
**Welcome to E-XFL.COM**

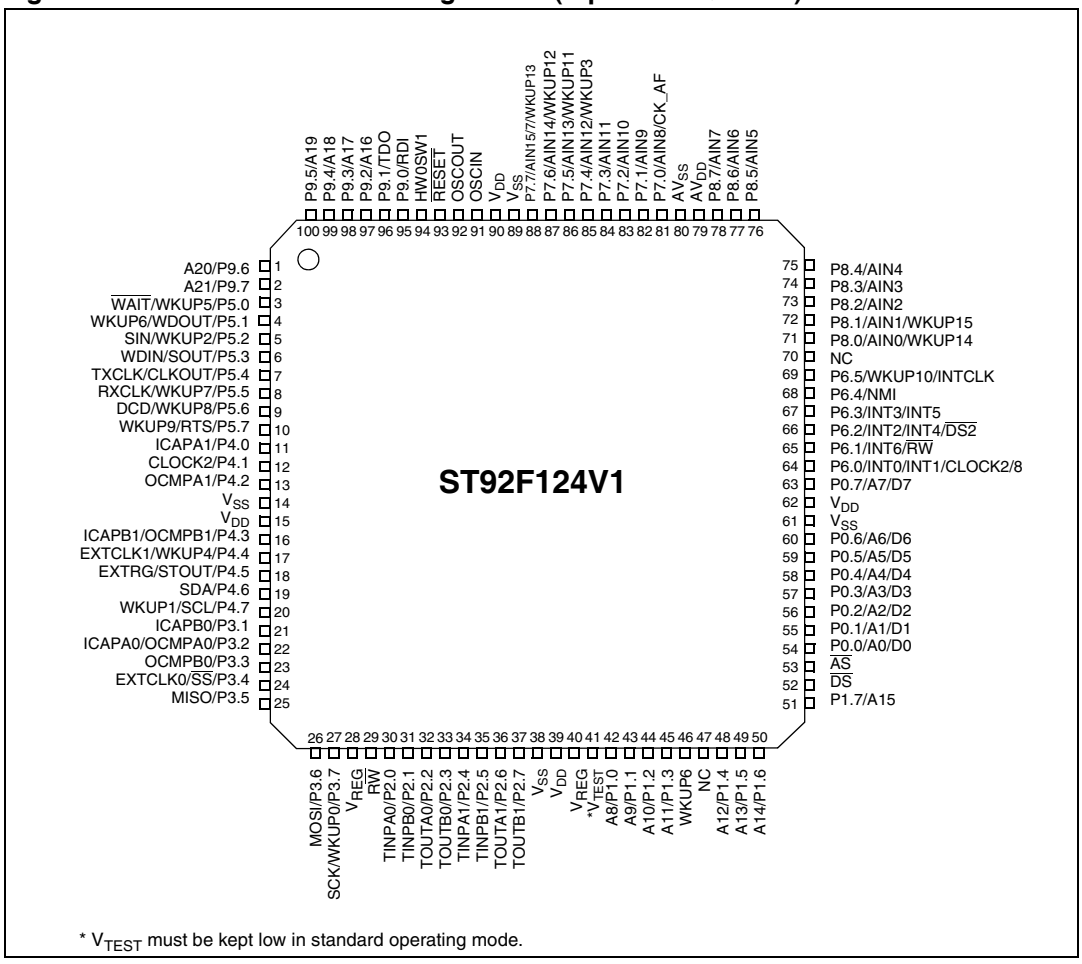
## What is "Embedded - Microcontrollers"?

"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"

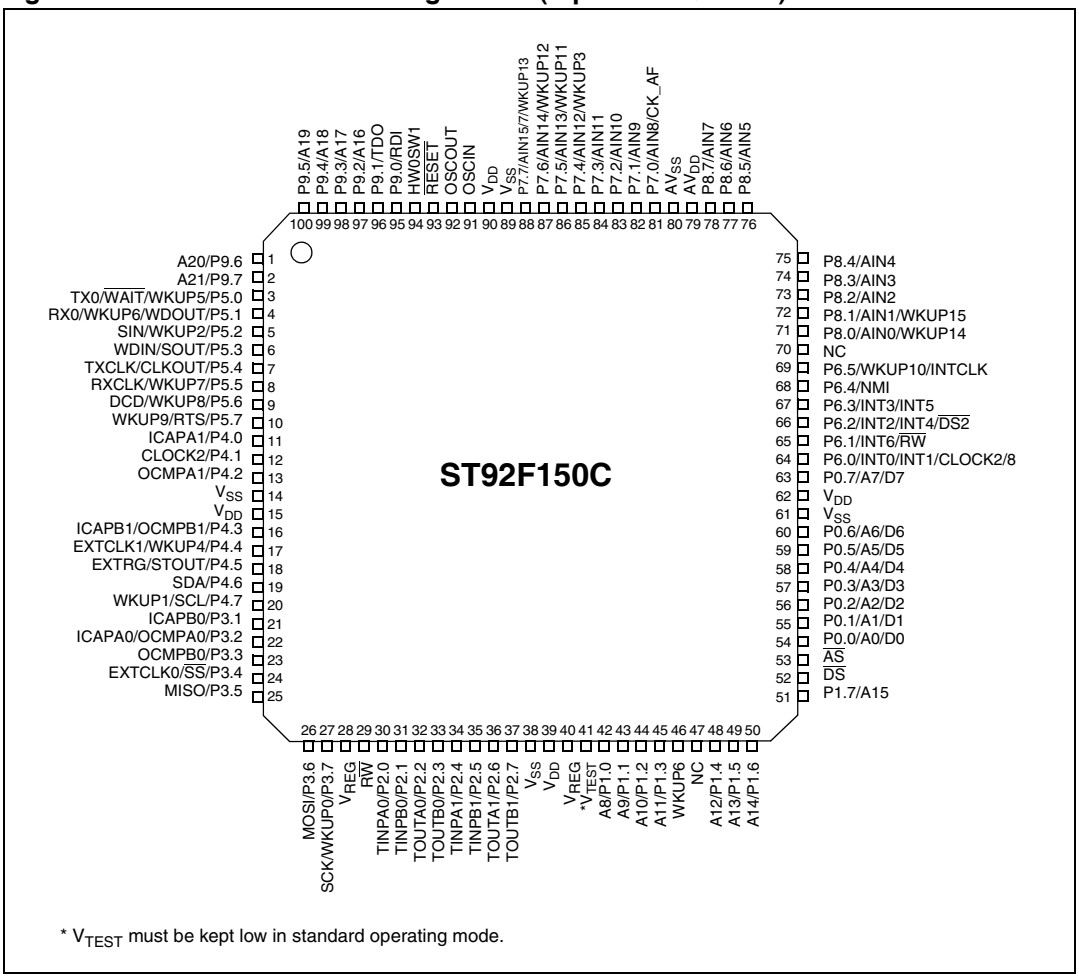
| Details | |
|---|---|
| Product Status | Obsolete |
| Core Processor | ST9 |
| Core Size | 8/16-Bit |
| Speed | 24MHz |
| Connectivity | EBI/EMI, I²C, LINbus, SCI, SPI |
| Peripherals | DMA, LVD, POR, PWM, WDT |
| Number of I/O | 77 |
| Program Memory Size | 128KB (128K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | 1K x 8 |
| RAM Size | 4K x 8 |
| Voltage - Supply (Vcc/Vdd) | 4.5V ~ 5.5V |
| Data Converters | A/D 16x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 105°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 100-QFP |
| Supplier Device Package | 100-PQFP (20x14) |
| Purchase URL | https://www.e-xfl.com/product-detail/stmicroelectronics/st92f124v1qb |

