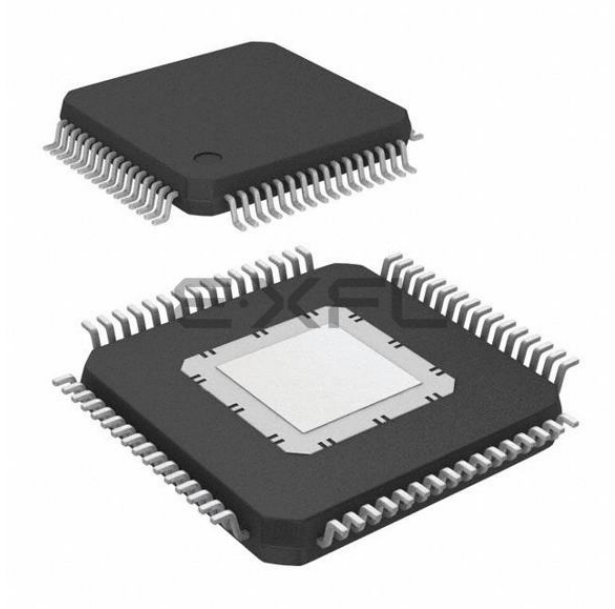

**Welcome to E-XFL.COM**

## 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 | S12Z |
| Core Size | 16-Bit |
| Speed | 50MHz |
| Connectivity | CANbus, SCI, SPI |
| Peripherals | DMA, POR, PWM, WDT |
| Number of I/O | 31 |
| Program Memory Size | 64KB (64K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | 512 x 8 |
| RAM Size | 4K x 8 |
| Voltage - Supply (Vcc/Vdd) | 3.5V ~ 40V |
| Data Converters | A/D 9x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 150°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 64-LQFP Exposed Pad |
| Supplier Device Package | 64-HLQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvmc64f2wkh |

| Port | Pin Name | ZVMC256 | ZVMC128/64 | ZVML128/64/32 | ZVML31 | ZVM32/16 | Pin Function & Priority[1] | I/O | Description | Routing Register Bit | Pin Function after Reset |
|---|---|---|---|---|---|---|---|---|---|---|---|
| AD | PAD7 | ✔ | ✔ | ✔ | ✔ | ✔ | AMPP1 | I | GDU AMP1 non-inverting input (+) | — | GPIO |
| | | ✔ | ✔ | ✔ | ✔ | ✔ | AN1_2 | I | ADC1 analog input | — | |
| | | ✔ | ✔ | ✔ | ✔ | ✔ | PTADL[7]/ KWADL[7] | I/O | General-purpose; with interrupt and wakeup | — | |
| | PAD6 | ✔ | ✔ | ✔ | ✔ | ✔ | AMPM1 | I | GDU AMP1 inverting input (-) | | |
| | | ✔ | ✔ | ✔ | ✔ | ✔ | ($\overline{SS0}$) | I/O | SPI0 slave select | SPI0SSRR | |
| | | ✔ | ✔ | ✔ | ✔ | ✔ | AN1_1 | I | ADC1 analog input | — | |
| | | ✔ | ✔ | ✔ | ✔ | ✔ | PTADL[6]/ KWADL[6] | I/O | General-purpose; with interrupt and wakeup | — | |
| | PAD5 | ✔ | ✔ | ✔ | ✔ | ✔ | AMP1 | O | GDU AMP1 output | — | |
| | | ✔ | ✔ | ✔ | ✔ | ✔ | AN1_0 | I | ADC1 analog input | — | |
| | | ✔ | ✔ | ✔ | ✔ | ✔ | PTADL[5]/ KWADL[5] | I/O | General-purpose; with interrupt and wakeup | — | |
| | PAD4 | ✔ | ✔ | ✔ | ✔ | ✔ | AN0_4 | I | ADC0 analog input | — | |
| | | ✔ | ✔ | ✔ | ✔ | ✔ | PTADL[4]/ KWADL[4] | I/O | General-purpose; with interrupt and wakeup | — | |
| | PAD3 | ✔ | ✔ | ✔ | ✔ | ✔ | AN0_3 | I | ADC0 analog input | — | |
| | | ✔ | ✔ | ✔ | ✔ | ✔ | PTADL[3]/ KWADL[3] | I/O | General-purpose; with interrupt and wakeup | — | |
| | PAD2 | ✔ | ✔ | ✔ | ✔ | ✔ | AMPP0 | I | GDU AMP0 non-inverting input (+) | — | |
| | | ✔ | ✔ | ✔ | ✔ | ✔ | AN0_2 | I | ADC0 analog input | — | |
| | | ✔ | ✔ | ✔ | ✔ | ✔ | PTADL[2]/ KWADL[2]/ | I/O | General-purpose; with interrupt and wakeup | — | |
| | PAD1 | ✔ | ✔ | ✔ | ✔ | ✔ | AMPM0 | I | GDU AMP0 inverting input (-) | — | |
| | | ✔ | ✔ | ✔ | ✔ | ✔ | AN0_1 | I | ADC0 analog input | — | |
| | | ✔ | ✔ | ✔ | ✔ | ✔ | PTADL[1]/ KWADL[1] | I/O | General-purpose; with interrupt and wakeup | — | |
| | PAD0 | ✔ | ✔ | ✔ | ✔ | ✔ | AMP0 | O | GDU AMP0 output | — | |
| | | ✔ | ✔ | ✔ | ✔ | ✔ | AN0_0 | I | ADC0 analog input | — | |
| | | ✔ | ✔ | ✔ | ✔ | ✔ | PTADL[0]/ KWADL[0] | I/O | General-purpose; with interrupt and wakeup | — | |

1. Signals in parentheses denote alternative module routing pins.

### 2.3.4.3 Port P Interrupt Flag Register (PIFP)

Address 0x02F7 PIFP Access: User read/write[1]

