial infrared physical layer specification.

### 10.4.2 LIN Support

This module provides some basic support for the LIN protocol. At first this is a break detect circuitry making it easier for the LIN software to distinguish a break character from an incoming data stream. As a further addition is supports a collision detection at the bit level as well as cancelling pending transmissions.

When the third edge occurs, the value previously latched from the serial data input pin is shifted into the LSB or MSB of the SPI shift register, depending on LSBFE bit. After this edge, the next bit of the master data is coupled out of the serial data output pin of the master to the serial input pin on the slave.

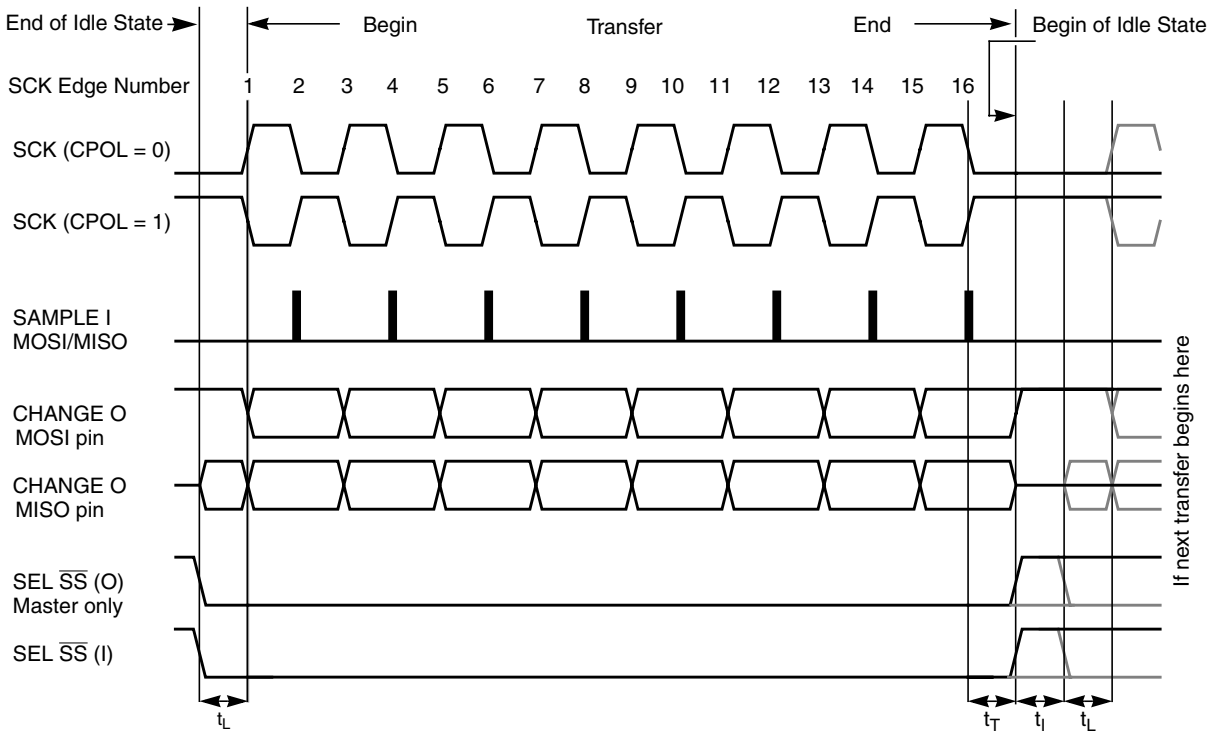
This process continues for a total of $n^1$ edges on the SCK line with data being latched on even numbered edges and shifting taking place on odd numbered edges.

Data reception is double buffered, data is serially shifted into the SPI shift register during the transfer and is transferred to the parallel SPI data register after the last bit is shifted in.

After $2n^1$ SCK edges:

- Data that was previously in the SPI data register of the master is now in the data register of the slave, and data that was in the data register of the slave is in the master.
- The SPIF flag bit in SPISR is set indicating that the transfer is complete.

Figure 11-14 shows two clocking variations for CPHA = 1. The diagram may be interpreted as a master or slave timing diagram because the SCK, MISO, and MOSI pins are connected directly between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The $\overline{SS}$ line is the slave select input to the slave. The $\overline{SS}$ pin of the master must be either high or reconfigured as a general-purpose output not affecting the SPI.