**Figure 8. ST92F124V1: Pin configuration (top-view LQFP100)**




* $V_{TEST}$ must be kept low in standard operating mode.

**Figure 12. ST92F150C: Pin configuration (top-view LQFP100)**




* $V_{TEST}$ must be kept low in standard operating mode.

**Table 6. I/O port alternate functions (continued)**

| Port name | Pin No. | | | Alternate functions | | |
|---|---|---|---|---|---|---|
| | LQFP64 | PQFP100 | LQFP100 | | | |
| P2.6 | 24 | 39 | 36 | TOUTA1 | O | Multifunction Timer 1 - Output A |
| P2.7 | 25 | 40 | 37 | TOUTB1 | O | Multifunction Timer 1 - Output B |
| P3.0 (2) | - | 73 | 70 | | | |
| P3.1 | - | 24 | 21 | ICAPB0 | I | Ext. Timer 0 - Input Capture B |
| P3.2 | - | 25 | 22 | ICAPA0 | I | Ext. Timer 0 - Input Capture A |
| | | | | OCMPA0 | O | Ext. Timer 0 - Output Compare A |
| P3.3 | - | 26 | 23 | OCMPB0 | O | Ext. Timer 0 - Output Compare B |
| P3.4 | - | 27 | 24 | EXTCLK0 | I | Ext. Timer 0 - Input Clock |
| | | | | SS | I | SPI - Slave Select |
| P3.5 | 14 | 28 | 25 | MISO | I/O | SPI - Master Input/Slave Output Data |
| P3.6 | 15 | 29 | 26 | MOSI | I/O | SPI - Master Output/Slave Input Data |
| P3.7 | 16 | 30 | 27 | SCK | I | SPI - Serial Input Clock |
| | | | | WKUP0 | I | Wake-up Line 0 |
| | | | | SCK | O | SPI - Serial Output Clock |
| P4.0 | - | 14 | 11 | ICAPA1 | I | Ext. Timer 1 - Input Capture A |
| P4.1 | - | 15 | 12 | CLOCK2 | O | CLOCK2 internal signal |
| P4.2 | - | 16 | 13 | OCMPA1 | O | Ext. Timer 1 - Output Compare A |
| P4.3 | - | 19 | 16 | ICAPB1 | I | Ext. Timer 1 - Input Capture B |
| | | | | OCMPB1 | O | Ext. Timer 1 - Output Compare B |
| P4.4 | - | 20 | 17 | EXTCLK1 | I | Ext. Timer 1 - Input Clock |
| | | | | WKUP4 | I | Wake-up Line 4 |
| P4.5 | 10 | 21 | 18 | EXTRG | I | ADC Ext. Trigger |
| | | | | STOUT | O | Standard Timer Output |
| P4.6 | 11 | 22 | 19 | SDA0 | I/O | $I^2C$ 0 Data |
| P4.7 | 12 | 23 | 20 | WKUP1 | I | Wake-up Line 1 |
| | | | | SCL0 | I/O | $I^2C$ 0 Clock |
| P5.0 | 1 | 6 | 3 | WAIT | I | External Wait Request |
| | | | | WKUP5 | I | Wake-up Line 5 |
| | | | | TX0 2) | O | CAN 0 output |
| P5.1 | 2 | 7 | 4 | WKUP6 | I | Wake-up Line 6 |
| | | | | RX0 (2) | I | CAN 0 input |
| | | | | WDOUT | O | Watchdog Timer Output |

