Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	S12
Core Size	16-Bit
Speed	20MHz
Connectivity	LINbus, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	3
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 15x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP Exposed Pad
Supplier Device Package	48-HLQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/sm912f634dv2aer2

Table 13. Static Electrical Characteristics - Voltage Regulator 5V (VDDX)

Ratings	Symbol	Min	Typ	Max	Unit
Normal Mode Output Voltage 1.0 mA < I_{VDDX} + $I_{VDDXINTERNAL}$ < 80 mA; 5.5 V < V_{SUP} < 27 V (20)	V_{DDXRUN}	4.75	5.00	5.25	V
Normal Mode Output Current Limitation (I_{VDDX})	$I_{VDDXLIMRUN}$	80	130	200	mA
Stop Mode Output Voltage (I_{VDDX} + $I_{VDDXINTERNAL}$ < 500 µA for $T_J \geq$ 25 °C; I_{VDDX} + $I_{VDDXINTERNAL}$ < 400 µA for T_J < 25 °C) (20)	$V_{DDXSTOP}$	-	5.0	5.5	V
Stop Mode Output Current Limitation (I_{VDDX})	$I_{VDDXLIMSTOP}$	-	-	20	mA
Line Regulation Normal Mode, I_{VDDX} = 80 mA Stop Mode, I_{VDDX} = 500 µA	LR_{XRUN} LR_{XSTOP}	- -	20 -	25 200	mV
Load Regulation Normal Mode, 1.0 mA < I_{VDDX} < 80 mA Normal Mode, V_{SUP} = 3.6 V, 1.0 mA < I_{VDDX} < 40 mA Stop Mode, 100 µA < I_{VDDX} < 500 µA	LD_{XRUN} LD_{XCRK} LD_{XSTOP}	- - -	15 - -	80 200 250	mV
External Capacitor	C_{VDDX}	1.0	-	10	µF
External Capacitor ESR	C_{VDDX_R}	-	-	10	Ω

Note:
20. $I_{VDDXINTERNAL}$ includes internal consumption from both analog and MCU die.

Table 14. Static Electrical Characteristics - Voltage Regulator 2.5 V (VDD)

Ratings	Symbol	Min	Typ	Max	Unit
Normal Mode Output Voltage 1.0 mA < I_{VDD} + $I_{VDDINTERNAL}$ ≤ 45 mA; 5.5 V < V_{SUP} < 27 V (21)	V_{DDRUN}	2,425	2.5	2,575	V
Normal Mode Output Current Limitation (I_{VDD}) T_J < 25 °C $T_J \geq$ 25 °C	$I_{VDDLIMRUN}$	- -	80 80	120 143	mA
Stop Mode Output Voltage´ (I_{VDD} + $I_{VDDINTERNAL}$ < 500 µA for $T_J \geq$ 25 °C; I_{VDD} + $I_{VDDINTERNAL}$ < 400 µA for T_J < 25 °C) (21)	V_{DDSTOP}	2.25	2.5	2.75	V
Stop Mode Output Current Limitation (I_{VDD})	$I_{VDDLIMSTOP}$	-	-	10	mA
Line Regulation Normal Mode, I_{VDD} = 45 mA Stop Mode, I_{VDD} = 1.0 mA	LR_{RUN} LR_{STOP}	- -	10 -	12.5 200	mV
Load Regulation Normal Mode, 1.0 mA < I_{VDD} < 45 mA Normal Mode, V_{SUP} = 3.6 V, 1.0 mA < I_{VDD} < 30 mA Stop Mode, 100 µA < I_{VDD} < 500 µA	LD_{RUN} LD_{CRK} LD_{STOP}	- - -	7.5 - -	40 40 200	mV
External Capacitor	C_{VDD}	1.0	-	10	µF
External Capacitor ESR	C_{VDD_R}	-	-	10	Ω

Note:
21. $I_{VDDINTERNAL}$ includes internal consumption from both analog and MCU die.

Table 20. Static Electrical Characteristics - General Purpose I/O - PTB[0…2] (continued)

Ratings	Symbol	Min	Typ	Max	Unit
Input low voltage (VS1 = 3.7 V)	$V_{IL3.7}$	V_{SS}-0.3	-	1.4	V
Input hysteresis (VS1 = 3.7 V)	$V_{HYS3.7}$	100	200	300	mV
Input leakage current (pins in high-impedance input mode) (V_{IN} = V_{DDX} or V_{SSX})	I_{IN}	-1.0	-	1.0	µA
Output high voltage (pins in output mode) Full drive I_{OH} = –10 mA	V_{OH}	V_{DDX}-0.8	-	-	V
Output low voltage (pins in output mode) Full drive I_{OL} = +10 mA	V_{OL}	-	-	0.8	V
Internal pull-up resistance (V_{IH} min > Input voltage > V_{IL} max)	R_{PUL}	26.25	37.5	48.75	kΩ
Input capacitance	C_{IN}	-	6.0	-	pF
Clamp Voltage when selected as analog input	V_{CL_AIN}	VDD	-	-	V
Analog Input impedance = 10 kΩ max, Capacitance = 12 pF	R_{AIN}	-	-	10	kΩ
Analog Input Capacitance = 12 pF	C_{AIN}	-	12	-	pF
Maximum current all PTB combined (VDDX capability)	I_{BMAX}	-15	-	15	mA
Output Drive strength at 10 MHz	C_{OUT}	-	-	100	pF

Table 21. Static Electrical Characteristics - Analog Digital Converter - ADC(24)

Ratings	Symbol	Min	Typ	Max	Unit
ADC2p5 Reference Voltage 5.5 V < V_{SUP} < 27 V	$V_{ADC2p5RUN}$	2,45	2.5	2,55	V
ADC2p5 Reference Stop Mode Output Voltage	$V_{ADC2p5STOP}$	-	-	100	mV
Line Regulation, Normal Mode	LR_{RUNA}	-	10	12.5	mV
External Capacitor	C_{ADC2p5}	0.1	-	1.0	µF
External Capacitor ESR	C_{VDD_R}	-	-	10	W
Scale Factor Error	E_{SCALE}	-1	-	1	LSB
Differential Linearity Error	E_{DNL}	-1.5	-	1.5	LSB
Integral Linearity Error	E_{INL}	-1.5	-	1.5	LSB
Zero Offset Error	E_{OFF}	-2.0	-	2.0	LSB
Quantization Error	E_Q	-0.5	-	0.5	LSB
Total Error with offset compensation	TE	-5.0	-	5.0	LSB
Bandgap measurement Channel (CH14) Valid Result Range (including ±7.0% bg1p25 sleep accuracy + high-impedance measurement error of ±5.0% at f_{ADC})(25)	AD_{CH14}	1.1	1.25	1.4	V

Note:
24. No external load allowed on the ADC2p5 pin.
25. Reduced ADC frequency lowers measurement error.