$t_L$ = Minimum leading time before the first SCK edge, not required for back-to-back transfers

$t_T$ = Minimum trailing time after the last SCK edge

$t_I$ = Minimum idling time between transfers (minimum $\overline{SS}$ high time), not required for back-to-back transfers

**Figure 11-14. SPI Clock Format 1 (CPHA = 1), with 8-Bit Transfer Width selected (XFRW = 0)**

### 12.4.1 Prescaler

The prescaler divides the bus clock by 1, 2, 4, 8, 16, 32, 64 or 128. The prescaler select bits, PR[2:0], select the prescaler divisor. PR[2:0] are in timer system control register 2 (TSCR2).

The prescaler divides the bus clock by a prescalar value. Prescaler select bits PR[2:0] of in timer system control register 2 (TSCR2) are set to define a prescalar value that generates a divide by 1, 2, 4, 8, 16, 32, 64 and 128 when the PRNT bit in TSCR1 is disabled.

By enabling the PRNT bit of the TSCR1 register, the performance of the timer can be enhanced. In this case, it is possible to set additional prescaler settings for the main timer counter in the present timer by using PTPSR[7:0] bits of PTPSR register generating divide by 1, 2, 3, 4,....20, 21, 22, 23,.....255, or 256.

### 12.4.2 Input Capture

Clearing the I/O (input/output) select bit, IOSx, configures channel x as an input capture channel. The input capture function captures the time at which an external event occurs. When an active edge occurs on the pin of an input capture channel, the timer transfers the value in the timer counter into the timer channel registers, TCx.

The minimum pulse width for the input capture input is greater than two bus clocks.

An input capture on channel x sets the CxF flag. The CxI bit enables the CxF flag to generate interrupt requests. Timer module or Pulse Accumulator must stay enabled (TEN bit of TSCR1 or PAEN bit of PACTL register must be set to one) while clearing CxF (writing one to CxF).

### 12.4.3 Output Compare

Setting the I/O select bit, IOSx, configures channel x when available as an output compare channel. The output compare function can generate a periodic pulse with a programmable polarity, duration, and frequency. When the timer counter reaches the value in the channel registers of an output compare channel, the timer can set, clear, or toggle the channel pin if the corresponding OCPDx bit is set to zero. An output compare on channel x sets the CxF flag. The CxI bit enables the CxF flag to generate interrupt requests. Timer module or Pulse Accumulator must stay enabled (TEN bit of TSCR1 or PAEN bit of PACTL register must be set to one) while clearing CxF (writing one to CxF).

The output mode and level bits, OMx and OLx, select set, clear, toggle on output compare. Clearing both OMx and OLx results in no output compare action on the output compare channel pin.

Setting a force output compare bit, FOCx, causes an output compare on channel x. A forced output compare does not set the channel flag.

The following channel 7 feature is available only when channel 7 exists. A channel 7 event, which can be a counter overflow when TTOV[7] is set or a successful output compare on channel 7, overrides output compares on all other output compare channels. The output compare 7 mask register masks the bits in the output compare 7 data register. The timer counter reset enable bit, TCRE, enables channel 7 output compares to reset the timer counter. A channel 7 output compare can reset the timer counter even if the IOC7 pin is being used as the pulse accumulator input.

### 15.4.4 Interrupts

The interrupt vector requested by the LIN Physical Layer is listed in Table 15-12. Vector address and interrupt priority is defined at MCU level.

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

**Table 15-12. Interrupt Vectors**

| Module Interrupt Source | Module Internal Interrupt Source | Local Enable |
|---|---|---|
| LIN Interrupt (LPI) | LIN Over-Current Interrupt (LPOCI) | LPOCIE = 1; available only in Normal Mode |

#### 15.4.4.1 Over-Current Interrupt

The output low side FET (transmitter) is protected against over-current. In case of an over-current condition occurring within a time frame called $t_{OCLIM}$ starting from a transition on TXD, the current through the transmitter is limited (the transmitter is not shut down), the transmitted data is lost and the bit LPOC remains at 0. If an over-current occurs out of this time frame, the transmitter is shut down and the bit LPOC in the LPSR register is set as long as the condition is present. The inhibition of an over-current within the time frame $t_{OCLIM}$ is meant to avoid “false” over-current conditions due to charging/discharging the LIN bus during transition phases.