**Figure 39. ST92F150/F250 external memory map**

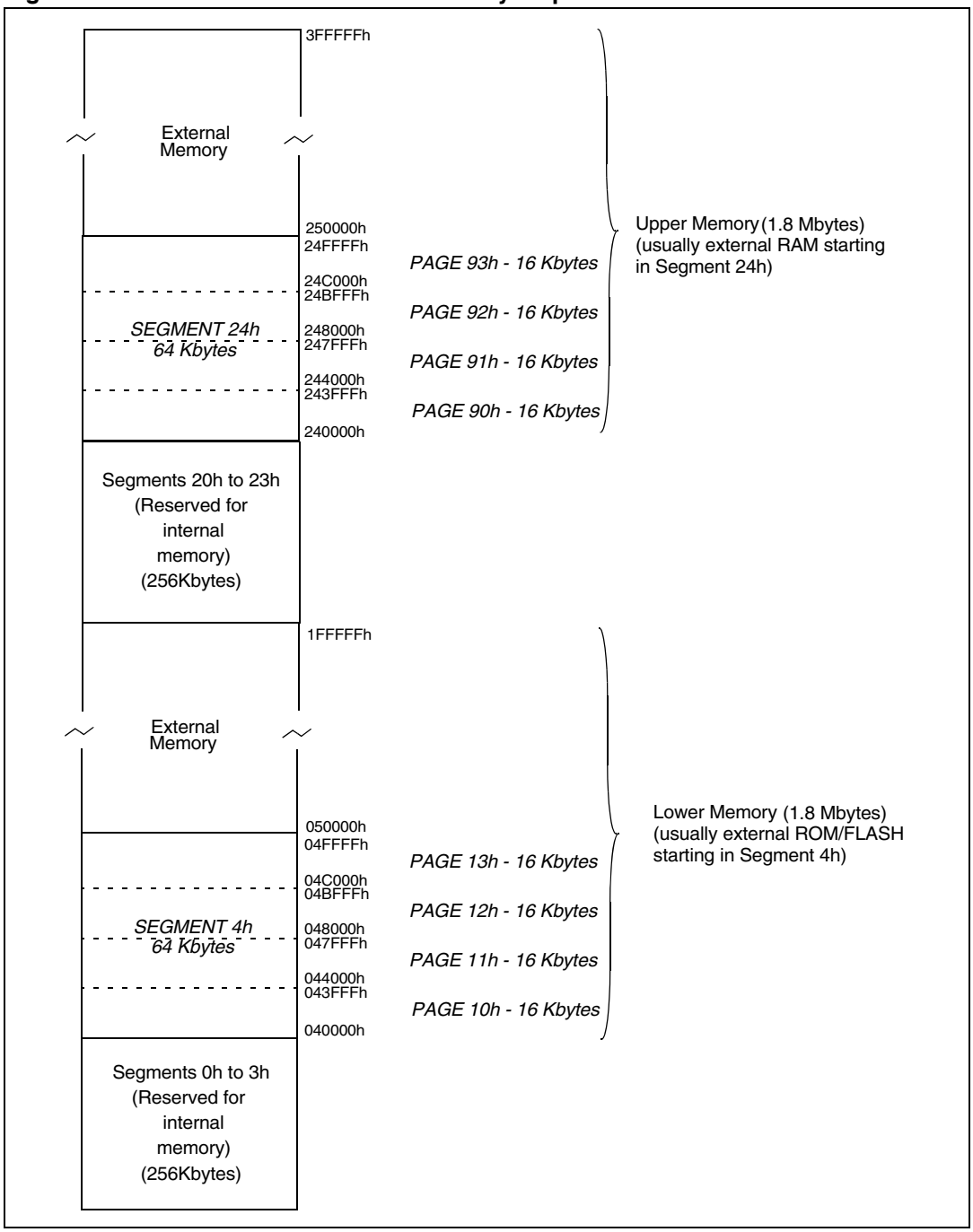


**Figure 40. ST92F124/F150/F250 TESTFLASH and E$^{3}$ TM memory map**

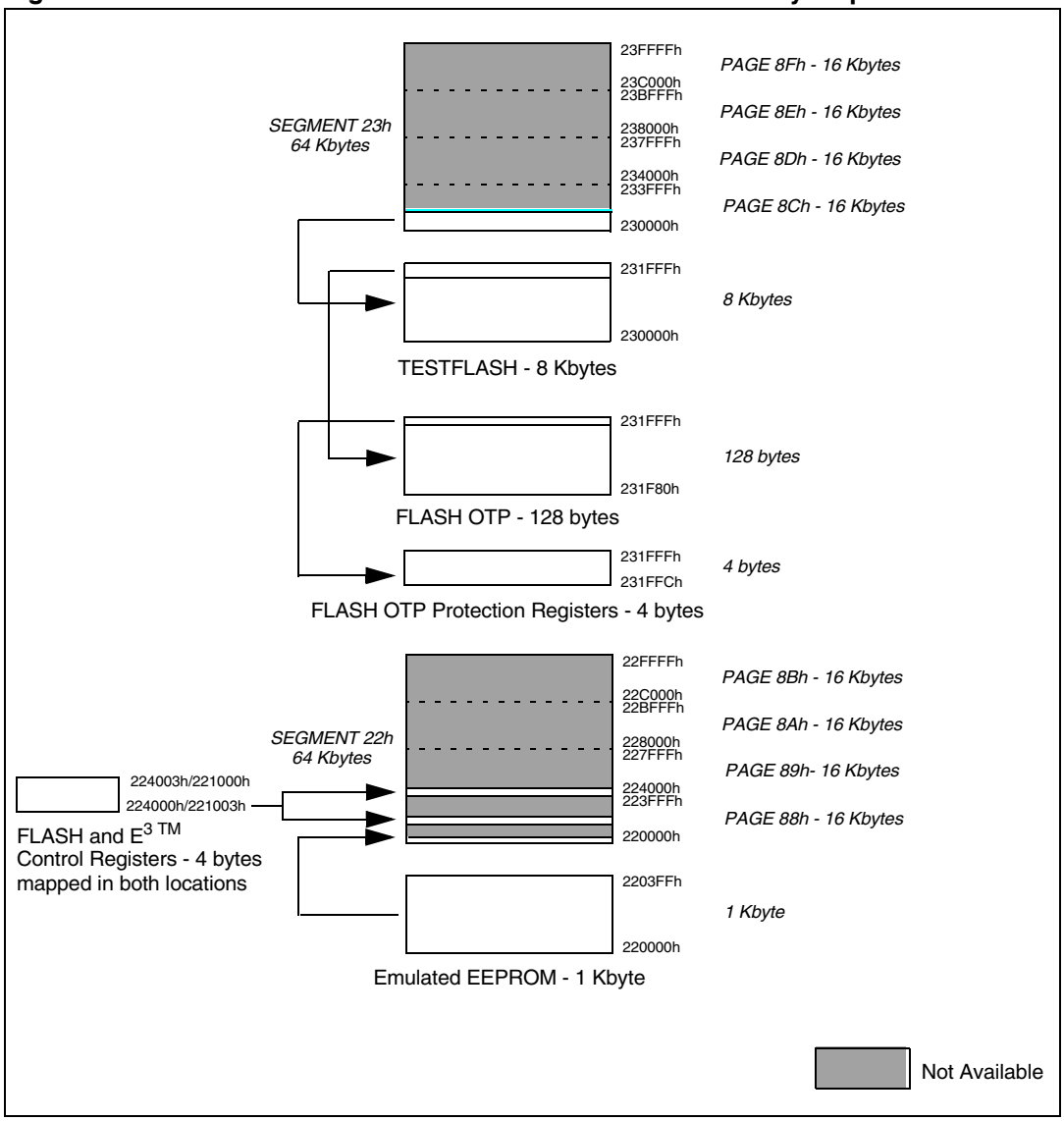


**Table 20. Detailed register map (continued)**

| Page (Dec) | Block | Reg. no. | Register name | Description | Reset value Hex. | Doc. page |
|---|---|---|---|---|---|---|
| 41 | CAN1 * Tx Mailbox 2 | R240 | MCSR | Mailbox Control Status Register | 00 | *437* |
| | | R241 | MDLC | Mailbox Data Length Control Register | x0 | *440* |
| | | R242 | MIDR0 | Mailbox Identifier Register 0 | xx | *439* |
| | | R243 | MIDR1 | Mailbox Identifier Register 1 | xx | *439* |
| | | R244 | MIDR2 | Mailbox Identifier Register 2 | xx | *439* |
| | | R245 | MIDR3 | Mailbox Identifier Register 3 | xx | *439* |
| | | R246 | MDAR0 | Mailbox Data Register 0 | xx | *440* |
| | | R247 | MDAR1 | Mailbox Data Register 1 | xx | *440* |
| | | R248 | MDAR2 | Mailbox Data Register 2 | xx | *440* |
| | | R249 | MDAR3 | Mailbox Data Register 3 | xx | *440* |
| | | R250 | MDAR4 | Mailbox Data Register 4 | xx | *440* |
| | | R251 | MDAR5 | Mailbox Data Register 5 | xx | *440* |
| | | R252 | MDAR6 | Mailbox Data Register 6 | xx | *440* |
| | | R253 | MDAR7 | Mailbox Data Register 7 | xx | *440* |
| | | R254 | MTSLR | Mailbox Time Stamp Low Register | xx | *441* |
| | | R255 | MTSHR | Mailbox Time Stamp High Register | xx | *441* |
| 42 | CAN1 * Filters | See *Page mapping for CAN 0 / CAN 1 on page 448* | | Filter Configuration Acceptance Filters 7:0 (5 register pages) | | |
| 43 | I/O Port 8 * | R248 | P8C0 | Port 8 Configuration Register 0 | 03 | *195* |
| | | R249 | P8C1 | Port 8 Configuration Register 1 | 00 | |
| | | R250 | P8C2 | Port 8 Configuration Register 2 | 00 | |
| | | R251 | P8DR | Port 8 Data Register | FF | |
| | I/O Port 9 * | R252 | P9C0 | Port 9 Configuration Register 0 | 00 | |
| | | R253 | P9C1 | Port 9 Configuration Register 1 | 00 | |
| | | R254 | P9C2 | Port 9 Configuration Register 2 | 00 | |
| | | R255 | P9DR | Port 9 Data Register | FF | |

### INTERRUPT MASK REGISTER LOW (SIMRL)

R246 - Read/Write
Register Page: 60
Reset value: 0000 0000 (00h)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| IMH1 | IMH0 | IMG1 | IMG0 | IMF1 | IMF0 | IME1 | IME0 |

Bits 7:0 = **IMxx** *Channel E to H Mask bits*
The IMxx bits are set and cleared by software to enable or disable on channel xx interrupts.

0: Interrupt masked

1: An interrupt is generated if the corresponding IPxx bit is set in the SIPRL register.

### INTERRUPT TRIGGER EVENT REGISTER HIGH (SITRH)

R247 - Read/Write
Register Page: 60
Reset value: 0000 0000 (00h)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | - | - | - | - | ITEI0 |

Bits 7:1 = Reserved.

Bit 0 = **ITEI0** *Channel I0 Trigger Event*
This bit is set and cleared by software to define the polarity of the channel I0 trigger event

0: The I0 pending bit will be set on the falling edge of the interrupt line

1: The I0 pending bit will be set on the rising edge of the interrupt line

*Note: The ITEI0 bit must be set to enable the SCI-A interrupt as the SCI-A interrupt event is a rising edge event.*

### INTERRUPT TRIGGER EVENT REGISTER LOW (SITRL)

R248 - Read/Write
Register Page: 60
Reset value: 0000 0000 (00h)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| ITEH1 | ITEH0 | ITEG1 | ITEG0 | ITEF1 | ITEF0 | ITEE1 | ITEE0 |

Bits 7:0 = **ITExx** *Channel E to H Trigger Event*