Table 65. Analog die Registers[64] - 0x0200–0x02FF D2D Blocking Access (D2DI) 2 of 3/ 0x0300–0x03FF D2D Non Blocking Access (D2DI) 3 of 3 (continued)

Offset	Name		7	6	5	4	3	2	1	0
0x9A	ADR10 (hi)	R	adr10 9	adr10 8	adr10 7	adr10 6	adr10 5	adr10 4	adr10 3	adr10 2
	ADC Data Result Reg 10	W								
0x9B	ADR10 (lo)	R	adr10 1	adr10 0	0	0	0	0	0	0
	ADC Data Result Reg 10	W								
0x9C	ADR11 (hi)	R	adr11 9	adr11 8	adr11 7	adr11 6	adr11 5	adr11 4	adr11 3	adr11 2
	ADC Data Result Reg 11	W								
0x9D	ADR11 (lo)	R	adr11 1	adr11 0	0	0	0	0	0	0
	ADC Data Result Reg 11	W								
0x9E	ADR12 (hi)	R	adr12 9	adr12 8	adr12 7	adr12 6	adr12 5	adr12 4	adr12 3	adr12 2
	ADC Data Result Reg 12	W								
0x9F	ADR12 (lo)	R	adr12 1	adr12 0	0	0	0	0	0	0
	ADC Data Result Reg 12	W								
0xA2	ADR14 (hi)	R	adr14 9	adr14 8	adr14 7	adr14 6	adr14 5	adr14 4	adr14 3	adr14 2
	ADC Data Result Reg 14	W								
0xA3	ADR14 (lo)	R	adr14 1	adr14 0	0	0	0	0	0	0
	ADC Data Result Reg 14	W								
0xA4	ADR15 (hi)	R	adr15 9	adr15 8	adr15 7	adr15 6	adr15 5	adr15 4	adr15 3	adr15 2
	ADC Data Result Reg 15	W								
0xA5	ADR15 (lo)	R	adr15 1	adr15 0	0	0	0	0	0	0
	ADC Data Result Reg 15	W								
0xC0	TIOS	R	0	0	0	0	IOS3	IOS2	IOS1	IOS0
	TIM InCap/OutComp Select	W								
0xC1	CFORC	R	0	0	0	0	0	0	0	0
	Timer Compare Force Reg	W					FOC3	FOC2	FOC1	FOC0
0xC2	OC3M	R	0	0	0	0	OC3M3	OC3M2	OC3M1	OC3M0
	Output Comp 3 Mask Reg	W								
0xC3	OC3D	R	0	0	0	0	OC3D3	OC3D2	OC3D1	OC3D0
	Output Comp 3 Data Reg	W								
0xC4	TCNT (hi)	R	tcnt 15	tcnt 14	tcnt 13	tcnt 12	tcnt 11	tcnt 10	tcnt 9	tcnt 8
	Timer Count Register	W								
0xC5	TCNT (lo)	R	tcnt 7	tcnt 6	tcnt 5	tcnt 4	tcnt 3	tcnt 2	tcnt 1	tcnt 0
	Timer Count Register	W								
0xC6	TSCR1	R	TEN	0	0	TFFCA	0	0	0	0
	Timer System Control Reg 1	W								
0xC7	TTOV	R	0	0	0	0	TOV3	TOV2	TOV1	TOV0
	Timer Toggle Overflow Reg	W								
0xC8	TCTL1	R	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0
	Timer Control Register 1	W								

4.5.2 Low Power Mode Operation

The D2D module is disabled in SLEEP mode. In Stop mode, the D2DINT signal is used to wake-up a powered down MCU. As the MCU could wake up without the S/MM912F634 analog die, a special command is recognized as a wake-up event during Stop mode. See Section 4.3, “Modes of Operation".

4.5.2.1 Normal Mode / Stop Mode

NOTE

The maximum allowed clock speed of the interface is limited to f_{D2D}.

While in Normal or Stop mode, D2DCLK acts as input only with pull present. D2D[3:0] operates as an input/output with pull-down always present. D2DINT acts as output only.

4.5.2.2 Sleep Mode

While in Sleep mode, all Interface data pins are pulled down to DGND to reduce power consumption.

4.6 Interrupts

Interrupts are used to signal a microcontroller that a peripheral needs to be serviced. While in Stop mode, the interrupt signal is used to signal Wake-up events. The interrupts are signaled by an active high level of the D2DINT pin, which remains high until the interrupt is acknowledged via the D2D-Interface. Interrupts are only asserted while in Normal mode.

4.6.1 Interrupt Source Identification

Once an Interrupt is signalized, there are two options to identify the corresponding source(s).

4.6.1.1 Interrupt Source Mirror

NOTE

The VSI - Voltage Status Interrupt combines the five status flags for the Low Battery Interrupt, Low Voltage Interrupt, High Voltage Interrupt, Voltage Regulator Overvoltage Interrupt, and the Voltage Regulator High Temperature Interrupt. The specific source can be identified by reading the Voltage Status Register - VSR.

All Interrupt sources in S/MM912F634 analog die are mirrored to a special Interrupt Source Register (ISR). This register is read only and indicates all currently pending Interrupts. Reading this register does not acknowledge any interrupt. An additional D2D access is necessary to serve the specific module.

4.6.1.1.1 Interrupt Source Register (ISR)

Table 78. Interrupt Source Register (ISR)

Offset[71] 0x00 (0x00 and 0x01 for 8Bit access) Access: User read

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	HOT	LSOT	HSOT	LINOT	SCI	RX	TX	ERR	TOV	CH3	CH2	CH1	CH0	VSI
W																

Note:

71. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 82. Interrupt Source Priority

Interrupt Source	IRQ	Priority
no interrupt pending or wake-up from Stop mode	0x00	1 (highest)
LVI - Low Voltage Interrupt	0x02	2
HTI - Voltage Regulator High Temperature Interrupt	0x04	3
LBI - Low Battery Interrupt	0x06	4
CH0 - TIM Channel 0 Interrupt	0x08	5
CH1 - TIM Channel 1 Interrupt	0x0A	6
CH2 - TIM Channel 2 Interrupt	0x0C	7
CH3 - TIM Channel 3 Interrupt	0x0E	8
TOV - Timer Overflow Interrupt	0x10	9
ERR - SCI Error Interrupt	0x12	10
TX - SCI Transmit Interrupt	0x14	11
RX - SCI Receive Interrupt	0x16	12
SCI - ADC Sequence Complete Interrupt	0x18	13
LINOT - LIN Driver Overtemperature Interrupt	0x1A	14
HSOT - High-side Overtemperature Interrupt	0x1C	15
LSOT - Low-side Overtemperature Interrupt	0x1E	16
HOT - HSUP Overtemperature Interrupt	0x20	17
HVI - High Voltage Interrupt	0x22	18
VROVI - Voltage Regulator Overvoltage Interrupt	0x24	19 (lowest)

4.6.2 Interrupt Sources

4.6.2.1 Voltage Status Interrupt (VSI)

The Voltage Status Interrupt - VSI combines the five interrupt sources of the Voltage Status Register. It is only available in the Interrupt Source Register (ISR). Acknowledge the interrupt by reading the Voltage Status Register - VSR. To issue a new interrupt, the condition has to vanish and occur again. See Section 4.4, “Power Supply" for details on the Voltage Status Register including masking information.

4.6.2.2 Low Voltage Interrupt (LVI)

Acknowledge the interrupt by reading the Voltage Status Register - VSR. To issue a new interrupt, the condition has to vanish and occur again. See Section 4.4, “Power Supply" for details on the Voltage Status Register including masking information.