The bit LPOCIF is set to 1 when the status of LPOC changes and it remains set until it has been cleared by writing a 1. If the bit LPOCIE is set in the LPIE register, an interrupt will be requested.

As long as LPOC is 1, the transmitter is disable.

**NOTE**

On entering Standby Mode (stop mode at the device level), the LPOCIF bit is not cleared.

## 15.5 Application Information

### 15.5.1 Over-current handling

In case of an over-current condition, the transmitter is switched off. The transmitter will stay disabled until the condition is gone. At this moment it is up to the software to activate again the transmitter through the RXONLY bit.

However, if the over-current occurs within a transition phase, the transmitter is internally limiting the current but no over-current event will be reported. Indeed, charging/discharging the bus can cause over-current events at each transition, which should not be reported. The time frame during which an over-current is not reported is equal to $t_{OCLIM}$ starting from a rising or a falling edge of txd.

**Table 17-15. FERSTAT Field Descriptions**

| Field | Description |
|---|---|
| 1<br>DFDIF | **Double Bit Fault Detect Interrupt Flag** — The setting of the DFDIF flag indicates that a double bit fault was detected in the stored parity and data bits during a Flash array read operation or that a Flash array read operation returning invalid data was attempted on a Flash block that was under a Flash command operation.[1] The DFDIF flag is cleared by writing a 1 to DFDIF. Writing a 0 to DFDIF has no effect on DFDIF.[2]<br>0 No double bit fault detected<br>1 Double bit fault detected or a Flash array read operation returning invalid data was attempted while command running |
| 0<br>SFDIF | **Single Bit Fault Detect Interrupt Flag** — With the IGNSF bit in the FCNFG register clear, the SFDIF flag indicates that a single bit fault was detected in the stored parity and data bits during a Flash array read operation or that a Flash array read operation returning invalid data was attempted on a Flash block that was under a Flash command operation.[1] The SFDIF flag is cleared by writing a 1 to SFDIF. Writing a 0 to SFDIF has no effect on SFDIF.<br>0 No single bit fault detected<br>1 Single bit fault detected and corrected or a Flash array read operation returning invalid data was attempted while command running |

1 The single bit fault and double bit fault flags are mutually exclusive for parity errors (an ECC fault occurrence can be either single fault or double fault but never both). A simultaneous access collision (Flash array read operation returning invalid data attempted while command running) is indicated when both SFDIF and DFDIF flags are high.

2 There is a one cycle delay in storing the ECC DFDIF and SFDIF fault flags in this register. At least one NOP is required after a flash memory read before checking FERSTAT for the occurrence of ECC errors.

### 17.3.2.9 P-Flash Protection Register (FPROT)

The FPROT register defines which P-Flash sectors are protected against program and erase operations.

The (unreserved) bits of the FPROT register are writable with the restriction that the size of the protected region can only be increased.

During the reset sequence, the FPROT register is loaded with the contents of the P-Flash protection byte in the Flash configuration field at global address 0x3_FF0C located in P-Flash memory (see Table 17-3) as indicated by reset condition ‘F’ in . To change the P-Flash protection that will be loaded during the reset sequence, the upper sector of the P-Flash memory must be unprotected, then the P-Flash protection byte must be reprogrammed. If a double bit fault is detected while reading the P-Flash phrase containing the P-Flash protection byte during the reset sequence, the FPOPEN bit will be cleared and remaining bits in the FPROT register will be set to leave the P-Flash memory fully protected.

Trying to alter data in any protected area in the P-Flash memory will result in a protection violation error and the FPVIOL bit will be set in the FSTAT register. The block erase of a P-Flash block is not possible if any of the P-Flash sectors contained in the same P-Flash block are protected.