The ITExx bits are set and cleared by software to define the polarity of the channel xx trigger event

0: The corresponding pending bit will be set on the falling edge of the interrupt line

1: The corresponding pending bit will be set on the rising edge of the interrupt line

It should be noted that selecting a non-existent CK_AF clock source is impossible, since such a selection requires that the auxiliary clock source be actually present and selected. In no event can a non-existent clock source be selected inadvertently.

It is up to the user program to switch back to a faster clock on the occurrence of an interrupt, taking care to respect the oscillator and PLL stabilization delays, as appropriate.

It should be noted that any of the low power modes may also be selected explicitly by the user program even when not in Wait for Interrupt mode, by setting the appropriate bits.

If the FREEN bit is set, the PLL is not stopped during Low Power WFI, increasing power consumption.

### 11.3.6 Interrupt generation

System clock selection modifies the CLKCTL and CLK_FLAG registers.

The clock control unit generates an external interrupt request (INTD0) in the following conditions:

- when CK_AF and CLOCK2/16 are selected or deselected as system clock source,
- when the system clock restarts after a hardware stop (when the STOP MODE feature is available on the specific device).
- when the PLL loses the programmed frequency in which it was locked, and when it re-locks

This interrupt can be masked by resetting the INT_SEL bit in the CLKCTL register. Note that this is the only case in the ST9 where an interrupt is generated with a high to low transition.

*Note: This register contains bits which relate to other functions; these are described in the chapter dealing with Device Architecture. Only those bits relating to Clock functions are described here.*

Bit 5 = **DIV2**: *Crystal Oscillator Clock Divided by 2*.
This bit controls the divide by 2 circuit which operates on CLOCK1.

0: No division of CLOCK1

1: CLOCK1 is internally divided by 2

Bits 4:2 = **PRS[2:0]**: *Clock Prescaling*.
These bits define the prescaler value used to prescale CPUCLK from INTCLK. When these three bits are reset, the CPUCLK is not prescaled, and is equal to INTCLK; in all other cases, the internal clock is prescaled by the value of these three bits plus one.

**CLOCK CONTROL REGISTER (CLKCTL)**
R240 - Read/Write
Register Page: 55

Reset Value: 0000 0000 (00h)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| INT_SEL | - | - | - | SRESEN | CKAF_SEL | WFI_CKSEL | LPOWFI |

Bit 7 = **INT_SEL**: *Interrupt Selection*.

0: The external interrupt channel input signal is selected (Reset state)

1: Select the internal RCCU interrupt as the source of the interrupt request

Bits 6:4 = **Reserved for test purposes**
Must be kept reset for normal operation.

Bit 3 = **SRESEN**: *Software Reset Enable.*

0: The HALT instruction turns off the quartz, the PLL and the CCU

1: A Reset is generated when HALT is executed

Bit 2 = **CKAF_SEL**: *Alternate Function Clock Select.*

0: CK_AF clock not selected

1: Select CK_AF clock

*Note: To check if the selection has actually occurred, check that CKAF_ST is set. If no clock is present on the CK_AF pin, the selection will not occur.*

Bit 1 = **WFI_CKSEL**: *WFI Clock Select*.
This bit selects the clock used in Low power WFI mode if LPOWFI = 1.

0: INTCLK during WFI is CLOCK2/16

Bit 1 = **IA0S**: *Interrupt Channel A0 Selection.*
This bit is set and cleared by software.