4.6.2.3 Voltage Regulator High Temperature Interrupt (HTI)

Acknowledge the interrupt by reading the Voltage Status Register - VSR. To issue a new interrupt, the condition has to vanish and occur again. See Section 4.4, “Power Supply" for details on the Voltage Status Register including masking information.

4.6.2.4 Low Battery Interrupt (LBI)

Acknowledge the interrupt by reading the Voltage Status Register - VSR. To issue a new interrupt, the condition has to vanish and occur again. See Section 4.4, “Power Supply" for details on the Voltage Status Register including masking information.

4.6.2.5 TIM Channel 0 Interrupt (CH0)

See Section 4.18, “Basic Timer Module - TIM (TIM16B4C)".

4.12.1.1 Block Diagram

The following Figure shows the basic structure of the LS drivers.

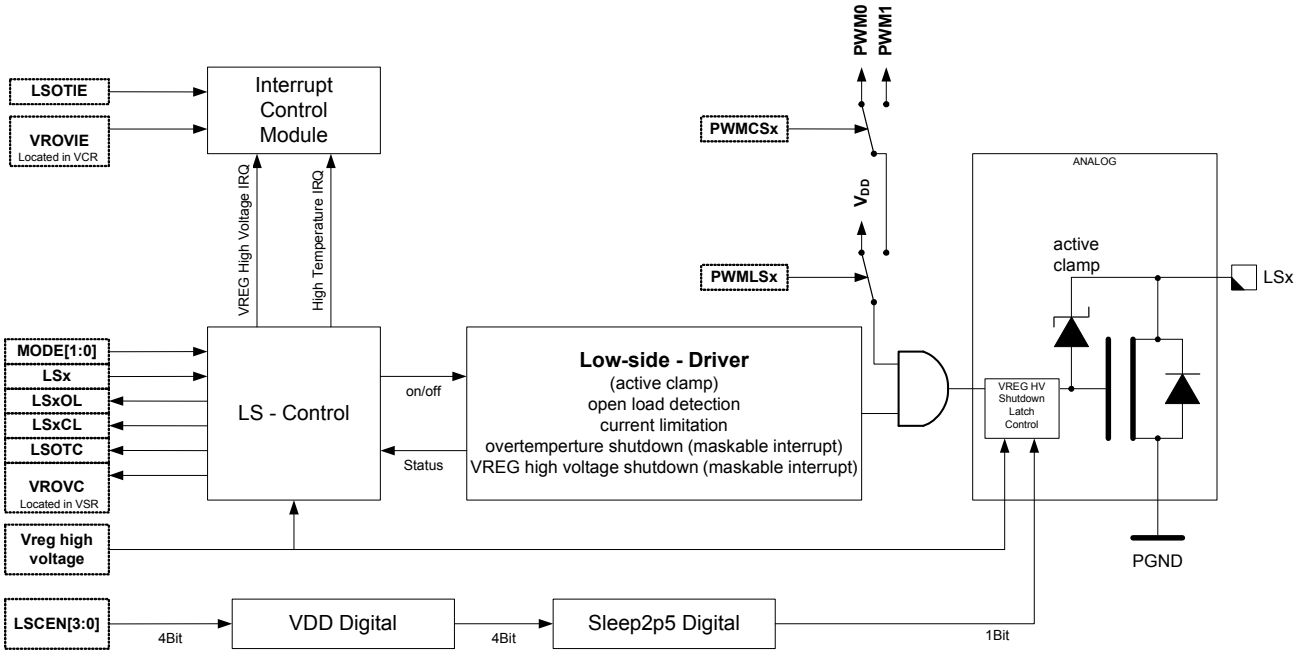

Figure 21. Low-side Drivers - Block Diagram

4.12.1.2 Modes of Operation

The Low-side module is active only in Normal mode; the Low-side drivers are disabled in Sleep and Stop mode.

4.12.2 External Signal Description

This section lists and describes the signals which do connect off-chip. Table 101 shows all the pins and their functions controlled by the Low-side module.

Table 101. Pin Functions and Priorities

Pin Name	Pin Function & Priority	I/O	Description	Pin Function after Reset
LS1	High Voltage Output	O	Low-side Power Output Driver, Active Clamping	LS1
LS2		O		LS2

4.12.3.2.2 Low-side Status Register (LSSR)

Table 105. Low-side Status Register (LSSR)

Offset[86] 0x31 Access: User read

	7	6	5	4	3	2	1	0
R	LSOTC	0	0	0	LS2CL	LS1CL	LS2OL	LS1OL
W								
Reset	0	0	0	0	0	0	0	0

Note:
86. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 106. LSSR - Register Field Descriptions

Field	Description
7 - LSOTC	Low-side Overtemperature condition present. Both drivers are turned off. Reading the register clears the LSOT interrupt flag if present. See Section 4.6, "Interrupts" for details.
3 - LS2CL	Low-side 2 Current Limitation
2 - LS1CL	Low-side 1 Current Limitation
1 - LS2OL	Low-side 2 Open Load [87]
0 - LS1OL	Low-side 1Open Load [87]

Note:
87. When the Low-side is in OFF state, the Open Load Detection function is not operating. When reading the LSSR register while the Low-side is operating in PWM and is in the OFF state, the LS1OL and LS2OL bits does not indicate Open Load.

4.12.3.2.3 Low-side Control Enable Register (LSCEN)

Table 107. Low-side Enable Register (LSEN)

Offset[88] 0x32 Access: User read/write

	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	**0**	**0**	**0**	**0**	**LSCEN**			
W								
Reset	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Note:
88. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 108. LSEN - Register Field Descriptions

Field	Description
3-0 LSCEN	Low-side Control Enable - To allow the LS Control via LSx, the correct value has to be written into the LSCEN Register. 0x5 - Low-side Control Enabled all other values - Low-side Control Disabled

4.12.4 Functional Description

The Low-side switches are controlled by the bits LS1:2 in the Low-side Control Register (LSCR). In order to control the Low-sides, the LSCEN register has to be correctly written once after RESET or VROV.

To protect the device against overvoltage when an inductive load (relay) is turned off an active clamp circuit is implemented.