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | OCIF1 | 0 | 0 | 0 | 0 | PIFP2 | PIFP1 | PIFP0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 2-25. Port P Interrupt Flag Register**

1. Read: Anytime
   Write: Anytime, write 1 to clear

**Table 2-29. Port P Interrupt Flag Register Field Descriptions**

| Field | Description |
|---|---|
| 7<br>OCIF1 | **Over-Current Interrupt Flag register —**<br><br>This flag asserts if an over-current condition is detected on PP0 (Section 2.4.5, “Over-Current Interrupt”).<br><br>Writing a logic “1” to the corresponding bit field clears the flag.<br><br>1 PP0 Over-current event occurred<br>0 No PP0 over-current event occurred |
| 2-0<br>PIFP2-0 | **See Section 2.3.3.7, “Port Interrupt Flag Register”** |

| Field | Description |
|---|---|
| 7-4 (MMCECL) ACC[3:0] | **Access Type Field** — The ACC[3:0] bits capture the type of memory access, which caused the access violation. The access type is captured in form of a 4 bit value which is assigned as follows:<br>0: none (no error condition detected)<br>1: opcode fetch<br>2: vector fetch<br>3: data load<br>4: data store<br>5-15: reserved |
| 3-0 (MMCECL) ERR[3:0] | **Error Type Field** — The EC[3:0] bits capture the type of the access violation. The type is captured in form of a 4 bit value which is assigned as follows:<br>0: none (no error condition detected)<br>1: access to an illegal access<br>2: uncorrectable ECC error<br>3-15:reserved |

The MMCEC register captures debug information about access violations. It is set to a non-zero value if a S12ZCPU access violation or an uncorrectable ECC error has occurred. At the same time this register is set to a non-zero value, access information is captured in the MMCPCn and MMCCCRn registers. The MMCECn, the MMCPCn and the MMCCCRn registers are not updated if the MMCECn registers contain a non-zero value. The MMCECn registers are cleared by writing the value 0xFFFF.

### 3.3.2.3 Captured S12ZCPU Condition Code Register (MMCCCRH, MMCCCRL)

Address: 0x0082 (MMCCCRH)

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

Address: 0x0083 (MMCCCRL)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | CPUX | 0 | CPUI | 0 | 0 | 0 | 0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 3-6. Captured S12ZCPU Condition Code Register (MMCCCRH, MMCCCRL)**

Read: Anytime

Write: Never

**Table 6-9. DBGTCRH Field Descriptions**

| Field | Description |
|---|---|
| 6<br>TSOURCE | **Trace Control Bits** — The TSOURCE enables the tracing session.<br>0 No CPU tracing/profiling selected<br>1 CPU tracing/profiling selected |
| 5–4<br>TRANGE | **Trace Range Bits** — The TRANGE bits allow filtering of trace information from a selected address range when tracing from the CPU in Detail mode. These bits have no effect in other tracing modes. To use a comparator for range filtering, the corresponding COMPE bit must remain cleared. If the COMPE bit is set then the comparator is used to generate events and the TRANGE bits have no effect. See Table 6-10 for range boundary definition. |
| 3–2<br>TRCMOD | **Trace Mode Bits** — See Section 6.4.5.2 for detailed Trace Mode descriptions. In Normal Mode, change of flow information is stored. In Loop1 Mode, change of flow information is stored but redundant entries into trace memory are inhibited. In Detail Mode, address and data for all memory and register accesses is stored. See Table 6-11. |
| 1–0<br>TALIGN | **Trigger Align Bits** — These bits control whether the trigger is aligned to the beginning, end or the middle of a tracing or profiling session. See Table 6-12. |

**Table 6-10. TRANGE Trace Range Encoding**

| TRANGE | Tracing Range |
|---|---|
| 00 | Trace from all addresses (No filter) |
| 01 | Trace only in address range from $00000 to Comparator D |
| 10 | Trace only in address range from Comparator C to $FFFFFF |
| 11 | Trace only in range from Comparator C to Comparator D |

**Table 6-11. TRCMOD Trace Mode Bit Encoding**

| TRCMOD | Description |
|---|---|
| 00 | Normal |
| 01 | Loop1 |
| 10 | Detail |
| 11 | Pure PC |

**Table 6-12. TALIGN Trace Alignment Encoding**

| TALIGN | Description |
|---|---|
| 00 | Trigger ends data trace |
| 01 | Trigger starts data trace |
| 10 | 32 lines of data trace follow trigger |
| 11[1] | Reserved |

1. Tracing/Profiling disabled.

Module Base + 0x0007

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | RTIF | 0 | 0 | LOCKIF | LOCK | 0 | OSCIF | UPOSC |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Figure 8-12. S12CPMU_UHV_V10_V6 Flags Register (CPMUIFLG)**

Read: Anytime

Write: Refer to each bit for individual write conditions

**Table 8-5. CPMUIFLG Field Descriptions**

| Field | Description |
|---|---|
| 7<br>RTIF | **Real Time Interrupt Flag —** RTIF is set to 1 at the end of the RTI period. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (RTIE=1), RTIF causes an interrupt request.<br>0 RTI time-out has not yet occurred.<br>1 RTI time-out has occurred. |
| 4<br>LOCKIF | **PLL Lock Interrupt Flag** — LOCKIF is set to 1 when LOCK status bit changes. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (LOCKIE=1), LOCKIF causes an interrupt request.<br>0 No change in LOCK bit.<br>1 LOCK bit has changed. |
| 3<br>LOCK | **Lock Status Bit** — LOCK reflects the current state of PLL lock condition. Writes have no effect. While PLL is unlocked (LOCK=0) $f_{PLL}$ is $f_{VCO}$ / 4 to protect the system from high core clock frequencies during the PLL stabilization time $t_{lock}$.<br>0 VCOCLK is not within the desired tolerance of the target frequency.<br>$f_{PLL} = f_{VCO}/4$.<br>1 VCOCLK is within the desired tolerance of the target frequency.<br>$f_{PLL} = f_{VCO}/(POSTDIV+1)$. |
| 1<br>OSCIF | **Oscillator Interrupt Flag** — OSCIF is set to 1 when UPOSC status bit changes. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (OSCIE=1), OSCIF causes an interrupt request.<br>0 No change in UPOSC bit.<br>1 UPOSC bit has changed. |
| 0<br>UPOSC | **Oscillator Status Bit —** UPOSC reflects the status of the oscillator. Writes have no effect. Entering Full Stop Mode UPOSC is cleared.<br>0 The oscillator is off or oscillation is not qualified by the PLL.<br>1 The oscillator is qualified by the PLL. |

### 8.3.2.8 S12CPMU_UHV_V10_V6 Interrupt Enable Register (CPMUINT)

This register enables S12CPMU_UHV_V10_V6 interrupt requests.