0: Watchdog End of Count is INTA0 source

1: External Interrupt pin is INTA0 source

**Warning: To avoid spurious interrupt requests, the IA0S bit should be accessed only when the interrupt logic is disabled (i.e. after the DI instruction). It is also necessary to clear any possible interrupt pending requests on channel A0 before enabling this interrupt channel. A delay instruction (e.g. a NOP instruction) must be inserted between the reset of the interrupt pending bit and the IA0S write instruction.**

Other bits are described in the Interrupt section.

## 14.2 Standard timer (STIM)

### 14.2.1 Introduction

The Standard Timer includes a programmable 16-bit down counter and an associated 8-bit prescaler with Single and Continuous counting modes capability. The Standard Timer uses an output (STOUT) pin. This pin may be independent pin or connected as Alternate Function of an I/O port bit.

STOUT can be used to generate a Square Wave or Pulse Width Modulated signal.

The Standard Timer is composed of a 16-bit down counter with an 8-bit prescaler. The input clock to the prescaler can be driven either by an internal clock equal to INTCLK divided by 4, or by CLOCK2/1024 derived directly from the external oscillator, thus providing a stable time reference independent from the PLL programming (refer to *Figure 88*).

The Standard Timer End Of Count condition is able to generate an interrupt which is connected to one of the external interrupt channels.

The End of Count condition is defined as the Counter Underflow, whenever 00h is reached.

0: Disable compare on CMP0R interrupt

1: Enable compare on CMP0R interrupt (or Comp0 DMA End of Block interrupt if CM0D=1)

Bit 1 = **CM1I**: *Compare 1 Interrupt mask*.
This bit is set and cleared by software.

0: Disable compare on CMP1R interrupt

1: Enable compare on CMP1R interrupt

Bit 0 = **OUI**:
*Overflow/Underflow interrupt mask*.
This bit is set and cleared by software.

0: Disable Overflow/Underflow interrupt

1: Enable Overflow/Underflow interrupt

**DMA COUNTER POINTER REGISTER (DCPR)**
R240 - Read/Write
Register Page: 9
Reset value: undefined

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| DCP7 | DCP6 | DCP5 | DCP4 | DCP3 | DCP2 | DMA SRCE | REG/ MEM |

Bits 7:2 = **DCP[7:2]**: *MSBs of DMA counter register address.*
These are the most significant bits of the DMA counter register address programmable by software. The DCP2 bit may also be toggled by hardware if the Timer DMA section for the Compare 0 channel is configured in Swap mode.

Bit 1 = **DMA-SRCE**: *DMA source selection.*
This bit is set and cleared by hardware.

0: DMA source is a Capture on REG0R register

1: DMA destination is a Compare on CMP0R register

Bit 0 = **REG/MEM**: *DMA area selection*.

This bit is set and cleared by software. It selects the source and destination of the DMA area

0: DMA from/to memory

1: DMA from/to Register File

**DMA ADDRESS POINTER REGISTER (DAPR)**
R241 - Read/Write

**Figure 129. SCI-M baud rate generator initialization sequence**

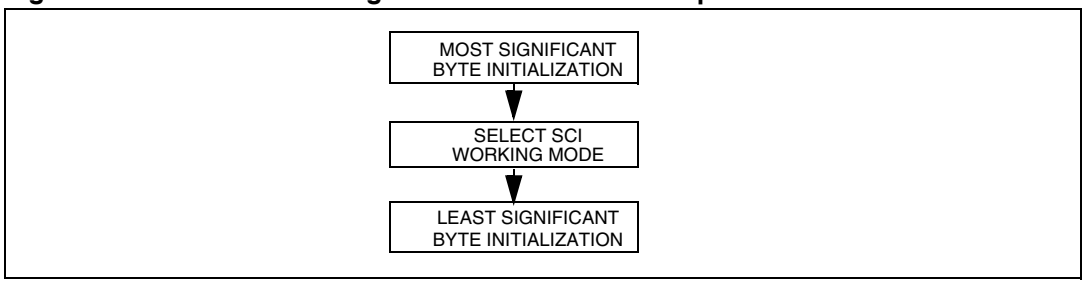



**Table 65. SCI-M baud rate generator divider values example 1**

| INTCLK: 19660.800 KHz | | | | | | | |
|---|---|---|---|---|---|---|---|
| Baud rate | clock factor | Desired freq (kHz) | Divisor | | Actual baud rate | Actual freq (kHz) | Deviation |
| | | | Dec | Hex | | | |
| 50.00 | 16 X | 0.80000 | 24576 | 6000 | 50.00 | 0.80000 | 0.0000% |
| 75.00 | 16 X | 1.20000 | 16384 | 4000 | 75.00 | 1.20000 | 0.0000% |
| 110.00 | 16 X | 1.76000 | 11170 | 2BA2 | 110.01 | 1.76014 | -0.00081% |
| 300.00 | 16 X | 4.80000 | 4096 | 1000 | 300.00 | 4.80000 | 0.0000% |
| 600.00 | 16 X | 9.60000 | 2048 | 800 | 600.00 | 9.60000 | 0.0000% |
| 1200.00 | 16 X | 19.20000 | 1024 | 400 | 1200.00 | 19.20000 | 0.0000% |
| 2400.00 | 16 X | 38.40000 | 512 | 200 | 2400.00 | 38.40000 | 0.0000% |
| 4800.00 | 16 X | 76.80000 | 256 | 100 | 4800.00 | 76.80000 | 0.0000% |
| 9600.00 | 16 X | 153.60000 | 128 | 80 | 9600.00 | 153.60000 | 0.0000% |
| 19200.00 | 16 X | 307.20000 | 64 | 40 | 19200.00 | 307.20000 | 0.0000% |
| 38400.00 | 16 X | 614.40000 | 32 | 20 | 38400.00 | 614.40000 | 0.0000% |
| 76800.00 | 16 X | 1228.80000 | 16 | 10 | 76800.00 | 1228.80000 | 0.0000% |

**Table 66. SCI-M baud rate generator divider values example 2**

| INTCLK: 24576 KHz | | | | | | | |
|---|---|---|---|---|---|---|---|
| Baud rate | Clock factor | Desired freq (kHz) | Divisor | | Actual baud rate | Actual freq (kHz) | Deviation |
| | | | Dec | Hex | | | |
| 50.00 | 16 X | 0.80000 | 30720 | 7800 | 50.00 | 0.80000 | 0.0000% |
| 75.00 | 16 X | 1.20000 | 20480 | 5000 | 75.00 | 1.20000 | 0.0000% |
| 110.00 | 16 X | 1.76000 | 13963 | 383B | 110.01 | 1.76014 | -0.00046% |
| 300.00 | 16 X | 4.80000 | 5120 | 1400 | 300.00 | 4.80000 | 0.0000% |
| 600.00 | 16 X | 9.60000 | 2560 | A00 | 600.00 | 9.60000 | 0.0000% |
| 1200.00 | 16 X | 19.20000 | 1280 | 500 | 1200.00 | 19.20000 | 0.0000% |
| 2400.00 | 16 X | 38.40000 | 640 | 280 | 2400.00 | 38.40000 | 0.0000% |

Each of these three interrupt sources has a pending bit (IERRP, IRXP, ITXP) in the I2CISR register that is set by hardware when the corresponding interrupt event occurs. An interrupt request is performed only if the corresponding mask bit is set (IERRM, IRXM, ITXM) in the I2CIMR register and the peripheral has a proper priority level.
The pending bit has to be reset by software.