**Table 17-16. FPROT Field Descriptions**

| Field | Description |
|---|---|
| 7<br>FPOPEN | **Flash Protection Operation Enable** — The FPOPEN bit determines the protection function for program or erase operations as shown in Table 17-17 for the P-Flash block.<br>0 When FPOPEN is clear, the FPHDIS and FPLDIS bits define unprotected address ranges as specified by the corresponding FPHS and FPLS bits<br>1 When FPOPEN is set, the FPHDIS and FPLDIS bits enable protection for the address range specified by the corresponding FPHS and FPLS bits |
| 6<br>RNV[6] | **Reserved Nonvolatile Bit** — The RNV bit should remain in the erased state for future enhancements. |
| 5<br>FPHDIS | **Flash Protection Higher Address Range Disable** — The FPHDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory ending with global address 0x3_FFFF.<br>0 Protection/Unprotection enabled<br>1 Protection/Unprotection disabled |
| 4–3<br>FPHS[1:0] | **Flash Protection Higher Address Size** — The FPHS bits determine the size of the protected/unprotected area in P-Flash memory as shown inTable 17-18. The FPHS bits can only be written to while the FPHDIS bit is set. |
| 2<br>FPLDIS | **Flash Protection Lower Address Range Disable** — The FPLDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory beginning with global address 0x3_8000.<br>0 Protection/Unprotection enabled<br>1 Protection/Unprotection disabled |
| 1–0<br>FPLS[1:0] | **Flash Protection Lower Address Size** — The FPLS bits determine the size of the protected/unprotected area in P-Flash memory as shown in Table 17-19. The FPLS bits can only be written to while the FPLDIS bit is set. |

**Table 17-17. P-Flash Protection Function**

| FPOPEN | FPHDIS | FPLDIS | Function[1] |
|---|---|---|---|
| 1 | 1 | 1 | No P-Flash Protection |
| 1 | 1 | 0 | Protected Low Range |
| 1 | 0 | 1 | Protected High Range |
| 1 | 0 | 0 | Protected High and Low Ranges |
| 0 | 1 | 1 | Full P-Flash Memory Protected |
| 0 | 1 | 0 | Unprotected Low Range |
| 0 | 0 | 1 | Unprotected High Range |
| 0 | 0 | 0 | Unprotected High and Low Ranges |

[1] For range sizes, refer to Table 17-18 and Table 17-19.

**Table 17-18. P-Flash Protection Higher Address Range**

| FPHS[1:0] | Global Address Range | Protected Size |
|---|---|---|
| 00 | 0x3_F800–0x3_FFFF | 2 Kbytes |
| 01 | 0x3_F000–0x3_FFFF | 4 Kbytes |
| 10 | 0x3_E000–0x3_FFFF | 8 Kbytes |
| 11 | 0x3_C000–0x3_FFFF | 16 Kbytes |

**Table 17-61. Program EEPROM Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] < 010 at command launch |
| | | Set if CCOBIX[2:0] > 101 at command launch |
| | | Set if command not available in current mode (see Table 17-26) |
| | | Set if an invalid global address [17:0] is supplied |
| | | Set if a misaligned word address is supplied (global address [0] != 0) |
| | | Set if the requested group of words breaches the end of the EEPROM block |
| | FPVIOL | Set if the selected area of the EEPROM memory is protected |
| | MGSTAT1 | Set if any errors have been encountered during the verify operation |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the verify operation |

## 17.4.6.16 Erase EEPROM Sector Command

The Erase EEPROM Sector operation will erase all addresses in a sector of the EEPROM block.

**Table 17-62. Erase EEPROM Sector Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x12 | Global address [17:16] to identify EEPROM block |
| 001 | Global address [15:0] anywhere within the sector to be erased. See Section 17.1.2.2 for EEPROM sector size. | |

Upon clearing CCIF to launch the Erase EEPROM Sector command, the Memory Controller will erase the selected Flash sector and verify that it is erased. The CCIF flag will set after the Erase EEPROM Sector operation has completed.

**Table 17-63. Erase EEPROM Sector Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 001 at command launch |
| | | Set if command not available in current mode (see Table 17-26) |
| | | Set if an invalid global address [17:0] is suppliedsee ) |
| | | Set if a misaligned word address is supplied (global address [0] != 0) |
| | FPVIOL | Set if the selected area of the EEPROM memory is protected |
| | MGSTAT1 | Set if any errors have been encountered during the verify operation |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the verify operation |

The Verify Backdoor Access Key command is monitored by the Memory Controller and an illegal key will prohibit future use of the Verify Backdoor Access Key command. A reset of the MCU is the only method to re-enable the Verify Backdoor Access Key command. The security as defined in the Flash security byte (0x3_FF0F) is not changed by using the Verify Backdoor Access Key command sequence. The backdoor keys stored in addresses 0x3_FF00-0x3_FF07 are unaffected by the Verify Backdoor Access Key command sequence. The Verify Backdoor Access Key command sequence has no effect on the program and erase protections defined in the Flash protection register, FPROT.