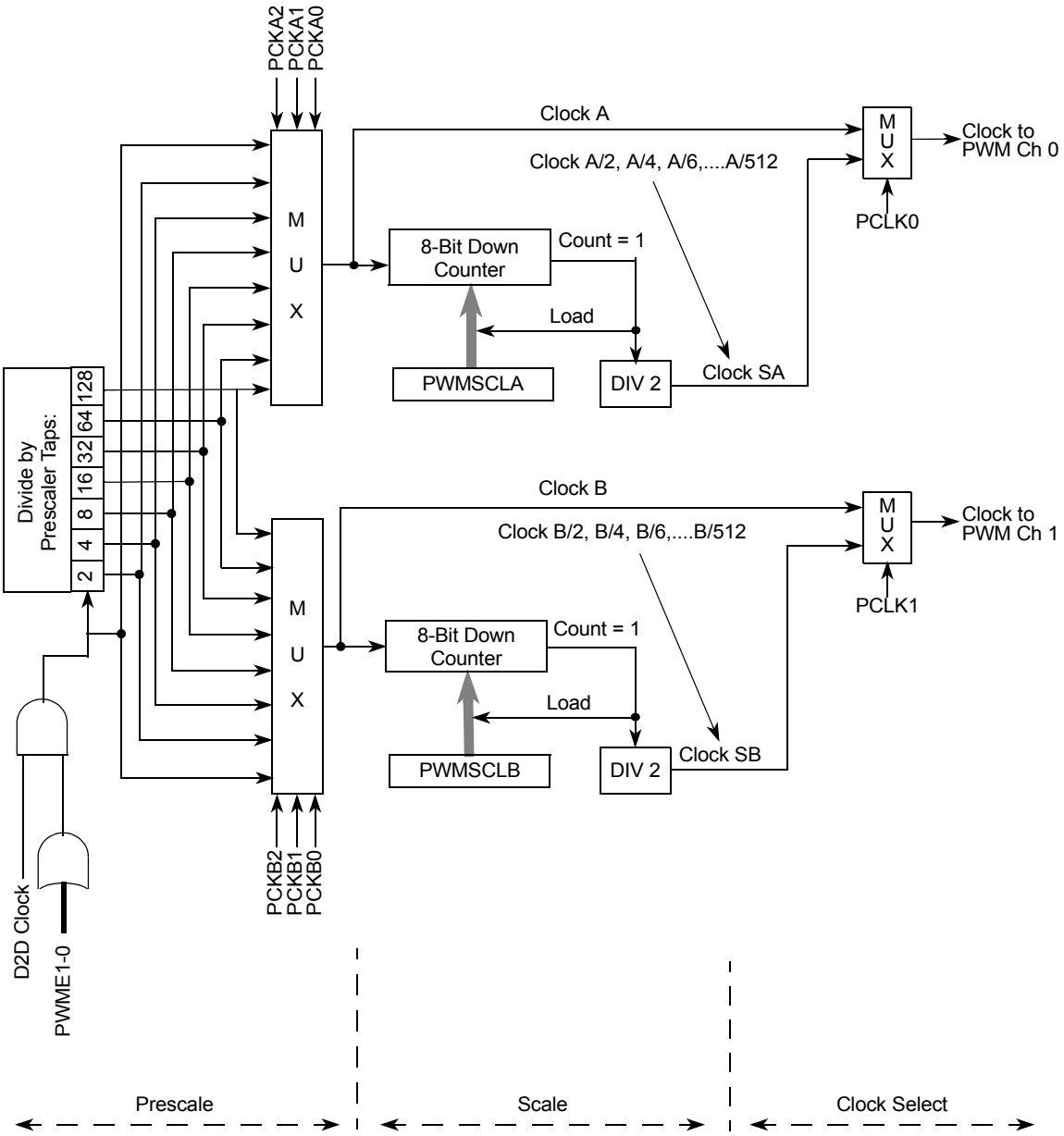

Figure 26. PWM Clock Select Block Diagram

NOTE

Clock SA = Clock A / (2 * PWMSCLA)

When PWMSCLA = $00, PWMSCLA value is considered a full scale value of 256. Clock A is thus divided by 512.

Clock A is used as an input to an 8-bit down counter. This down counter loads a user programmable scale value from the scale register (PWMSCLA). When the down counter reaches one, a pulse is output and the 8-bit counter is re-loaded. The output signal from this circuit is further divided by two. This gives a greater range with only a slight reduction in granularity. Clock SA equals clock A divided by two times the value in the PWMSCLA register.

NOTE

Clock SB = Clock B / (2 * PWMSCLB)

When PWMSCLB = $00, PWMSCLB value is considered a full scale value of 256. Clock B is thus divided by 512.

9-bit data mode typically is used in conjunction with parity to allow eight bits of data plus the parity in the ninth bit. Or it is used with address-mark wake-up so the ninth data bit can serve as the wake-up bit. In custom protocols, the ninth bit can also serve as a software-controlled marker.

4.15.3.5.2 Stop Mode Operation

During all stop modes, clocks to the SCI module are halted. In stop1 and stop2 modes, all SCI register data is lost and must be re-initialized upon recovery from these two stop modes. No SCI module registers are affected in stop3 mode.

The receive input active edge detect circuit is still active in stop3 mode, but not in stop2. An active edge on the receive input brings the CPU out of stop3 mode if the interrupt is not masked (RXEDGIE = 1).

Note that because the clocks are halted, the SCI module resumes operation upon exit from stop (only in stop3 mode). Software should ensure stop mode is not entered while there is a character being transmitted out of or received into the SCI module.

4.15.3.5.3 Loop Mode

When LOOPS = 1, the RSRC bit in the same register chooses between loop mode (RSRC = 0) or single-wire mode (RSRC = 1). Loop mode is sometimes used to check software, independent of connections in the external system, to help isolate system problems. In this mode, the transmitter output is internally connected to the receiver input and the RxD pin is not used by the SCI, so it reverts to a general purpose port I/O pin.

4.15.3.5.4 Single-wire Operation

When LOOPS = 1, the RSRC bit in the same register chooses between loop mode (RSRC = 0) or single-wire mode (RSRC = 1). Single-wire mode is used to implement a half-duplex serial connection. The receiver is internally connected to the transmitter output and to the TxD pin. The RxD pin is not used and reverts to a general purpose port I/O pin.

In single-wire mode, the TXDIR bit in SCIC3 controls the direction of serial data on the TxD pin. When TXDIR = 0, the TxD pin is an input to the SCI receiver and the transmitter is temporarily disconnected from the TxD pin so an external device can send serial data to the receiver. When TXDIR = 1, the TxD pin is an output driven by the transmitter. In single-wire mode, the internal loop back connection from the transmitter to the receiver causes the receiver to receive characters sent out by the transmitter.

4.16 High Voltage Inputs - Lx

Six High Voltage capable inputs are implemented with the following features:

- Digital Input Capable
- Analog Input Capable with selectable voltage divider.
- Wake-up Capable during Low Power mode. See Section 4.8, “Wake-up / Cyclic Sense".

When used as analog inputs to sense voltages outside the module a series resistor must be used on the used input. When a Lx input is not selected in the analog multiplexer, the voltage divider is disconnected from this input. When a Lx input is selected in the analog multiplexer, it is disconnected in low power mode if configured as Wake-up input. Unused Lx pins are recommended to be connected to GND to improve EMC behavior.