*Note: Until the pending bit is reset (while the corresponding mask bit is set), the peripheral processes an interrupt request. So, if at the end of an interrupt routine the pending bit is not reset, another interrupt request is performed.*

*Before the end of the transmission and reception interrupt routines, the I2CSR1.BTF flag bit should be checked, to acknowledge any interrupt requests that occurred during the interrupt routine and to avoid masking subsequent interrupt requests.*

*The "Error" event interrupt pending bit (I2CISR.IERRP) is forced high when the error event flags are set (ADD10, ADSL and SB flags of the I2CSR1 register; SCLF, ADDTX, AF, STOPF, ARLO and BERR flags of the I2CSR2 register).*

*Moreover the Transmitting End Of Block interrupt has the same priority as the "Peripheral Ready to Transmit" interrupt and the Receiving End Of Block interrupt has the same priority as the "Data received" interrupt.*

### 14.8.6 DMA features

The peripheral can use the ST9+ on-chip Direct Memory Access (DMA) channels to provide high-speed data transaction between the peripheral and contiguous locations of Register File, and Memory. The transactions can occur from and toward the peripheral. The maximum number of transactions that each DMA channel can perform is 222 if the register file is selected or 65536 if memory is selected. The control of the DMA features is performed using registers placed in the peripheral register page (I2CISR, I2CIMR, I2CRDAP, I2CRDC, I2CTDAP, I2CTDC).

Each DMA transfer consists of three operations:

- A load from/to the peripheral data register (I2CDR) to/from a location of Register File/Memory addressed through the DMA Address Register (or Register pair)
- A post-increment of the DMA Address Register (or Register pair)
- A post-decrement of the DMA transaction counter, which contains the number of transactions that have still to be performed.

The priority level of the DMA features of the $I^2C$ interface with respect to the other peripherals and the CPU is the same as programmed in the I2CISR register for the interrupt sources. In the internal priority level order of the peripheral, the "Error" interrupt sources have higher priority, followed by DMA, "Data received" and "Receiving End Of Block" interrupts, "Peripheral Ready to Transmit" and "Transmitting End Of Block".
Refer to the Interrupt and DMA chapters for details on the priority levels.

The DMA features are enabled by setting the corresponding enabling bits (RXDM, TXDM) in the I2CIMR register. It is possible to select also the direction of the DMA transactions.

Once the DMA transfer is completed (the transaction counter reaches 0 value), an interrupt request to the CPU is generated. This kind of interrupt is called "End Of Block". The peripheral sends two different "End Of Block" interrupts depending on the direction of the DMA (Receiving End Of Block - Transmitting End Of Block). These interrupt sources have dedicated interrupt pending bits in the I2CIMR register (REOBP, TEOBP) and they are mapped on the same interrupt vectors as respectively "Data Received" and "Peripheral

Bits 6:0 = **CC[6:0]** *9-bit divider programming*
Implementation of a programmable clock divider. These bits and the CC[8:7] bits of the I2CECCR register select the speed of the bus ($F_{SCL}$) depending on the $I^2C$ mode.
They are not cleared when the interface is disabled (I2CCR.PE=0).

Refer to the Electrical Characteristics section for the table of values (*Table 133 on page 493*).

*Note: The programmed frequency is available with no load on SCL and SDA pins.*

### $I^2C$ OWN ADDRESS REGISTER 1 (I2COAR1)

R244 - Read / Write
Register Page: 20 (I2C_0) or 22 (I2C_1)
Reset Value: 0000 0000 (00h)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| ADD7 | ADD6 | ADD5 | ADD4 | ADD3 | ADD2 | ADD1 | ADD0 |

7-bit Addressing Mode

Bits 7:1 = **ADD[7:1]** *Interface address*.
These bits define the $I^2C$ bus address of the interface.
They are not cleared when the interface is disabled (I2CCR.PE=0).

Bit 0 = **ADD0** *Address direction bit.*
This bit is don't care; the interface acknowledges either 0 or 1.
It is not cleared when the interface is disabled (I2CCR.PE=0).

*Note: Address 01h is always ignored.*

10-bit Addressing Mode

Bits 7:0 = **ADD[7:0]** *Interface address*.
These are the least significant bits of the $I^2C$bus address of the interface.
They are not cleared when the interface is disabled (I2CCR.PE=0).

### $I^2C$ OWN ADDRESS REGISTER 2 (I2COAR2)

R245 - Read / Write
Register Page: 20 (I2C_0) or 22 (I2C_1)
Reset Value: 0000 0000 (00h)

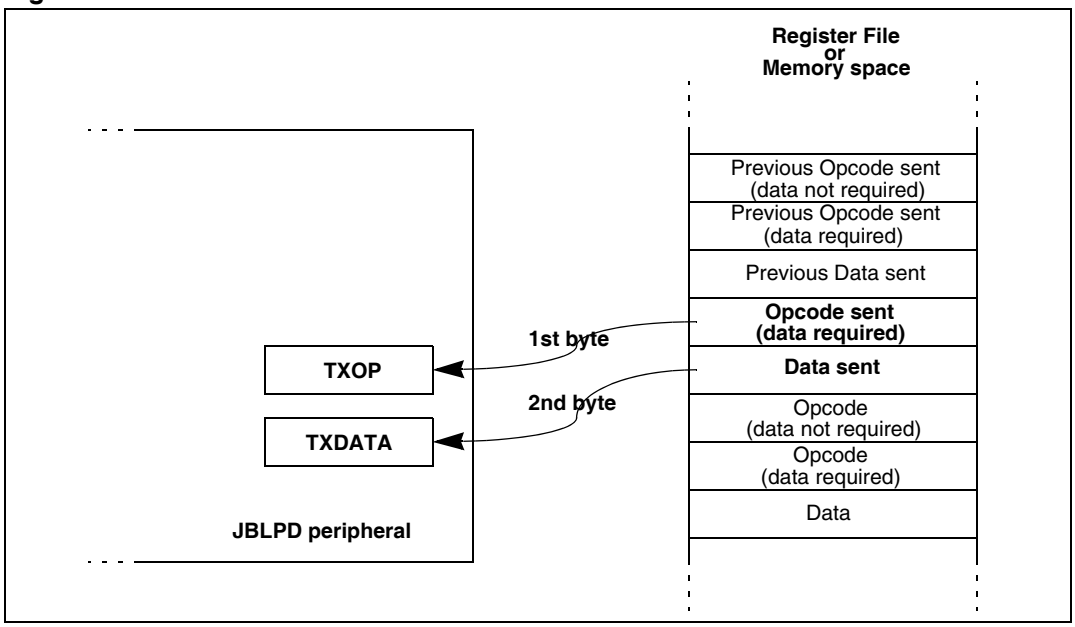
| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| FREQ1 | FREQ0 | EN10BIT | FREQ2 | 0 | ADD9 | ADD8 | 0 |