After the backdoor keys have been correctly matched, the MCU will be unsecured. After the MCU is unsecured, the sector containing the Flash security byte can be erased and the Flash security byte can be reprogrammed to the unsecure state, if desired. In the unsecure state, the user has full control of the contents of the backdoor keys by programming addresses 0x3_FF00-0x3_FF07 in the Flash configuration field.

### 17.5.2 Unsecuring the MCU in Special Single Chip Mode using BDM

A secured MCU can be unsecured in special single chip mode by using the following method to erase the P-Flash and EEPROM memory:

1. Reset the MCU into special single chip mode
2. Delay while the BDM executes the Erase Verify All Blocks command write sequence to check if the P-Flash and EEPROM memories are erased
3. Send BDM commands to disable protection in the P-Flash and EEPROM memory
4. Execute the Erase All Blocks command write sequence to erase the P-Flash and EEPROM memory. Alternatively the Unsecure Flash command can be executed, if so the steps 5 and 6 below are skeeped.
5. After the CCIF flag sets to indicate that the Erase All Blocks operation has completed, reset the MCU into special single chip mode
6. Delay while the BDM executes the Erase Verify All Blocks command write sequence to verify that the P-Flash and EEPROM memory are erased

If the P-Flash and EEPROM memory are verified as erased, the MCU will be unsecured. All BDM commands will now be enabled and the Flash security byte may be programmed to the unsecure state by continuing with the following steps:

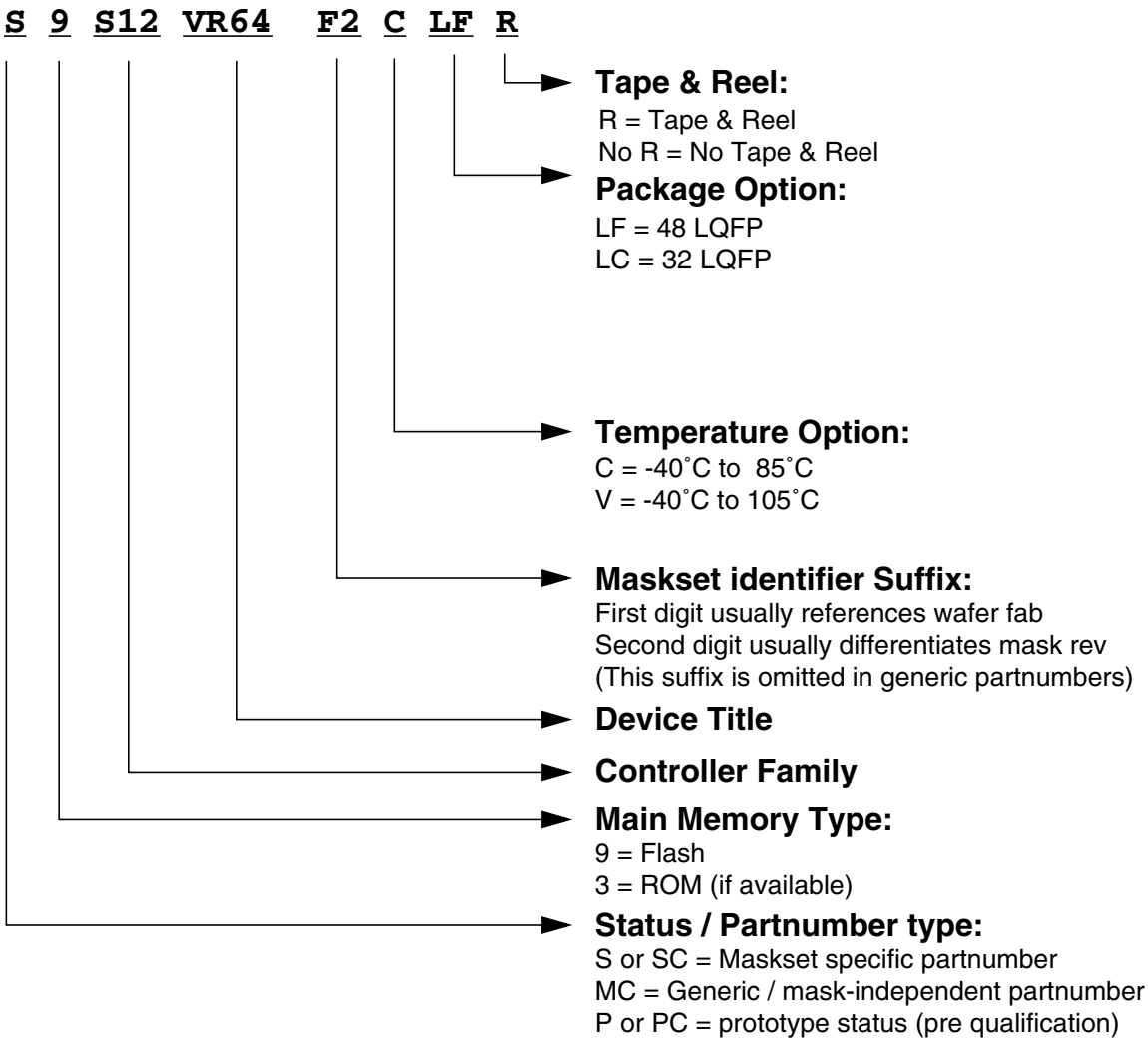
7. Send BDM commands to execute the Program P-Flash command write sequence to program the Flash security byte to the unsecured state
8. Reset the MCU