Module Base + 0x0008

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | RTIE | 0 | 0 | LOCKIE | 0 | 0 | OSCIE | 0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Figure 8-13. S12CPMU_UHV_V10_V6 Interrupt Enable Register (CPMUINT)**

Read: Anytime

Write: Anytime

**Table 8-6. CPMUINT Field Descriptions**

| Field | Description |
|---|---|
| 7<br>RTIE | Real Time Interrupt Enable Bit<br>0 Interrupt requests from RTI are disabled.<br>1 Interrupt will be requested whenever RTIF is set. |
| 4<br>LOCKIE | PLL Lock Interrupt Enable Bit<br>0 PLL LOCK interrupt requests are disabled.<br>1 Interrupt will be requested whenever LOCKIF is set. |
| 1<br>OSCIE | Oscillator Corrupt Interrupt Enable Bit<br>0 Oscillator Corrupt interrupt requests are disabled.<br>1 Interrupt will be requested whenever OSCIF is set. |

**Table 8-13. RTI Frequency Divide Rates for RTDEC=1**

| RTR[3:0] | RTR[6:4] = | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **000 ($1x10^3$)** | **001 ($2x10^3$)** | **010 ($5x10^3$)** | **011 ($10x10^3$)** | **100 ($20x10^3$)** | **101 ($50x10^3$)** | **110 ($100x10^3$)** | **111 ($200x10^3$)** |
| **0000 (÷1)** | $1x10^3$ | $2x10^3$ | $5x10^3$ | $10x10^3$ | $20x10^3$ | $50x10^3$ | $100x10^3$ | $200x10^3$ |
| **0001 (÷2)** | $2x10^3$ | $4x10^3$ | $10x10^3$ | $20x10^3$ | $40x10^3$ | $100x10^3$ | $200x10^3$ | $400x10^3$ |
| **0010 (÷3)** | $3x10^3$ | $6x10^3$ | $15x10^3$ | $30x10^3$ | $60x10^3$ | $150x10^3$ | $300x10^3$ | $600x10^3$ |
| **0011 (÷4)** | $4x10^3$ | $8x10^3$ | $20x10^3$ | $40x10^3$ | $80x10^3$ | $200x10^3$ | $400x10^3$ | $800x10^3$ |
| **0100 (÷5)** | $5x10^3$ | $10x10^3$ | $25x10^3$ | $50x10^3$ | $100x10^3$ | $250x10^3$ | $500x10^3$ | $1x10^6$ |
| **0101 (÷6)** | $6x10^3$ | $12x10^3$ | $30x10^3$ | $60x10^3$ | $120x10^3$ | $300x10^3$ | $600x10^3$ | $1.2x10^6$ |
| **0110 (÷7)** | $7x10^3$ | $14x10^3$ | $35x10^3$ | $70x10^3$ | $140x10^3$ | $350x10^3$ | $700x10^3$ | $1.4x10^6$ |
| **0111 (÷8)** | $8x10^3$ | $16x10^3$ | $40x10^3$ | $80x10^3$ | $160x10^3$ | $400x10^3$ | $800x10^3$ | $1.6x10^6$ |
| **1000 (÷9)** | $9x10^3$ | $18x10^3$ | $45x10^3$ | $90x10^3$ | $180x10^3$ | $450x10^3$ | $900x10^3$ | $1.8x10^6$ |
| **1001 (÷10)** | $10\ x10^3$ | $20x10^3$ | $50x10^3$ | $100x10^3$ | $200x10^3$ | $500x10^3$ | $1x10^6$ | $2x10^6$ |
| **1010 (÷11)** | $11\ x10^3$ | $22x10^3$ | $55x10^3$ | $110x10^3$ | $220x10^3$ | $550x10^3$ | $1.1x10^6$ | $2.2x10^6$ |
| **1011 (÷12)** | $12x10^3$ | $24x10^3$ | $60x10^3$ | $120x10^3$ | $240x10^3$ | $600x10^3$ | $1.2x10^6$ | $2.4x10^6$ |
| **1100 (÷13)** | $13x10^3$ | $26x10^3$ | $65x10^3$ | $130x10^3$ | $260x10^3$ | $650x10^3$ | $1.3x10^6$ | $2.6x10^6$ |
| **1101 (÷14)** | $14x10^3$ | $28x10^3$ | $70x10^3$ | $140x10^3$ | $280x10^3$ | $700x10^3$ | $1.4x10^6$ | $2.8x10^6$ |
| **1110 (÷15)** | $15x10^3$ | $30x10^3$ | $75x10^3$ | $150x10^3$ | $300x10^3$ | $750x10^3$ | $1.5x10^6$ | $3x10^6$ |
| **1111 (÷16)** | $16x10^3$ | $32x10^3$ | $80x10^3$ | $160x10^3$ | $320x10^3$ | $800x10^3$ | $1.6x10^6$ | $3.2x10^6$ |

### 8.3.2.19 Autonomous Clock Trimming Register (CPMUACLKTR)

The CPMUACLKTR register configures the trimming of the Autonomous Clock (ACLK - trimmable internal RC-Oscillator) which can be selected as clock source for some CPMU features.

Module Base + 0x0013

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | ACLKTR5 | ACLKTR4 | ACLKTR3 | ACLKTR2 | ACLKTR1 | ACLKTR0 | 0 | 0 |
| W | | | | | | | | |
| Reset | F | F | F | F | F | F | 0 | 0 |

After de-assert of System Reset a value is automatically loaded from the Flash memory.

**Figure 8-26. Autonomous Clock Trimming Register (CPMUACLKTR)**

Read: Anytime

Write: Anytime

**Table 8-20. CPMUACLKTR Field Descriptions**

| Field | Description |
|---|---|
| 7–2<br>ACLKTR[5:0] | **Autonomous Clock Period Trimming Bits** — See Table 8-21 for trimming effects. The ACLKTR[5:0] value represents a signed number influencing the ACLK period time. |

**Table 8-21. Trimming Effect of ACLKTR[5:0]**

| ACLKTR[5:0] | Decimal | ACLK frequency |
|---|---|---|
| 100000 | -32 | lowest |
| 100001 | -31 | increasing |
| .... | | |
| 111111 | -1 | |
| 000000 | 0 | mid |
| 000001 | +1 | increasing |
| .... | | |
| 011110 | +30 | |
| 011111 | +31 | highest |

### 9.5.2.5 ADC Format Register (ADCFMT)

Module Base + 0x0004

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | DJM | 0 | 0 | 0 | 0 | SRES[2:0] | | |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Figure 9-8. ADC Format Register (ADCFMT)**

Read: Anytime

Write: Bits DJM and SRES[2:0] are writable if bit ADC_EN clear or bit SMOD_ACC set

**Table 9-8. ADCFMT Field Descriptions**

| Field | Description |
|---|---|
| 7<br>DJM | **Result Register Data Justification** — Conversion result data format is always unsigned. This bit controls justification of conversion result data in the conversion result list.<br>0 Left justified data in the conversion result list.<br>1 Right justified data in the conversion result list. |
| 2-0<br>SRES[2:0] | **ADC Resolution Select** — These bits select the resolution of conversion results. See Table 9-9 for coding. |

**Table 9-9. Selectable Conversion Resolution**

| SRES[2] | SRES[1] | SRES[0] | ADC Resolution |
|---|---|---|---|
| 0 | 0 | 0 | 8-bit data |
| 0 | 0 | 1 | Reserved[1.] |
| 0 | 1 | 0 | 10-bit data |
| 0 | 1 | 1 | Reserved[1.] |
| 1 | 0 | 0 | 12-bit data |
| 1 | x | x | Reserved[(1)] |

1. Reserved settings cause a severe error at ADC conversion start whereby the CMD_EIF flag is set and ADC ceases operation

### 11.3.2.12 Timer Input Capture/Output Compare Registers High and Low 0–3(TCxH and TCxL)

Module Base + 0x0010 = TC0H 0x0018=RESERVD
0x0012 = TC1H 0x001A=RESERVD
0x0014=TC2H 0x001C=RESERVD
0x0016=TC3H 0x001E=RESERVD

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 11-18. Timer Input Capture/Output Compare Register x High (TCxH)**

Module Base + 0x0011 = TC0L 0x0019 =RESERVD
0x0013 = TC1L 0x001B=RESERVD
0x0015 =TC2L 0x001D=RESERVD
0x0017=TC3L 0x001F=RESERVD

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>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 11-19. Timer Input Capture/Output Compare Register x Low (TCxL)**

[1] This register is available only when the corresponding channel exists and is reserved if that channel does not exist. Writes to a reserved register have no functional effect. Reads from a reserved register return zeroes.

Depending on the TIOS bit for the corresponding channel, these registers are used to latch the value of the free-running counter when a defined transition is sensed by the corresponding input capture edge detector or to trigger an output action for output compare.

Read: Anytime

Write: Anytime for output compare function.Writes to these registers have no meaning or effect during input capture. All timer input capture/output compare registers are reset to 0x0000.

**NOTE**

Read/Write access in byte mode for high byte should take place before low byte otherwise it will give a different result.

Module Base + 0x0005

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | TCNT7 | TCNT6 | TCNT5 | TCNT4 | TCNT3 | TCNT2 | TCNT1 | TCNT0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 12-7. Timer Count Register Low (TCNTL)**

The 16-bit main timer is an up counter.

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

Read: Anytime

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

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

### 12.3.2.4 Timer System Control Register 1 (TSCR1)

Module Base + 0x0006

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

= Unimplemented or Reserved

**Figure 12-8. Timer System Control Register 1 (TSCR1)**

Read: Anytime

Write: Anytime

**Table 12-4. TSCR1 Field Descriptions**

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

# Chapter 15
# Pulse Width Modulator with Fault Protection (PMF15B6CV4)

**Table 15-1. Revision History**

| Rev. No. (Item No.) | Date (Submitted By) | Sections Affected | Substantial Change(s) |
|---|---|---|---|
| V03.22 | 02 Sep 2013 | 15.3.2.4/15-574<br>15.3.2.11/15-579 | • Corrected PINVx bit descriptions<br>• Improved read description of PMFOUTB |
| V03.23 | 10 Oct 2013 | 15.2.8/15-565<br>15.3.2.18/15-585<br>15.3.2.22/15-589<br>15.8.1.1/15-629 | • Corrected pmf_reload_is_async signal description<br>• Enhanced note at PMFCINV register<br>• Corrected write value limitations for PMFMODx registers<br>• Corrected register write protection bit names<br>• Orthographical corrections after review |
| V03.24 | 08 Nov 2013 | 15.3.2.8/15-577<br>Table 15-15<br>15.4.7/15-613 | • Updated PMFFIF bit description<br>• Updated note to QSMP table<br>• Updated Asymmetric PWM output description<br>• Replaced ‘fault clearing’ with ‘fault recovery’ to avoid ambiguity with flags<br>• Various minor corrections.<br>• |
| V03.25 | 03 Dec 2013 | 15.3.2.18/15-585 | • Updated note at PMFCINV register |
| V04.00 | 03 Dec 2013 | 15.3.2.3/15-573<br>15.3.2.11/15-579<br>15.3.2.18/15-585 | • Added write protection to REV1-0 bits (WP)<br>• Added PWM read through PMFOUTB (generator output read option)<br>• Updated note at CINV*n* bits |
| V04.1 | 05 Nov 2015 | Figure 15-51./15-606<br>Figure 15-52./15-607Figure 15-53./15-607 | • correct figure Figure 15-51./15-606, Figure 15-52./15-607,Figure 15-53./15-607<br>• update DMPx register description |

Glossary

**Table 15-2. Glossary of Terms**

| Term | Definition |
|---|---|
| Set | Discrete signal is in active logic state. |
| Clear | A discrete signal is in inactive logic state. |
| Pin | External physical connection. |
| Signal | Electronic construct whose state or change in state conveys information. |

**Figure 18-31. Examples of Intrinsic Dead Time**

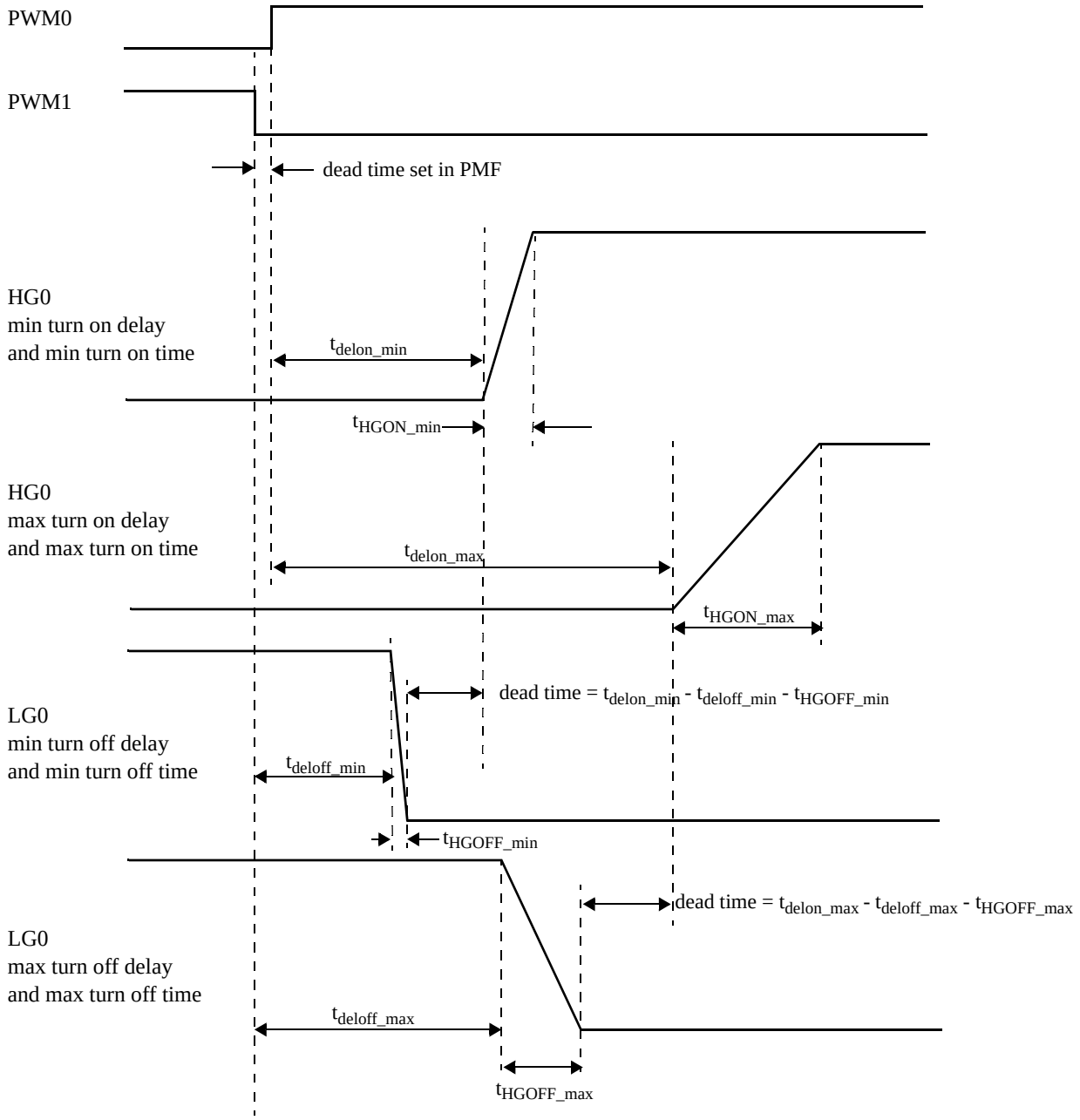

**Table 20-36. Erase Verify Block Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 001 at command launch |
| | | Set if an invalid global address [23:0] is supplied see Table 20-3) |
| | FPVIOL | None |
| | MGSTAT1 | Set if any errors have been encountered during the read or if blank check failed. |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the read or if blank check failed. |

### 20.4.7.3 Erase Verify P-Flash Section Command

The Erase Verify P-Flash Section command will verify that a section of code in the P-Flash memory is erased. The Erase Verify P-Flash Section command defines the starting point of the code to be verified and the number of phrases.

**Table 20-37. Erase Verify P-Flash Section Command FCCOB Requirements**

| Register | FCCOB Parameters | |
|---|---|---|
| FCCOB0 | 0x03 | Global address [23:16] of a P-Flash block |
| FCCOB1 | Global address [15:0] of the first phrase to be verified | |
| FCCOB2 | Number of phrases to be verified | |

Upon clearing CCIF to launch the Erase Verify P-Flash Section command, the Memory Controller will verify the selected section of Flash memory is erased. The CCIF flag will set after the Erase Verify P-Flash Section operation has completed. If the section is not erased, it means blank check failed, both MGSTAT bits will be set.

**Table 20-38. Erase Verify P-Flash Section Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 010 at command launch |
| | | Set if command not available in current mode (see Table 20-29) |
| | | Set if an invalid global address [23:0] is supplied see Table 20-3) |
| | | Set if a misaligned phrase address is supplied (global address [2:0] != 000) |
| | | Set if the requested section crosses a the P-Flash address boundary |
| | FPVIOL | None |
| | MGSTAT1 | Set if any errors have been encountered during the read or if blank check failed. |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the read or if blank check failed. |

**Table 21-6. CPSR Register Field Descriptions**

| Field | Description |
|---|---|
| 4<br>CPCLVL | CANL Voltage Failure Low Status Bit<br>This bit reflects the CANL voltage failure low monitor status.<br><br>0 Condition $V_{CANL} > V_{L0}$<br>1 Condition $V_{CANL} \leq V_{L0}$ |
| 3<br>CPDT | CPTXD-Dominant Timeout Status Bit<br>This bit is set to 1, if CPTXD is dominant for longer than $t_{CPTXDDT}$. It signals a timeout event and remains set until CPTXD returns to recessive level for longer than 1 µs.<br><br>0 No CPTXD-timeout occurred or CPTXD has ceased to be dominant after timeout<br>1 CPTXD-dominant timeout occurred and CPTXD is still dominant |

# 21.5 Functional Description

## 21.5.1 General

The CAN Physical Layer provides an interface for the SoC-integrated MSCAN controller.

## 21.5.2 Modes

Figure 21-10 shows the possible mode transitions depending on control bit CPE, device reduced performance mode (“RPM”; refer to “Low Power Modes” section in device overview) and bus error conditions.

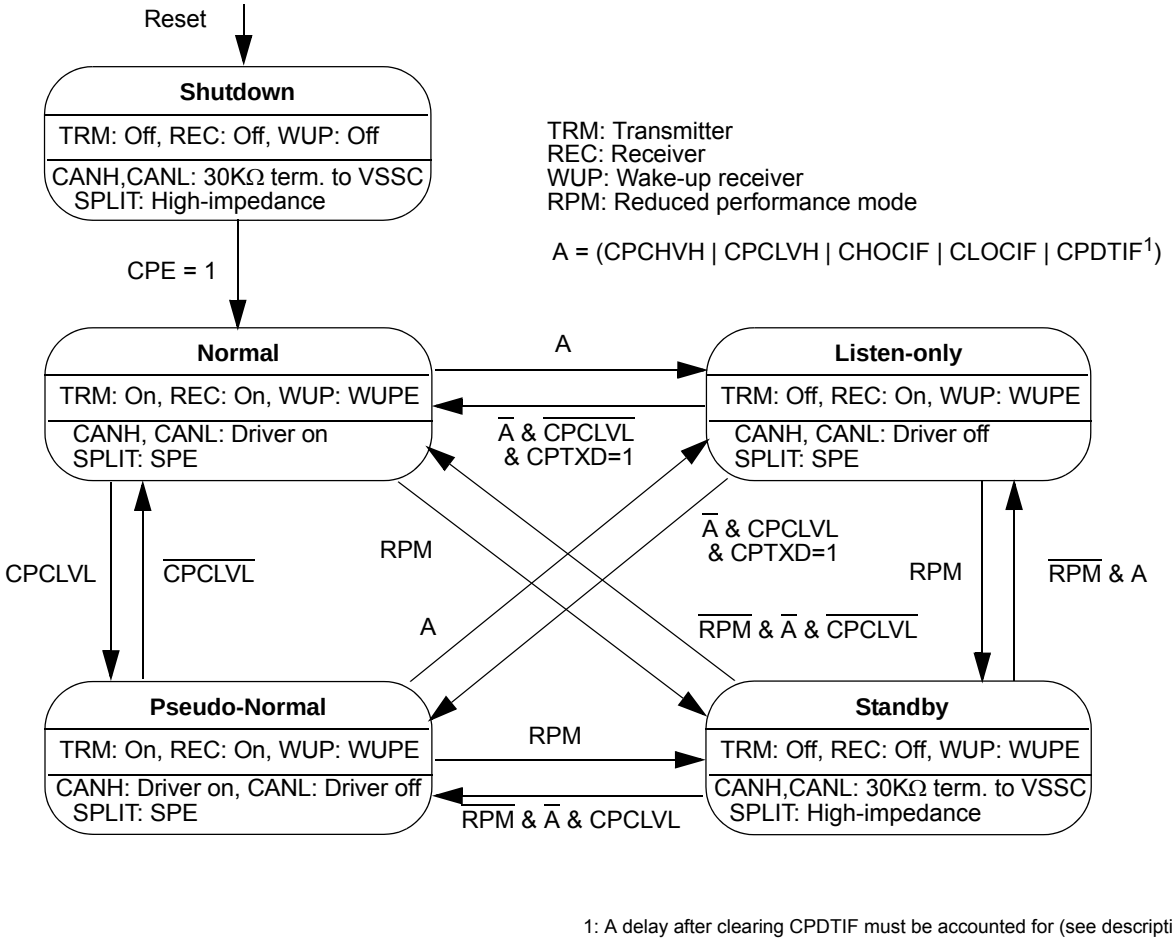


**Figure 21-10. CAN Physical Layer Mode Transitions**

### 21.5.2.1 Shutdown Mode

Shutdown mode is a low power mode and entered out of reset. The transceiver, wake-up, bus error diagnostic, dominant timeout and interrupt functionality are disabled. CANH and CANL lines are pulled

2. Configure CAN Physical Layer slew rate
3. Enable CAN Physical Layer interrupts
4. Optionally enable SPLIT pin
5. Configure wake-up filter or disable wake-up receiver in case of other wake-up sources
6. Enable CAN Physical Layer to enter normal mode
7. Start CAN communication

## 21.6.2 Wake-up Mechanism

In stop mode the CAN Physical Layer passes CAN bus states to CPRXD if the wake-up function is enabled (CPCR[WUPE1:WUPE0]≠0). In order to wake up the device from stop mode, the wake-up interrupt of the connected MSCAN module is used.

If CPCR[WUPE1:WUPE0]=b10 the CAN Physical Layer is transparent in stop mode and the MSCAN can be used with or without its integrated low-pass filter for wake-up. Refer to the MSCAN chapter for details on configuring and enabling the wake-up function.

For increased robustness against false wake-up, a CAN Physical Layer pulse filter can optionally be enabled to mask the first (CPCR[WUPE1:WUPE0]=b01) or first two (CPCR[WUPE1:WUPE0]=b11) wake-up events after entering stop mode. The appropriate number of masked pulses depends on the individual CAN bus network topology.

Note that the MSCAN can generate a wake-up interrupt immediately after it acknowledges sleep mode (CANCTL1[SPLAK]=1) whereas the CAN Physical Layer pulse filter takes effect only after entering stop mode. To avoid a false wake-up in between these two events, the MSCAN low-pass filter should also be activated (CANCTL1[WUPM]=1). After sleep mode acknowledge the CPU STOP instruction should be executed before the expiration of $t_{WUP}$(min) to enable the CAN Physical Layer pulse filter in time.

## 21.6.3 Bus Error Handling

Upon CAN bus error voltage high failures and over-current events listen-only is entered immediately and the transmitter is turned off. This mode is maintained as long as voltage failure conditions persist or, in case of over-current events, application software re-enables the transmit driver by clearing the related flags.

All high and low voltage levels for both CAN bus lines are continuously reflected in their related voltage failure status bits. A change in a status bit sets the corresponding flag and generates an interrupt if enabled. As long as any of the voltage failure high status bits is set, the transmit driver remains off. It will be turned on again automatically as soon as all voltage failure conditions have disappeared. In case of a voltage failure low condition on CANL only the CANL driver is disabled. A voltage failure low condition on CANH has no effect on the transmitter.

Voltage failure errors have informational purpose. If the application detects frequent CAN protocol errors it is advisable to take the appropriate action. No software action is need to re-enable the transmit driver.

An over-current event on either CAN bus line sets the related flag and turns off the transmit driver. This error can only be detected while driving the bus dominant. In contrast to the voltage failure the over-current

# Chapter 22
# Pulse-Width Modulator (S12PWM8B8CV2)

**Table 22-1. Revision History**

| Revision Number | Revision Date | Sections Affected | Description of Changes |
|---|---|---|---|
| v02.00 | Feb. 20, 2009 | All | Initial revision of scalable PWM. Started from pwm_8b8c (v01.08). |

## 22.1 Introduction

The Version 2 of S12 PWM module is a channel scalable and optimized implementation of S12 PWM8B8C Version 1. The channel is scalable in pairs from PWM0 to PWM7 and the available channel number is 2, 4, 6 and 8. The shutdown feature has been removed and the flexibility to select one of four clock sources per channel has improved. If the corresponding channels exist and shutdown feature is not used, the Version 2 is fully software compatible to Version 1.

### 22.1.1 Features

The scalable PWM block includes these distinctive features:

- Up to eight independent PWM channels, scalable in pairs (PWM0 to PWM7)
- Available channel number could be 2, 4, 6, 8 (refer to device specification for exact number)
- Programmable period and duty cycle for each channel
- Dedicated counter for each PWM channel
- Programmable PWM enable/disable for each channel
- Software selection of PWM duty pulse polarity for each channel
- Period and duty cycle are double buffered. Change takes effect when the end of the effective period is reached (PWM counter reaches zero) or when the channel is disabled.
- Programmable center or left aligned outputs on individual channels
- Up to eight 8-bit channel or four 16-bit channel PWM resolution
- Four clock sources (A, B, SA, and SB) provide for a wide range of frequencies
- Programmable clock select logic

### 22.1.2 Modes of Operation

There is a software programmable option for low power consumption in wait mode that disables the input clock to the prescaler.

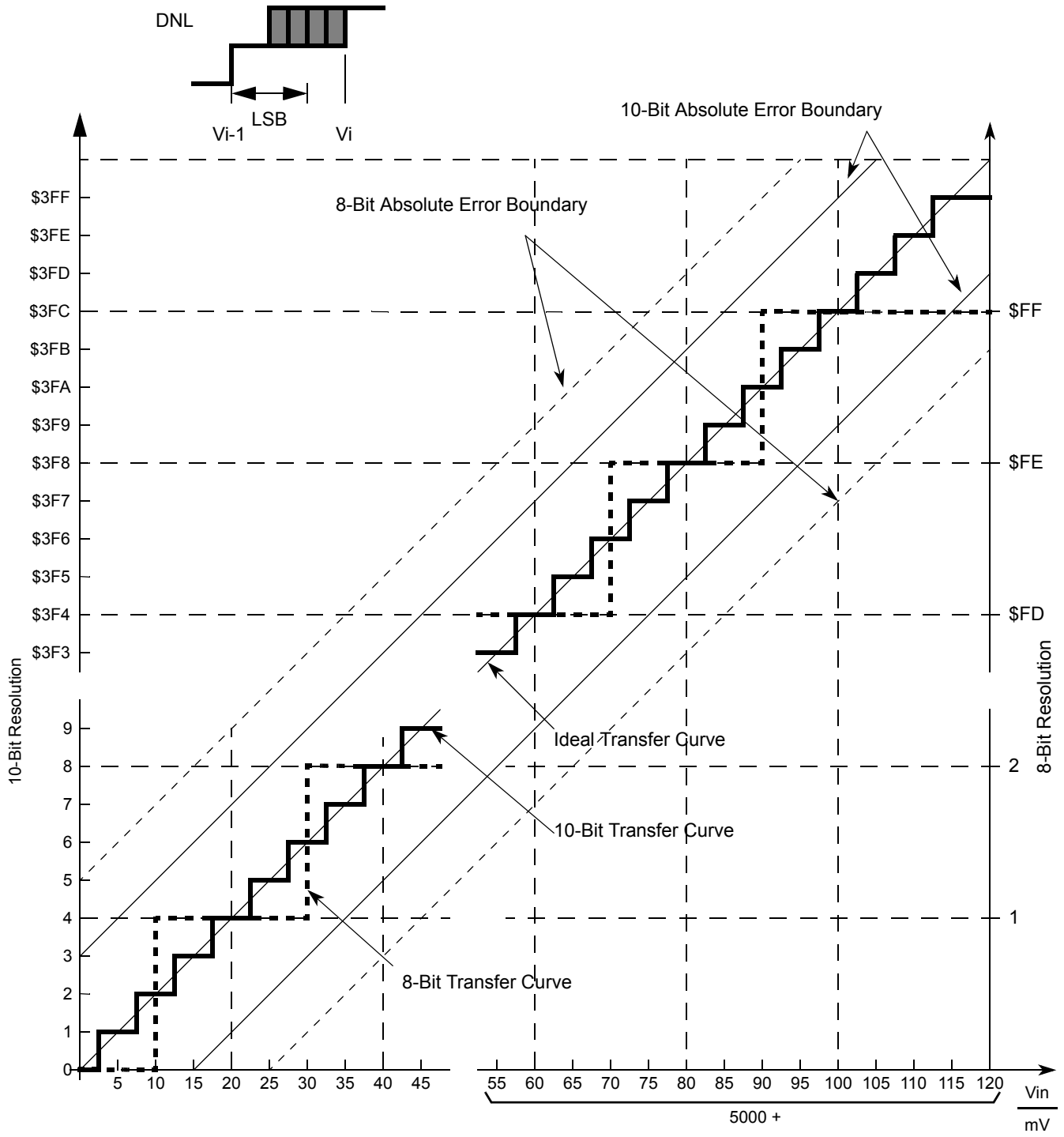


**Figure C-2. ADC Accuracy Definitions**

# Appendix D
# LIN/HV PHY Electrical Specifications

## D.1 Static Electrical Characteristics

**Table D-1. Static electrical characteristics of the LIN/HV PHY (Junction Temperature From -40°C To +175°C)**

| Characteristics noted under conditions 5.5V <= $V_{LINSUP}$ <= 18V unless otherwise noted(1) (2) (3). 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** |
| 1 | | $V_{LINSUP}$ range for LIN compliant electrical characteristics | $V_{LINSUP\_LIN}$ | 5.5[1 2] | 12 | 18 | V |
| 2 | | Current limitation into the LIN pin in dominant state(4) $V_{LIN}$ = $V_{LINSUP\_LIN\_MAX}$ | $I_{LIN\_LIM}$ | 40 | — | 200 | mA |
| 3 | | Input leakage current in dominant state, driver off, internal pull-up on ($V_{LIN}$ = 0V, $V_{LINSUP}$ = 12V) | $I_{LIN\_PAS\_dom}$ | -1 | — | — | mA |
| 4 | | Input leakage current in recessive state, driver off (5V<$V_{LINSUP}$<18V, 5V<$V_{LIN}$<18V, $V_{LIN}$ => $V_{LINSUP}$ ) | $I_{LIN\_PAS\_rec}$ | — | — | 20 | µA |
| 5 | | Input leakage current when ground disconnected ($GND_{Device}$ = $V_{LINSUP}$, 0V<$V_{LIN}$<18V, $V_{LINSUP}$ = 12V) | $I_{LIN\_NO\_GND}$ | -1 | — | 1 | mA |
| 6 | | Input leakage current when battery disconnected ($V_{LINSUP}$ = GND, 0<$V_{LIN}$<18V) | $I_{LIN\_NO\_BAT}$ | — | — | 30 | µA |
| 7 | | Receiver dominant state | $V_{LINdom}$ | — | — | 0.4 | $V_{LINSUP}$ |
| 8 | | Receiver recessive state | $V_{LINrec}$ | 0.6 | — | — | $V_{LINSUP}$ |
| 9 | | $V_{LIN\_CNT}$ =($V_{th\_dom}$+ $V_{th\_rec}$)/2 | $V_{LIN\_CNT}$ | 0.475 | 0.5 | 0.525 | $V_{LINSUP}$ |
| 10 | | $V_{HYS}$ = $V_{th\_rec}$ -$V_{th\_dom}$ | $V_{HYS}$ | — | — | 0.175 | $V_{LINSUP}$ |
| 11 | | Maximum capacitance allowed on slave node | $C_{slave}$ | — | 220 | 250 | pF |
| 12a | | Capacitance of LIN pin -40°C < $T_J$ < 150°C, Recessive state | $C_{int}$ | — | 20 | — | pF |
| 12b | | Capacitance of LIN pin -40°C < $T_J$ < 150°C, Recessive state | $C_{int}$ | — | — | 45 | pF |
| 12c | | Capacitance of LIN pin 150°C < $T_J$ < 175°C, Recessive state | $C_{int}$ | — | — | 39 | pF |
| 13 | | Internal pull-up (slave) | $R_{slave}$ | 27 | 34 | 40 | kΩ |

1. For 3.5V<= $V_{LINSUP}$ <5V, the LIN/HV PHY is still working but with degraded parametrics.

2. For 5V<= $V_{LINSUP}$ <5.5V, characterization showed that all parameters generally stay within the indicated specification, except the duty cycles D2 and D4 which may increase and potentially go beyond their maximum limits for highly loaded buses.

3. The $V_{LINSUP}$ voltage is provided by the VLINSUP supply. This supply mapping is described in device level documentation.