Bits 7:6,4 = **FREQ[2:0]** *Frequency bits.*

IMPORTANT: To guarantee correct operation, set these bits before enabling the interface (while I2CCR.PE=0).

These bits can be set only when the interface is disabled (I2CCR.PE=0). To configure the interface to $I^2C$ specified delays, select the value corresponding to the microcontroller internal frequency INTCLK

**Figure 153. DMA in transmission mode**




## 14.9.7 Register description

The JBLPD peripheral uses 48 registers that are mapped in a single page of the ST9 register file.

Twelve registers are mapped from R240 (F0h) to R251 (FBh): these registers are usually used to control the JBLPD. See *Section : Un-stacked registers* for a detailed description of these registers.

Thirty-six registers are mapped from R252 (FCh) to R255 (FFh). This is obtained by creating 9 sub-pages, each containing 4 registers, mapped in the same register addresses; 4 bits (RSEL[3:0]) of a register (OPTIONS) are used to select the current sub-page. See *Section : Stacked registers* section for a detailed description of these registers.

The ST9 Register File page used is 23 (17h).

*Note: Bits marked as "Reserved" should be left at their reset value to guarantee software compatibility with future versions of the JBLPD.*

Bit 7 = **TTO** *Transmitter Timeout Flag*
The TTO bit is set when the VPWO pin has been in a logic one (or active) state for longer than 1 ms. This flag is the output of a diagnostic circuit based on the prescaled system clock input. If the 4X bit is not set, the TTO will trip if the VPWO is constantly active for 1000 prescaled clock cycles. If the 4X bit is set, then the TTO will timeout at 4000 prescaled clock cycles. When the TTO flag is set then the diagnostic circuit will disable the VPWO signal, and disable the JBLPD peripheral. The user program must then clear the JE bit to remove the TTO error. It can then retry the block by setting the JE bit again.
The TTO bit can be used to determine if the external J1850 bus is shorted low. Since the transmitter looks for proper edges returned at the VPWI pin for its timing, a lack of edges seen at VPWI when trying to transmit (assuming the RBRK does not get set) would indicate a constant low condition. The user program can take appropriate actions to test the J1850 bus circuit when a TTO occurs.

*Note: A transmit attempt must occur to detect a bus shorted low condition.*
*The TTO bit is cleared while the CONTROL.JE bit is reset or while the CONTROL.JDIS bit is set. TTO is cleared on reset.*

0: VPWO line at 1 for less than 1 ms

1: VPWO line at 1 for longer than 1 ms

Bit 6 = **TDUF** *Transmitter Data Underflow.*
The TDUF will be set to a logic one if the transmitter expects more information to be transmitted, but a TXOP write has not occurred in time (by the end of transmission of the last bit).
The transmitter knows to expect more information from the user program when transmitting messages or type 3 IFRs only. If an opcode is written to TXOP that does not include appending a CRC byte, then the JBLPD peripheral assumes more data is to be written. When the JBLPD peripheral has shifted out the data byte it must have the next data byte in time to place it directly next to it. If the user program does not place new data in the TXDATA register and write the TXOP register with a proper opcode, then the CRC byte which is being kept tabulated by the transmitter is logically inverted and transmitted out the VPWO pin. This will ensure that listeners will detect this message as an error. In this case the TDUF bit is set to a logic one.
TDUF is cleared by reading the ERROR register with TDUF set. TDUF is also cleared on reset, while the CONTROL.JE bit is reset or while the CONTROL.JDIS bit is set.

0: No transmitter data underflow condition occurred

1: Transmitter data underflow condition occurred

Bit 5 = **RDOF** *Receiver Data Overflow*
The RDOF gets set to a logic one if the data in the RXDATA register has not been read and new data is ready to be transferred to the RXDATA register. The old RXDATA information is lost since it is overwritten with new data.
RDOF is cleared by reading the ERROR register with RDOF set, while the CONTROL.JE bit is reset or while the CONTROL.JDIS bit is set, or on reset.

0: No receiver data overflow condition occurred

1: Receiver data overflow condition occurred

Bit 4 = **TRA** *Transmit Request Aborted*
The TRA gets set to a logic one if a transmit opcode is aborted by the JBLPD state machine. Many conditions may cause a TRA. They are explained in the transmit opcode section. If the TRA bit gets set after a TXOP write, then a transmit is not attempted, and the TRDY bit is not cleared.
If a TRA error condition occurs, then the requested transmit is aborted, and the JBLPD peripheral takes appropriate measures as described under the TXOP register section.
TRA is cleared on reset, while the CONTROL.JE bit is reset or while the CONTROL.JDIS bit is set.

0: No transmission request aborted

1: Transmission request aborted

Bit 3 = **RBRK** *Received Break Symbol Flag*
The RBRK gets set to a logic one if a valid break (BRK) symbol is detected from the filtered VPWI pin. A Break received from the J1850 bus will cancel queued transmits of all types. The RBRK bit remains set as long as the break character is detected from the VPWI. Reads of the ERROR register will not clear the RBRK bit as long as a break character is being received. Once the break character is gone, a final read of the ERROR register clears this bit.
An RBRK error occurs once for a frame if it is received during a frame. Afterwards, the receiver is disabled from receiving information (other than the break) until an EOFM symbol is received.
RBRK bit is cleared on reset, while the CONTROL.JE bit is reset or while the CONTROL.JDIS bit is set.
The RBRK bit can be used to detect J1850 bus shorted high conditions. If RBRK is read as a logic high multiple times before an EOFM occurs, then a possible bus shorted high condition exists. The user program can take appropriate measures to test the bus if this condition occurs.

*Note: This bit does not necessarily clear when ERROR is read.*

0: No valid Break symbol received

1: Valid Break symbol received

Bit 2 = **CRCE** *Cyclic Redundancy Check Error*
The receiver section always keeps a running tab of the CRC of all data bytes received from the VPWI since the last EOD symbol. The CRC check is performed when a valid EOD symbol is received both after a message string (subsequent to an SOF symbol) and after an IFR3 string (subsequent to an NB0 symbol). If the received CRC check fails, then the CRCE bit is set to a logic one. CRC errors are inhibited if the JBLPD peripheral is in the “sleep or filter and NOT presently transmitting” mode. A CRC error occurs once for a frame. Afterwards, the receiver is disabled until an EOFM symbol is received and queued transmits for the present frame are cancelled (but the TRA bit is not set). CRCE is cleared when ERROR is read. It is also cleared while the CONTROL.JE bit is reset or while the CONTROL.JDIS bit is set, or on reset.

0: No CRC error detected

1: CRC error detected

### JBLPD RECEIVER DMA TRANSACTION COUNTER REGISTER (RDCPR)

R253 - RSEL[3:0]=0000b
Register Page: 23
Reset Value: xxxx xxxx (xxh)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RF/MEM |

To select this register, the RSEL[3:0] bits of the OPTIONS register must be reset

Bit 7:1 = **RC[7:1]** *Receiver DMA Counter Pointer.*
RDCPR contains the address of the pointer (in the Register File) of the DMA receiver transaction counter when the DMA between Peripheral and Memory Space is selected. Otherwise, if the DMA between Peripheral and Register File is selected, this register points to a pair of registers that are used as DMA Address register and DMA Transaction Counter. See *DMA between JBLPD and register file* and *DMA between JBLPD and memory space* for more details on the use of this register.

Bit 0 = **RF/MEM** *Receiver Register File/Memory Selector.*
If this bit is set to “1”, then the Register File will be selected as Destination, otherwise the Memory space will be used.

0: Receiver DMA with Memory space

1: Receiver DMA with Register File

### JBLPD TRANSMITTER DMA ADDRESS POINTER REGISTER (TDAPR)

R254 - RSEL[3:0]=0000b
Register Page: 23
Reset Value: xxxx xxxx (xxh)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| TA7 | TA6 | TA5 | TA4 | TA3 | TA2 | TA1 | PS |

To select this register, the RSEL[3:0] bits of the OPTIONS register must be reset

Bit 7:1 = **TA[7:1]** *Transmitter DMA Address Pointer.*
TDAPR contains the address of the pointer (in the Register File) of the Transmitter DMA data source when the DMA between the Memory Space and the peripheral is selected. Otherwise, when the DMA between Register File and the peripheral is selected, this register has no meaning.
See *DMA between JBLPD and memory space* for more details on the use of this register.

Bit 0 = **PS** *Memory Segment Pointer Selector.*
This bit is set and cleared by software. It is only meaningful if TDCPR.RF/MEM = 1.

0: The ISR register is used to extend the address of data transmitted by DMA (see MMU chapter)

1: The DMASR register is used to extend the address of data transmitted by DMA (see MMU chapter)

Bit 5 = **TERR** *Transmission Error*
- Read/Clear
This bit is updated by hardware after each transmission attempt.

0: The previous transmission was successful
1: The previous transmission failed due to an error

Bit 4 = **ALST** *Arbitration Lost*
- Read/Clear
This bit is updated by hardware after each transmission attempt.

0: The previous transmission was successful
1: The previous transmission failed due to an arbitration lost

Bit 3 = **TXOK** *Transmission OK*
- Read/Clear
The hardware updates this bit after each transmission attempt.

0: The previous transmission failed
1: The previous transmission was successful

*Note: This bit has the same value as the corresponding TXOKx bit in the CTSR register.*

Bit 2 = **RQCP** *Request Completed*
- Read/Clear
Set by hardware when the last request (transmit or abort) has been performed.

Cleared by software writing a “1” or by hardware on transmission request.

*Note: This bit has the same value as the corresponding RQCPx bit of the CTSR register.*

Clearing this bit clears all the status bits (TXOK, ALST and TERR) in the MCSR register and the RQCP and TXOK bits in the CTSR register.

Bit 1 = **ABRQ** *Abort Request for Mailbox*
- Read/Set
Set by software to abort the transmission request for the corresponding mailbox.

Cleared by hardware when the mailbox becomes empty.

Setting this bit has no effect when the mailbox is not pending for transmission.

Bit 0 = **TXRQ** *Transmit Mailbox Request*

- Read/Set
Set by software to request the transmission for the corresponding mailbox.

Cleared by hardware when the mailbox becomes empty.

*Note: This register is implemented only in transmit mailboxes. In receive mailboxes, the MFMI register is mapped at this location.*

### COMPARE RESULT REGISTER (CRR)

R243 - Read/Write
Register Page: 63
Reset Value: 0000 xxxx (0xh)

Two adjacent channels (identified as A and B) can be selected through CLR1 register programming (bits CC[3:0]); a level window for the converted analog input can be defined on these channels.

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| CBU | CAU | CBL | CAL | x | x | x | x |

Bits 7 = **CBU**: *Compare Register Ch. B Upper Threshold*
Set when converted data on channel B is greater than the threshold value set in UTBHR/UTBLR registers.

Bits 6 = **CAU**: *Compare Register Ch. A Upper Threshold*
Set when converted data on channel A is greater than the threshold value set in UTAHR/UTALR registers.

Bits 5 = **CBL**: *Compare Register Ch. B Lower Threshold*
Set when converted data on channel B is less than the threshold value set in LTBHR/LTBLR registers.

Bits 4 = **CAL:** *Compare Register Ch. A Lower Threshold*
Set when converted data on channel A is less than the threshold value set in LTAHR/LTALR registers.

Bits 3:0 = Don't care

### LOWER THRESHOLD REGISTERS (LTiHR/LTiLR)

The two pairs of Lower Threshold High/Low registers are used to store the user programmable lower threshold 10-bit values, to be compared with the current conversion results, thus setting the lower window limit.

### CHANNEL A LOWER THRESHOLD HIGH REGISTER (LTAHR)

R244 - Read
Register Page: 63
Reset Value: undefined

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| LTA.9 | LTA.8 | LTA.7 | LTA.6 | LTA.5 | LTA.4 | LTA.3 | LTA.2 |

Bits 7:0 = **LTA.[9:2]**: *Channel A [9:2] bit Lower Threshold*