4.16.1 Register Definition

4.16.1.1 Lx Status Register (LXR)

Table 140. Lx Status Register (LXR)

Offset[109] 0x08 Access: User read

	7	6	5	4	3	2	1	0
R	0	0	L5	L4	L3	L2	L1	L0
W								

Note:
109. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

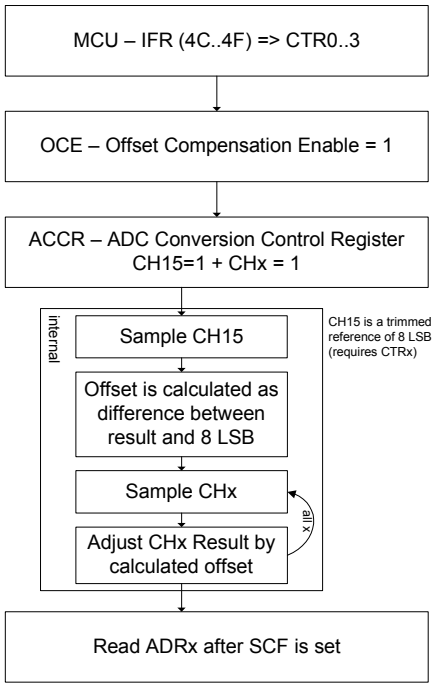

Figure 39. Automatic Offset Compensation

4.19.5.3 Conversion Timing

The conversion timing is based on the ADCCLK generated by the ADC prescaler (PS) out of the D2DCLK signal. The prescaler needs to be configured to have the ADCCLK match the specified f_{ADC} clock limits. A conversion is divided into the following 27+ clock cycles:

- 9 cycle sampling time
- 18 cycle remaining conversion time
- A worst case (only channel 14) of 15 clock cycles to count up to the selected channel (15, 0, 1,....14)
- 4 cycles between two channels

Example 1. Single Conversion Channel 10 (VSENSE)

12c (count up to Ch10) + 9c (sample) + 18c (conversion) = 39 cycles from start to end of conversion.

Example 2. Sequence of Channel 10 (VSENSE) + Channel 15 (Offset Compensation)

1c (count) + 9c (sample Ch15) + 18c (conversion Ch15) + 4c (in between) + 0c (count further to Ch10 is performed while converting ch15) + 9c (sample) + 18c (conversion) = 59 cycles from start to end of both conversions.

4.20 Current Sense Module - ISENSE

The Current Sense Module is implemented to amplify the voltage drop across an external shunt resistor to measure the actual application current using the internal Analog Digital Converter Channel 9. Typical application is the motor current in a window lift control module

4.28.1.4 Modes of Operation

This subsection lists and briefly describes all operating modes supported by the MMC.

4.28.1.4.1 Power Saving Modes

- Run mode
  MMC is functional during normal run mode.
- Wait mode
  MMC is functional during wait mode.
- Stop mode
  MMC is inactive during stop mode.

4.28.1.4.2 Functional Modes

- Single chip modes
  In normal and special single chip mode the internal memory is used.

4.28.1.5 Block Diagram

Figure 48 shows a block diagram of the MMC.

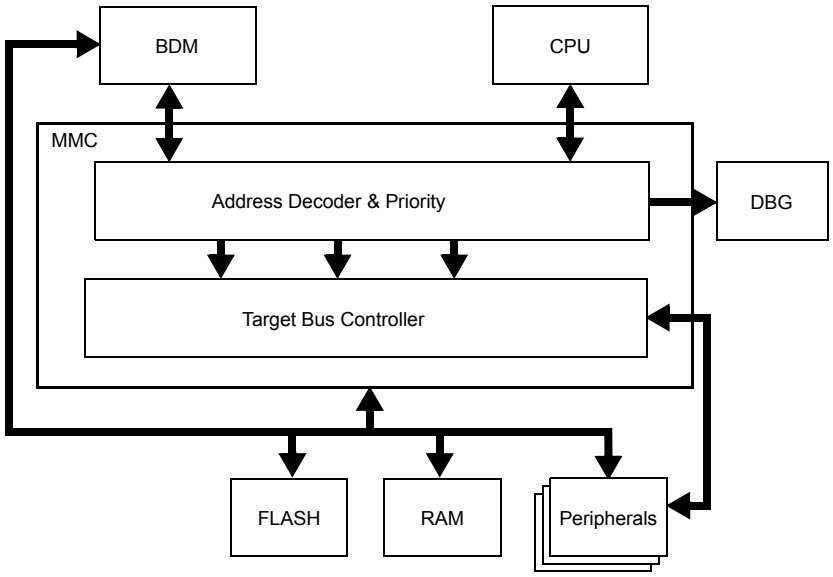

Figure 48. MMC Block Diagram External Signal Description

The user is advised to refer to the SoC Guide for port configuration and location of external bus signals. Some pins may not be bonded out in all implementations. Table 239 outlines the pin names and functions. It also provides a brief description of their operation.

Table 239. External Input Signals Associated with the MMC

Signal	I/O	Description	Availability
MODC	I	Mode input	Latched after $\overline{\text{RESET}}$ (active low)

4.28.2 Memory Map and Registers

4.28.2.1 Module Memory Map

A summary of the registers associated with the MMC block is shown in Figure 240. Detailed descriptions of the registers and bits are given in the following subsections.

4.31.1.1 Glossary Of Terms

COF — Change Of Flow. Change in the program flow due to a conditional branch, indexed jump or interrupt.
BDM — Background Debug mode
S12SBDM — Background Debug module
WORD — 16 bit data entity
Data Line — 20 bit data entity
CPU — S12SCPU module
DBG — S12SDBG module
Tag — Tags can be attached to CPU opcodes as they enter the instruction pipe. If the tagged opcode reaches the execution stage a tag hit occurs.

4.31.1.2 Overview

The comparators monitor the bus activity of the CPU module. A match can initiate a state sequencer transition. On a transition to the Final State, bus tracing is triggered and/or a breakpoint can be generated. Independent of comparator matches a transition to Final State with associated tracing and breakpoint can be triggered immediately by writing to the TRIG control bit. The trace buffer is visible through a 2-byte window in the register address map and can be read out using standard 16-bit word reads. Tracing is disabled when the MCU system is secured.

4.31.1.3 Features

- Three comparators (A, B, and C)
  - Comparators A compares the full address bus and full 16-bit data bus
  - Comparator A features a data bus mask register
  - Comparators B and C compare the full address bus only
  - Each comparator features selection of read or write access cycles
  - Comparator B allows selection of byte or word access cycles
  - Comparator matches can initiate state sequencer transitions
- Three comparator modes
  - Simple address/data comparator match mode
  - Inside address range mode, Addmin ≤ Address ≤ Addmax
  - Outside address range match mode, Address < Addmin or Address > Addmax
- Two types of matches
  - Tagged — This matches just before a specific instruction begins execution
  - Force — This is valid on the first instruction boundary after a match occurs
- Two types of breakpoints
  - CPU breakpoint entering BDM on breakpoint (BDM)
  - CPU breakpoint executing SWI on breakpoint (SWI)
- Trigger mode independent of comparators
  - TRIG Immediate software trigger
- Four trace modes
  - Normal: change of flow (COF) PC information is stored (see Section 4.31.4.5.3, “Normal Mode") for change of flow definition.
  - Loop1: same as Normal but inhibits consecutive duplicate source address entries
  - Detail: address and data for all cycles except free cycles and opcode fetches are stored
  - Pure PC: All program counter addresses are stored
- 4-stage state sequencer for trace buffer control
  - Tracing session trigger linked to Final State of state sequencer
  - Begin and End alignment of tracing to trigger

4.31.1.4 Modes of Operation

The DBG module can be used in all MCU functional modes. During BDM hardware accesses and whilst the BDM module is active, CPU monitoring is disabled. When the CPU enters active BDM Mode through a BACKGROUND command, the DBG module, if already armed, remains armed. The DBG module tracing is disabled if the MCU is secure, however, breakpoints can still be generated

4.31.3.2.3 Debug Trace Control Register (DBGTCR)

Table 269. Debug Trace Control Register (DBGTCR)

Address: 0x0022

	7	6	5	4	3	2	1	0
R	0	TSOURCE	0	0	TRCMOD		0	TALIGN
W								
Reset	0	0	0	0	0	0	0	0

Read: Anytime
Write: Bit 6 only when DBG is neither secure nor armed. Bits 3,2,0 anytime the module is disarmed.

Table 270. DBGTCR Field Descriptions

Field	Description
6 TSOURCE	**Trace Source Control Bit** — The TSOURCE bit enables a tracing session given a trigger condition. If the MCU system is secured, this bit cannot be set and tracing is inhibited. This bit must be set to read the trace buffer. 0 Debug session without tracing requested 1 Debug session with tracing requested
3–2 TRCMOD	**Trace Mode Bits** — See Section 4.31.4.5.2, “Trace Modes" 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. In Pure PC mode the program counter value for each instruction executed is stored. See Table 271.
0 TALIGN	**Trigger Align Bit** — This bit controls whether the trigger is aligned to the beginning or end of a tracing session. 0 Trigger at end of stored data 1 Trigger before storing data

Table 271. TRCMOD Trace Mode Bit Encoding

TRCMOD	Description
00	Normal
01	Loop1
10	Detail
11	Pure PC

4.31.3.2.4 Debug Control Register2 (DBGC2)

Table 272. Debug Control Register2 (DBGC2)

Address: 0x0023

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

Read: Anytime
Write: Anytime the module is disarmed.

This register configures the comparators for range matching.

Table 280. State Control Register Access Encoding

COMRV	Visible State Control Register
00	DBGSCR1
01	DBGSCR2
10	DBGSCR3
11	DBGMFR

4.31.3.2.7.1 Debug State Control Register 1 (DBGSCR1)

Table 281. Debug State Control Register 1 (DBGSCR1)

Address: 0x0027

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	SC2	SC1	SC0
W								
Reset	0	0	0	0	0	0	0	0

Read: If COMRV[1:0] = 00
Write: If COMRV[1:0] = 00 and DBG is not armed.

This register is visible at 0x0027 only with COMRV[1:0] = 00. The state control register 1 selects the targeted next state whilst in State1. The matches refer to the match channels of the comparator match control logic as depicted in Figure 67 and described in Section 4.31.3.2.8.1, “Debug Comparator Control Register (DBGXCTL)””. Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

Table 282. DBGSCR1 Field Descriptions

Field	Description
2–0 SC[2:0]	These bits select the targeted next state whilst in State1, based upon the match event.

Table 283. State1 Sequencer Next State Selection

SC[2:0]	Description
000	Any match to Final State
001	Match1 to State3
010	Match2 to State2
011	Match1 to State2
100	Match0 to State2..... Match1 to State3
101	Match1 to State3..... Match0 Final State
110	Match0 to State2..... Match2 to State3
111	Either Match0 or Match1 to State2..... Match2 has no effect

The priorities described in Table 314 dictate in the case of simultaneous matches, the match on the lower channel number (0,1,2) has priority. The SC[2:0] encoding ensures a match leading to final state has priority over all other matches.

4.31.4.5.3.2 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 indicates 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.

4.31.4.5.3.3 Pure PC Mode

NOTE:
When tracing is terminated using forced breakpoints, latency in breakpoint generation means opcodes following the opcode causing the breakpoint can be stored to the trace buffer. The number of opcodes is dependent on program flow. This should be avoided by using tagged breakpoints.

In Pure PC mode, tracing from the CPU the PC addresses of all executed opcodes, including illegal opcodes are stored.

4.31.4.5.4 Trace Buffer Organization

ADRH, ADRM, and ADRL denote address high, middle, and low byte respectively. CRW and CSZ indicate R/W and size access information. The numerical suffix refers to the tracing count. The information format for Loop1, Pure PC and Normal modes is identical. In Detail mode, the address and data for each entry are stored on consecutive lines. Thus, the maximum number of entries is 32. In this case, DBGCNT bits are incremented twice, once for the address line and once for the data line, on each trace buffer entry.

Single byte data accesses in Detail mode is always stored to the low byte of the trace buffer (DATAL) and the high byte is cleared. When tracing word accesses, the byte at the lower address is always stored to trace buffer byte1, and the byte at the higher address is stored to byte0.

Table 315. Trace Buffer Organization (20-bit wide buffer)

Mode	Entry Number	4-bits	8-bits	8-bits
		Field 2	Field 1	Field 0
Detail Mode	Entry 1	CSZ1,CRW1,ADRH1	ADRM1	ADRL1
		0	DATAH1	DATAL1
	Entry 2	CSZ2,CRW2,ADRH2	ADRM2	ADRL2
		0	DATAH2	DATAL2
Other Modes	Entry 1	PCH1	PCM1	PCL1
	Entry 2	PCH2	PCM2	PCL2

4.31.4.5.4.1 Information Bit Organization

The format of the bits is dependent upon the active trace mode, as described by the following.

4.31.4.5.4.2 Field2 Bits in Detail Mode

Table 316. Field2 Information Bits in Detail Mode

Bit 3	Bit 2	Bit 1	Bit 0
CSZ	CRW	ADRH17	ADRH16

In Detail mode, the CSZ and CRW bits indicate the type of access being made by the CPU.

Table 317. Field Descriptions

Field	Description
3 CSZ	**Access Type Indicator**— This bit indicates if the access was a byte or word size when tracing in Detail mode 0 Word Access 1 Byte Access
2 CRW	**Read Write Indicator** — This bit indicates if the corresponding stored address corresponds to a read or write access when tracing in Detail Mode. 0 Write Access 1 Read Access
1 ADRH17	**Address Bus bit 17**— Corresponds to system address bus bit 17.
0 ADRH16	**Address Bus bit 16**— Corresponds to system address bus bit 16.

4.31.4.5.4.3 Field2 Bits in Normal, Pure PC and Loop1 Modes

Table 318. Information Bits PCH

Bit 3	Bit 2	Bit 1	Bit 0
CSD	CVA	PC17	PC16

Table 319. PCH Field Descriptions

Field	Description
3 CSD	**Source Destination Indicator** — In Normal and Loop1 mode this bit indicates if the corresponding stored address is a source or destination address. This bit has no meaning in Pure PC mode. 0 Source Address 1 Destination Address
2 CVA	**Vector Indicator** — In Normal and Loop1 mode this bit indicates if the corresponding stored address is a vector address. Vector addresses are destination addresses, thus if CVA is set, then the corresponding CSD is also set. This bit has no meaning in Pure PC mode. 0 Non-Vector Destination Address 1 Vector Destination Address
1 PC17	**Program Counter bit 17**— In Normal, Pure PC, and Loop1 mode this bit corresponds to program counter bit 17.
0 PC16	**Program Counter bit 16**— In Normal, Pure PC, and Loop1 mode this bit corresponds to program counter bit 16.

4.31.4.5.5 Reading Data from Trace Buffer

The data stored in the Trace Buffer can be read, provided the DBG module is not armed, is configured for tracing (TSOURCE bit is set) and the system not secured. When the ARM bit is written to 1 the trace buffer is locked to prevent reading. The trace buffer can only be unlocked for reading by a single aligned word write to DBGTB when the module is disarmed.

The Trace Buffer can only be read through the DBGTB register using aligned word reads, any byte or misaligned reads return 0 and do not cause the trace buffer pointer to increment to the next trace buffer address. The Trace Buffer data is read out first-in first-out. By reading CNT in DBGCNT, the number of valid lines can be determined. DBGCNT does not decrement as data is read.

Whilst reading, an internal pointer is used to determine the next line to be read. After a tracing session, the pointer points to the oldest data entry, thus if no overflow has occurred, the pointer points to line0, otherwise it points to the line with the oldest entry. The pointer is initialized by each aligned write to DBGTBH to point to the oldest data again. This enables an interrupted trace buffer read sequence to be easily restarted from the oldest data entry.

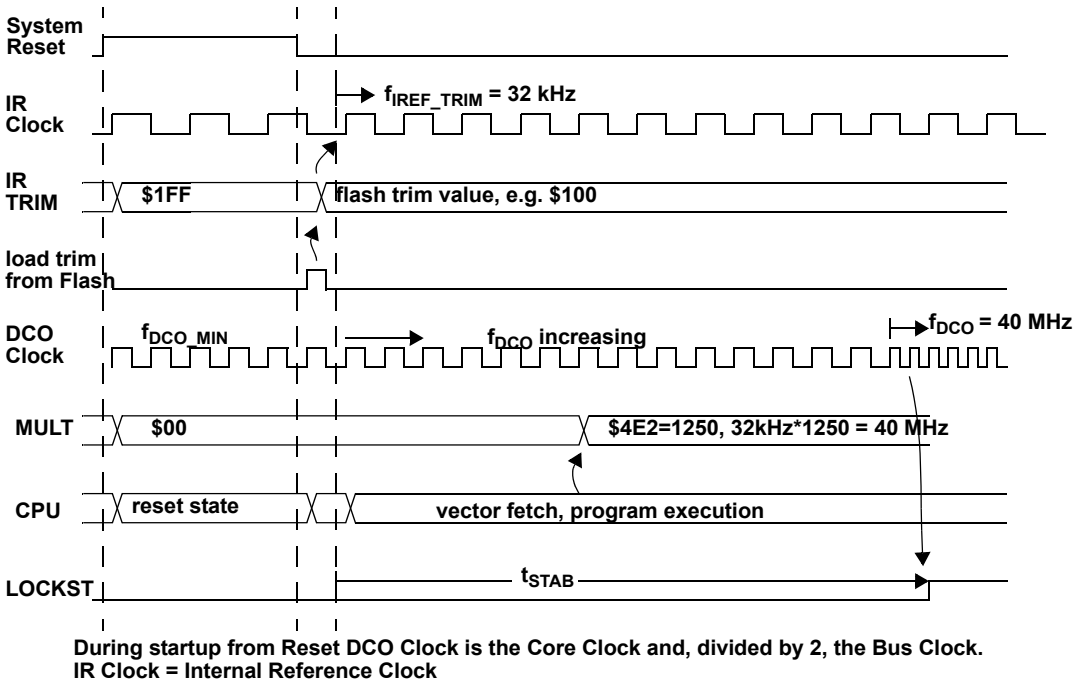

Figure 71. Example for Startup of Clock System After Reset

4.32.4.2 Stop Mode Using DCO Clock as a Bus Clock

An example of what happens going into stop mode and exiting stop mode after an interrupt is shown in Figure 72.

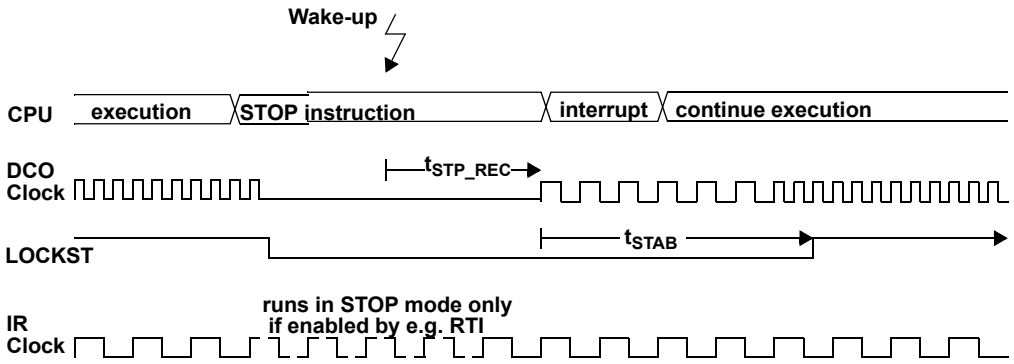

Figure 72. Example of STOP Mode Using DCO Clock as Bus Clock

4.32.4.3 Stop Mode Using Oscillator Clock as Bus Clock

An example of what happens going into stop mode and exiting stop mode after an interrupt is shown in Figure 73.

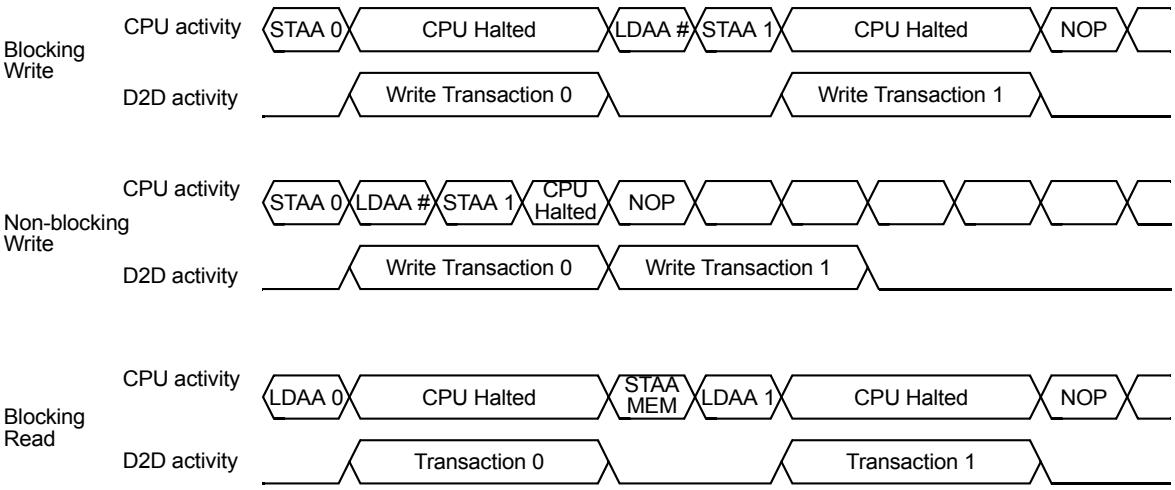

Figure 94. Blocking and Non-blocking Transfers.

4.37.4.2.1 Blocking Writes

When writing to the address window associated with blocking transactions, the CPU is held until the transaction is completed, before completing the instruction. Figure 94 shows the behavior of the CPU for a blocking write transaction shown in the following example.

```
STAA    BLK_WINDOW+OFFS0 ; WRITE0 8-bit as a blocking transaction
LDAA    #BYTE1
STAA    BLK_WINDOW+OFFS1 ; WRITE1 is executed after WRITE0 transaction is completed
NOP
```

Blocking writes should be used when clearing interrupt flags located in the target, or other writes which require the operation at the target is completed before proceeding with the CPU instruction stream.

4.37.4.2.2 Non-blocking Writes

When writing to the address window associated with non-blocking transactions, the CPU can continue before the transaction is completed. However, if there was a transaction ongoing when doing the 2nd write, the CPU is held until the first one is completed before executing the 2nd one. Figure 94 shows the behavior of the CPU for a blocking write transaction shown in the following example.

```
STAA    NONBLK_WINDOW+OFFS0; write 8-bit as a blocking transaction
LDAA    #BYTE1              ; load next byte
STAA    NONBLK_WINDOW+OFFS1; executed right after the first
NOP
```

As Figure 94 illustrates, non-blocking writes have a performance advantage, but care must be taken the following instructions are not affected by the change in the target caused by the previous transaction.

4.37.4.2.3 Blocking Read

When reading from the address window associated with blocking transactions, the CPU is held until the data is returned from the target, before completing the instruction.Figure 94 shows the behavior of the CPU for a blocking read transaction shown in the following example.

```
LDAA    BLK_WINDOW+OFFS0 ; Read 8-bit as a blocking transaction
STAA    MEM              ; Store result to local Memory
LDAA    BLK_WINDOW+OFFS1 ; Read 8-bit as a blocking transaction
```

4.37.4.2.4 Non-blocking Read

Read access to the non-blocking window is reserved for future use. When reading from the address window associated with non-blocking writes, the read returns an all 0s data byte or word. This behavior can change in future revisions.

4.38.3.2.4 SPI Status Register (SPISR)

Table 415. SPI Status Register (SPISR)

0x00EB

	7	6	5	4	3	2	1	0
R	SPIF	0	SPTEF	MODF	0	0	0	0
W								
Reset	0	0	1	0	0	0	0	0

Read: Anytime
Write: Has no effect

Table 416. SPISR Field Descriptions

Field	Description
7 SPIF	**SPIF Interrupt Flag** — This bit is set after a received data byte has been transferred into the SPI data register. This bit is cleared by reading the SPISR register (with SPIF set) followed by a read access to the SPI data register. 0 Transfer not yet complete. 1 New data copied to SPIDR.
5 SPTEF	**SPI Transmit Empty Interrupt Flag** — If set, this bit indicates the transmit data register is empty. To clear this bit and place data into the transmit data register, SPISR must be read with SPTEF = 1, followed by a write to SPIDR. Any write to the SPI data register without reading SPTEF = 1, is effectively ignored. 0 SPI data register not empty. 1 SPI data register empty.
4 MODF	**Mode Fault Flag** — This bit is set if the $\overline{SS}$ input becomes low, while the SPI is configured as a master and mode fault detection is enabled, the MODFEN bit of SPICR2 register is set. Refer to MODFEN bit description in Section 4.38.3.2.2, "SPI Control Register 2 (SPICR2)". The flag is cleared automatically by a read of the SPI status register (with MODF set) followed by a write to the SPI control register 1. 0 Mode fault has not occurred. 1 Mode fault has occurred.

4.38.3.2.5 SPI Data Register (SPIDR)

Table 417. SPI Data Register (SPIDR)

0x00ED

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

Read: Anytime; normally read only when SPIF is set
Write: Anytime

The SPI data register is both the input and output register for SPI data. A write to this register allows a data byte to be queued and transmitted. For an SPI configured as a master, a queued data byte is transmitted immediately after the previous transmission has completed. The SPI transmitter empty flag SPTEF in the SPISR register indicates when the SPI data register is ready to accept new data.

Received data in the SPIDR is valid when SPIF is set. If SPIF is cleared and a byte has been received, the received byte is transferred from the receive shift register to the SPIDR and SPIF is set. If SPIF is set and not serviced, and a second byte has been received, the second received byte is kept as valid byte in the receive shift register until the start of another transmission. The byte in the SPIDR does not change.

- The SCK is output for the master mode and input for the slave mode.
- The $\overline{SS}$ is the input or output for the master mode, and it is always the input for the slave mode.
- The bidirectional mode does not affect SCK and $\overline{SS}$ functions.

4.38.4.6 Error Conditions

The SPI has one error condition:

- Mode fault error

4.38.4.6.1 Mode Fault Error

NOTE

If a mode fault error occurs and a received data byte is pending in the receive shift register, this data byte is lost.

If the $\overline{SS}$ input becomes low while the SPI is configured as a master, it indicates a system error where more than one master may be trying to drive the MOSI and SCK lines simultaneously. This condition is not permitted in normal operation. The MODF bit in the SPI status register is set automatically, provided the MODFEN bit is set.

In the special case where the SPI is in master mode and MODFEN bit is cleared, the $\overline{SS}$ pin is not used by the SPI. In this case, the mode fault error function is inhibited and MODF remains cleared. In case the SPI system is configured as a slave, the $\overline{SS}$ pin is a dedicated input pin. Mode fault error doesn't occur in slave mode.

If a mode fault error occurs, the SPI is switched to slave mode, with the exception that the slave output buffer is disabled. So SCK, MISO, and MOSI pins are forced to be high-impedance inputs, to avoid any possibility of conflict with another output driver. A transmission in progress is aborted and the SPI is forced into idle state.

If the mode fault error occurs in the bidirectional mode for a SPI system configured in master mode, output enable of the MOMI (MOSI in bidirectional mode) is cleared if it was set. No mode fault error occurs in the bidirectional mode for SPI system configured in slave mode.

The mode fault flag is cleared automatically by a read of the SPI status register (with MODF set), followed by a write to the SPI control register 1. If the mode fault flag is cleared, the SPI becomes a normal master or slave again.

4.38.4.7 Low Power Mode Options

4.38.4.7.1 SPI in Run Mode

In run mode with the SPI system enable (SPE) bit in the SPI control register clear, the SPI system is in a low-power, disabled state. SPI registers remain accessible, but clocks to the core of this module are disabled.

4.38.4.7.2 SPI in Wait Mode

NOTE

Care must be taken when expecting data from a master while the slave is in wait or stop mode. Even though the shift register continues to operate, the rest of the SPI is shut down (i.e., a SPIF interrupt is **not** generated until exiting stop or wait mode). Also, the byte from the shift register is not copied into the SPIDR register until after the slave SPI has exited wait or stop mode. In slave mode, a received byte pending in the receive shift register is lost when entering wait or stop mode. An SPIF flag and SPIDR copy is generated only if wait mode is entered or exited during a transmission. If the slave enters wait mode in idle mode and exits wait mode in idle mode, neither a SPIF nor a SPIDR copy occurs.

SPI operation in wait mode depends upon the state of the SPISWAI bit in SPI control register 2.

- If SPISWAI is clear, the SPI operates normally when the CPU is in wait mode
- If SPISWAI is set, SPI clock generation ceases and the SPI module enters a power conservation state when the CPU is in wait mode.
- If SPISWAI is set and the SPI is configured for master, any transmission and reception in progress stops at wait mode entry. The transmission and reception resumes when the SPI exits wait mode.