### 17.5.3 Mode and Security Effects on Flash Command Availability

The availability of Flash module commands depends on the MCU operating mode and security state as shown in Table 17-26.

| Characteristics noted under conditions 7V ≤ VSUP ≤ 18 V, -40˚C ≤ $T_J$ ≤ 150˚C unless otherwise noted[1]. Typical values noted reflect the approximate parameter mean at $T_A$ = 25˚C under nominal conditions unless otherwise noted. | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Num** | **C** | **Ratings** | **Symbol** | **Min** | **Typ** | **Max** | **Unit** |
| 6 | P | Duty cycle 1 | D1 | 0.396 | | | |
| 7 | P | Duty cycle 2 | D2 | | | 0.581 | |
| LIN PHYSICAL LAYER: DRIVER CHARACTERISTICS FOR SLOW SLEW RATE - 10.4KBIT/S | | | | | | | |
| 8 | T | Rising/falling edge time (min to max / max to min) | $t_{rise}$ | | 17 | | µs |
| 9 | T | Over-current masking window (IRC trimmed at 1MHz) | $t_{OCLIM}$ | 31 | | 33 | µs |
| 10 | P | Duty cycle 3 | D3 | 0.417 | | | |
| 11 | P | Duty cycle 4 | D4 | | | 0.590 | |
| LIN PHYSICAL LAYER: DRIVER CHARACTERISTICS FOR FAST MODE SLEW RATE - 100KBIT/S UP TO 250KBIT/S | | | | | | | |
| 12 | T | Rising/falling edge time (min to max / max to min) | $t_{rise}$ | | 0.8 | | µs |
| 13 | T | Over-current masking window (IRC trimmed at 1MHz) | $t_{OCLIM}$ | 5 | | 7 | µs |

[1]For 3.5V<=VSUP<7V, the LINPHY is still working but with degraded parametrics.


S 9 S12 VR64 F2 C LF R
Tape & Reel:
R = Tape & Reel
No R = No Tape & Reel
Package Option:
LF = 48 LQFP
LC = 32 LQFP
Temperature Option:
C = -40˚C to 85˚C
V = -40˚C to 105˚C
Maskset identifier Suffix:
First digit usually references wafer fab
Second digit usually differentiates mask rev
(This suffix is omitted in generic partnumbers)
Device Title
Controller Family
Main Memory Type:
9 = Flash
3 = ROM (if available)
Status / Partnumber type:
S or SC = Maskset specific partnumber
MC = Generic / mask-independent partnumber
P or PC = prototype status (pre qualification)

# Appendix P
# Detailed Register Address Map

## P.1 Detailed Register Map

The following tables show the detailed register map of the MC9S12VR64.

### 0x0000-0x0009 Port Integration Module (PIM) Map 1 of 4

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| | | W | | | | | | | | |
| 0x0000 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0001 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0002 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0003 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0004 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0005 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0006 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0007 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0008 | PORTE | R | 0 | 0 | 0 | 0 | 0 | 0 | PE1 | PE0 |
| | | W | | | | | | | | |
| 0x0009 | DDRE | R | 0 | 0 | 0 | 0 | 0 | 0 | DDRE1 | DDRE0 |
| | | W | | | | | | | | |

### 0x000A-0x000B Module Mapping Conrol (MMC) Map 1 of 2

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x000A | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x000B | MODE | R | MODC | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |