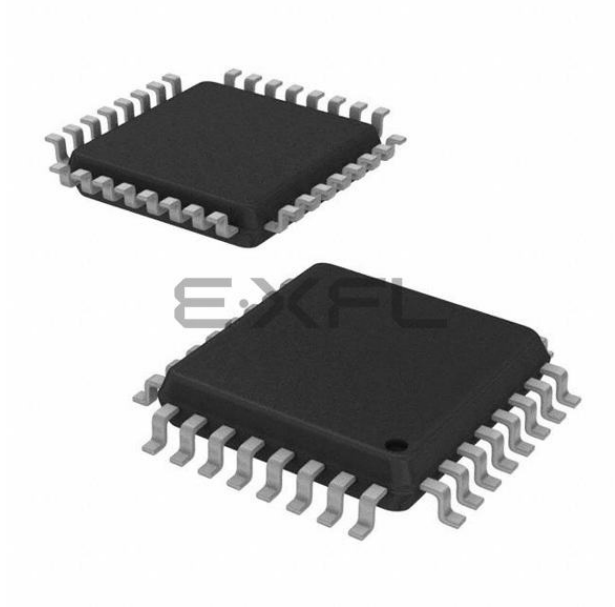


Welcome to E-XFL.COM

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	S12Z
Core Size	16-Bit
Speed	32MHz
Connectivity	CANbus, I²C, IrDA, LINbus, SCI, SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	19
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	5.5V ~ 18V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvl12f0clcr

Chapter 11
Digital Analog Converter (DAC_8B5V_V2)

Table 5-7. BDC Command Types

Command Type	Secure Status	BDC Status	CPU Status	Command Set
Always-available	Secure or Unsecure	Enabled or Disabled	—	• Read/write access to BDCCSR • Mass erase flash memory using ERASE_FLASH • SYNC • ACK enable/disable
Non-intrusive	Unsecure	Enabled	Code execution allowed	• Read/write access to BDCCSR • Memory access • Memory access with status • Mass erase flash memory using ERASE_FLASH • Debug register access • BACKGROUND • SYNC • ACK enable/disable
Active background	Unsecure	Active	Code execution halted	• Read/write access to BDCCSR • Memory access • Memory access with status • Mass erase flash memory using ERASE_FLASH • Debug register access • Read or write CPU registers • Single-step the application • Exit active BDM to return to the application program (GO) • SYNC • ACK enable/disable

Non-intrusive commands are used to read and write target system memory locations and to enter active BDM. Target system memory includes all memory and registers within the global memory map, including external memory.

Active background commands are used to read and write all memory locations and CPU resources. Furthermore they allow single stepping through application code and to exit from active BDM.

Non-intrusive commands can only be executed when the BDC is enabled and the device unsecure. Active background commands can only be executed when the system is not secure and is in active BDM.

Non-intrusive commands do not require the system to be in active BDM for execution, although, they can still be executed in this mode. When executing a non-intrusive command with the ACK pulse handshake protocol disabled, the BDC steals the next bus cycle for the access. If an operation requires multiple cycles, then multiple cycles can be stolen. Thus if stolen cycles are not free cycles, the application code execution is delayed. The delay is negligible because the BDC serial transfer rate dictates that such accesses occur infrequently.

For data read commands, the external host must wait at least 16 BDCSI clock cycles after sending the address before attempting to obtain the read data. This is to be certain that valid data is available in the BDC shift register, ready to be shifted out. For write commands, the external host must wait 16 bdcsi cycles after sending the data to be written before attempting to send a new command. This is to avoid disturbing the BDC shift register before the write has been completed. The external host must wait at least for 16 bdcsi cycles after a control command before starting any new serial command.

drive at the latest after 6 clock cycles, before the target drives a brief high speedup pulse seven target clock cycles after the perceived start of the bit time. The host should sample the bit level about 10 target clock cycles after it started the bit time.

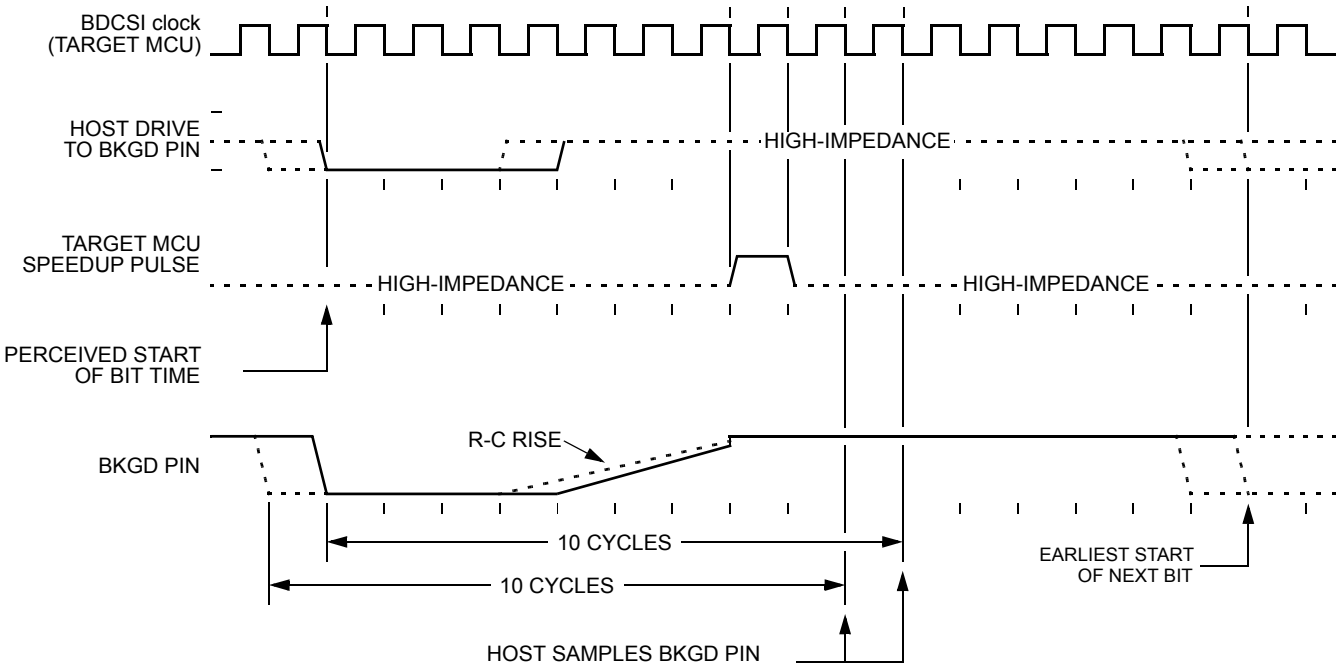


Figure 5-7. BDC Target-to-Host Serial Bit Timing (Logic 1)

Figure 5-8 shows the host receiving a logic 0 from the target. The host initiates the bit time but the target finishes it. Since the target wants the host to receive a logic 0, it drives the BKGD pin low for 13 target clock cycles then briefly drives it high to speed up the rising edge. The host samples the bit level about 10 target clock cycles after starting the bit time.

The handshake protocol is enabled by the ACK_ENABLE command. The BDC sends an ACK pulse when the ACK_ENABLE command has been completed. This feature can be used by the host to evaluate if the target supports the hardware handshake protocol. If an ACK pulse is issued in response to this command, the host knows that the target supports the hardware handshake protocol.

Unlike the normal bit transfer, where the host initiates the transmission by issuing a negative edge on the BKGD pin, the serial interface ACK handshake pulse is initiated by the target MCU by issuing a negative edge on the BKGD pin. Figure 5-9 specifies the timing when the BKGD pin is being driven. The host must follow this timing constraint in order to avoid the risk of an electrical conflict at the BKGD pin.

When the handshake protocol is enabled, the STEAL bit in BDCCSR selects if bus cycle stealing is used to gain immediate access. If STEAL is cleared, the BDC is configured for low priority bus access using free cycles, without stealing cycles. This guarantees that BDC accesses remain truly non-intrusive to not affect the system timing during debugging. If STEAL is set, the BDC gains immediate access, if necessary stealing an internal bus cycle.

NOTE

If bus steals are disabled then a loop with no free cycles cannot allow access. In this case the host must recognize repeated NORESP messages and then issue a BACKGROUND command to stop the target and access the data.

Figure 5-10 shows the ACK handshake protocol without steal in a command level timing diagram. The READ_MEM.B command is used as an example. First, the 8-bit command code is sent by the host, followed by the address of the memory location to be read. The target BDC decodes the command. Then an internal access is requested by the BDC. When a free bus cycle occurs the READ_MEM.B operation is carried out. If no free cycle occurs within 512 core clock cycles then the access is aborted, the NORESP flag is set and the target generates a Long-ACK pulse.

Having retrieved the data, the BDC issues an ACK pulse to the host controller, indicating that the addressed byte is ready to be retrieved. After detecting the ACK pulse, the host initiates the data read part of the command.

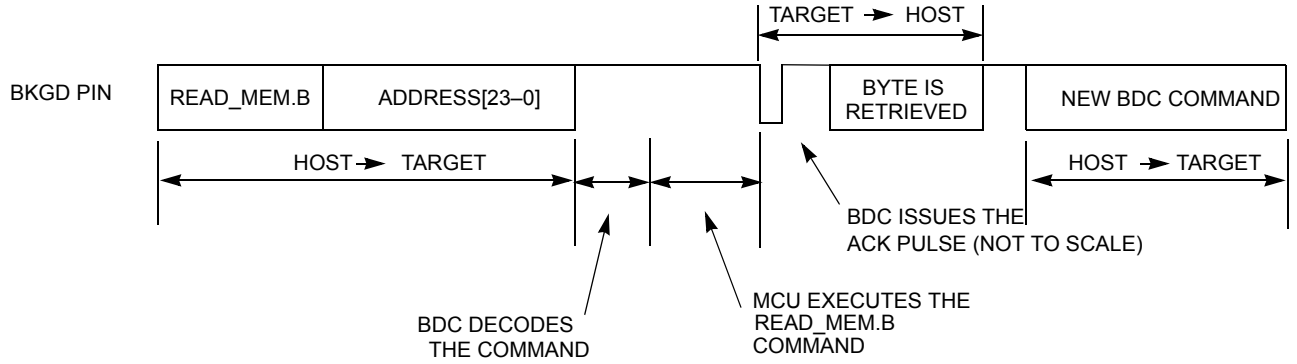


Figure 5-10. Handshake Protocol at Command Level

Alternatively, setting the STEAL bit configures the handshake protocol to make an immediate internal access, independent of free bus cycles.

7.4.3.1 Comparator Match Events

7.4.3.1.1 Opcode Address Comparator Match

The comparator is loaded with the address of the selected instruction and the comparator control register INST bit is set. When the opcode reaches the execution stage of the instruction queue a match occurs just before the instruction executes, allowing a breakpoint immediately before the instruction boundary. The comparator address register must contain the address of the first opcode byte for the match to occur. Opcode address matches are data independent thus the RWE and RW bits are ignored. CPU compares are disabled when BDM becomes active.

7.4.3.1.2 Data Access Comparator Match

Data access matches are generated when an access occurs at the address contained in the comparator address register. The match can be qualified by the access data and by the access type (read/write). The breakpoint occurs a maximum of 2 instructions after the access in the CPU flow. Note, if a COF occurs between access and breakpoint, the opcode address of the breakpoint can be elsewhere in the memory map.

Opcode fetches are not classed as data accesses. Thus data access matches are not possible on opcode fetches.

7.4.3.2 External Event

The DBGEEV input signal can force a state sequencer transition, independent of internal comparator matches. The DBGEEV is an input signal mapped directly to a device pin and configured by the EEVE field in DBGC1. The external events can change the state sequencer state.

If configured to change the state sequencer state, then the external match is mapped to DBGSCRx bits C3SC[1:0]. The DBGEFR bit EEVF is set when an external event occurs.

7.4.3.3 Setting The TRIG Bit

Independent of comparator matches it is possible to initiate a breakpoint by writing the TRIG bit in DBGC1 to a logic “1”. This forces the state sequencer into the Final State. the transition to Final State is followed immediately by a transition to State0.

Breakpoints, if enabled, are issued on the transition to State0.

7.4.3.4 Event Priorities

If simultaneous events occur, the priority is resolved according to Table 7-31. Lower priority events are suppressed. It is thus possible to miss a lower priority event if it occurs simultaneously with an event of a higher priority. The event priorities dictate that in the case of simultaneous matches, the match on the higher comparator channel number (3,1,0) has priority.

If a write access to DBGC1 with the ARM bit position set occurs simultaneously to a hardware disarm from an internal event, then the ARM bit is cleared due to the hardware disarm.

– Enable the external oscillator (OSCE bit).
– Wait for oscillator to start up (UPOSC=1).
– Select the Oscillator Clock (OSCCLK) as source of the Bus Clock (PLLSEL=0).

— The PLLCLK is on and used to qualify the external oscillator clock.

9.1.2.2 Wait Mode

For S12CPMU_UHV Wait Mode is the same as Run Mode.

9.1.2.3 Stop Mode

Stop mode can be entered by executing the CPU STOP instruction. See device level specification for more details.

The voltage regulator is in Reduced Performance Mode (RPM).

NOTE

The voltage regulator output voltage may degrade to a lower value than in Full Performance Mode (FPM), additionally the current sourcing capability is substantially reduced (see also Appendix for VREG electrical parameters). Only clock source ACLK is available and the Power On Reset (POR) circuitry is functional. The Low Voltage Interrupt (LVI) and Low Voltage Reset (LVR) are disabled.

The API is available.

The Phase Locked Loop (PLL) is off.

The Internal Reference Clock (IRC1M) is off.

Core Clock and Bus Clock are stopped.

Depending on the setting of the PSTP and the OSCE bit, Stop Mode can be differentiated between Full Stop Mode (PSTP = 0 or OSCE=0) and Pseudo Stop Mode (PSTP = 1 and OSCE=1). In addition, the behavior of the COP in each mode will change based on the clocking method selected by COPOSCSEL[1:0].

- **Full Stop Mode (PSTP = 0 or OSCE=0)**
  External oscillator (XOSCLCP) is disabled.
  — If COPOSCSEL1=0:
  The COP and RTI counters halt during Full Stop Mode.
  After wake-up from Full Stop Mode the Core Clock and Bus Clock are running on PLLCLK (PLLSEL=1). COP and RTI are running on IRCCLK (COPOSCSEL0=0, RTIOSCSEL=0).
  — If COPOSCSEL1=1:
  The clock for the COP is derived from ACLK (trimmable internal RC-Oscillator clock). During Full Stop Mode the ACLK for the COP can be stopped (COP static) or running (COP active) depending on the setting of bit CSAD. When bit CSAD is set the ACLK clock source for the

Table 9-16. COP Watchdog Rates if COPOSCSEL1=1.

CR2	CR1	CR0	COPCLK Cycles to time-out (COPCLK is ACLK divided by 2)
0	0	0	COP disabled
0	0	1	2^{7}
0	1	0	2^{9}
0	1	1	2^{11}
1	0	0	2^{13}
1	0	1	2^{15}
1	1	0	2^{16}
1	1	1	2^{17}

Table 9-28. TC trimming of the frequency of the IRC1M at ambient temperature

TCTRIM[4:0]	IRC1M Indicative relative TC variation	IRC1M indicative frequency drift for relative TC variation
00000	0 (nominal TC of the IRC)	0%
00001	-0.27%	-0.5%
00010	-0.54%	-0.9%
00011	-0.81%	-1.3%
00100	-1.08%	-1.7%
00101	-1.35%	-2.0%
00110	-1.63%	-2.2%
00111	-1.9%	-2.5%
01000	-2.20%	-3.0%
01001	-2.47%	-3.4%
01010	-2.77%	-3.9%
01011	-3.04	-4.3%
01100	-3.33%	-4.7%
01101	-3.6%	-5.1%
01110	-3.91%	-5.6%
01111	-4.18%	-5.9%
10000	0 (nominal TC of the IRC)	0%
10001	+0.27%	+0.5%
10010	+0.54%	+0.9%
10011	+0.81%	+1.3%
10100	+1.07%	+1.7%
10101	+1.34%	+2.0%
10110	+1.59%	+2.2%
10111	+1.86%	+2.5%
11000	+2.11%	+3.0%
11001	+2.38%	+3.4%
11010	+2.62%	+3.9%
11011	+2.89%	+4.3%
11100	+3.12%	+4.7%
11101	+3.39%	+5.1%
11110	+3.62%	+5.6%
11111	+3.89%	+5.9%

NOTE

Since the IRC1M frequency is not a linear function of the temperature, but more like a parabola, the above relative variation is only an indication and should be considered with care.

9.5.4 PLL Clock Monitor Reset

In case of loss of PLL clock oscillation or the PLL clock frequency is below the failure assert frequency f_{PMFA} (see device electrical characteristics for values), the S12CPMU_UHV generates a PLL Clock Monitor Reset. In Full Stop Mode the PLL and the PLL clock monitor are disabled.

9.5.5 Computer Operating Properly Watchdog (COP) Reset

The COP (free running watchdog timer) enables the user to check that a program is running and sequencing properly. When the COP is being used, software is responsible for keeping the COP from timing out. If the COP times out it is an indication that the software is no longer being executed in the intended sequence; thus COP reset is generated.

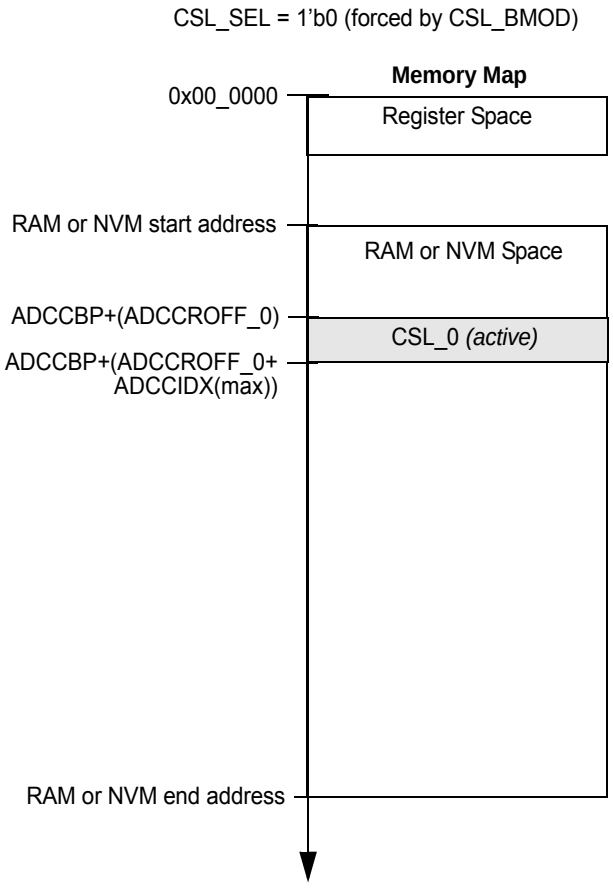
The clock source for the COP is either ACLK, IRCCLK or OSCCLK depending on the setting of the COPOSCSEL0 and COPOSCSEL1 bit.

Depending on the COP configuration there might be a significant latency time until COP is active again after exit from Stop Mode due to clock domain crossing synchronization. This latency time occurs if COP clock source is ACLK and the CSAD bit is set (please refer to CSAD bit description for details)

Table 9-35 gives an overview of the COP condition (run, static) in Stop Mode depending on legal configuration and status bit settings:

Table 9-35. COP condition (run, static) in Stop Mode

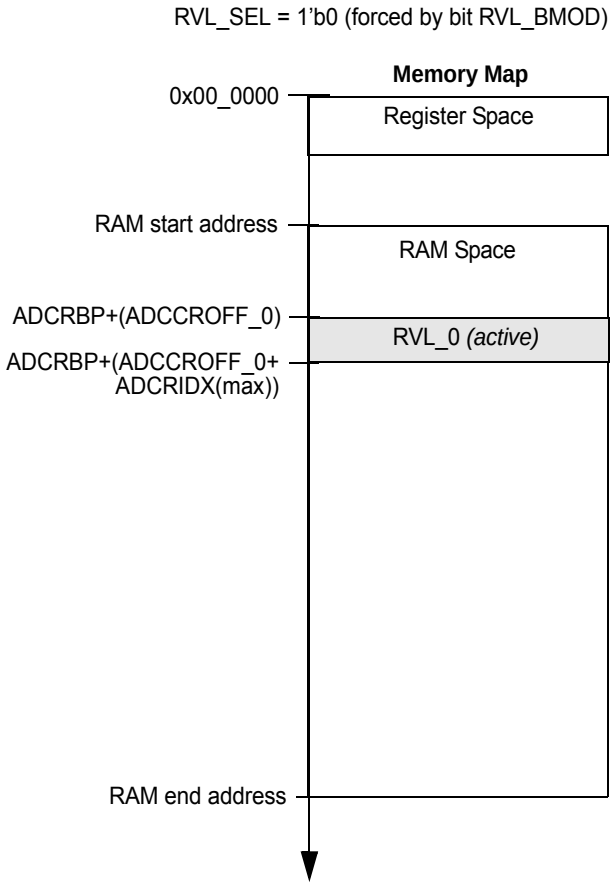
COPOSCSEL1	CSAD	PSTP	PCE	COPOSCSEL0	OSCE	UPOSC	COP counter behavior in Stop Mode (clock source)
1	0	x	x	x	x	x	Run (ACLK)
1	1	x	x	x	x	x	Static (ACLK)
0	x	1	1	1	1	1	Run (OSCCLK)
0	x	1	1	0	0	x	Static (IRCCLK)
0	x	1	1	0	1	x	Static (IRCCLK)
0	x	1	0	0	x	x	Static (IRCCLK)
0	x	1	0	1	1	1	Static (OSCCLK)
0	x	0	1	1	1	1	Static (OSCCLK)
0	x	0	1	0	1	x	Static (IRCCLK)
0	x	0	1	0	0	0	Static (IRCCLK)
0	x	0	0	1	1	1	Satic (OSCCLK)
0	x	0	0	0	1	1	Static (IRCCLK)
0	x	0	0	0	1	0	Static (IRCCLK)
0	x	0	0	0	0	0	Static (IRCCLK)



Note: Address register names in () are not absolute addresses instead they are a sample offset or sample index

Figure 10-32. Command Sequence List Schema in Single Buffer Mode

While the ADC is enabled, one CSL is active (indicated by bit CSL_SEL) and the corresponding list should not be modified anymore. At the same time the alternative CSL can be modified to prepare the ADC for new conversion sequences in CSL double buffered mode. When the ADC is enabled, the command address registers (ADCCBP, ADCCROFF_0/2, ADCCIDX) are read only and register ADCCIDX is under control of the ADC.



Note: Address register names in () are not absolute addresses instead they are a sample offset or sample index

Figure 10-34. Result Value List Schema in Single Buffer Mode

While ADC is enabled, one Result Value List is active (indicated by bit RVL_SEL). The conversion Result Value List can be read anytime. When the ADC is enabled the conversion result address registers (ADCRBP, ADCCROFF_0/1, ADCRIDX) are read only and register ADCRIDX is under control of the ADC.

A conversion result is always stored as 16bit entity in unsigned data representation. Left and right justification inside the entity is selected via the DJM control bit. Unused bits inside an entity are stored zero.

Table 10-33. Conversion Result Justification Overview

Conversion Resolution (SRES[1:0])	Left Justified Result (DJM = 1’b0)	Right Justified Result (DJM = 1’b1)
8 bit	{Result[7:0],8’b00000000}	{8’b00000000,Result[7:0]}
10 bit	{Result[9:0],6’b000000}	{6’b000000,Result[9:0]}
12 bit	{Result[11:0],4’b0000}	{4’b0000,Result[11:0]}

10.6.3.2.6 Conversion flow control in case of conversion sequence control bit overrun scenarios

Restart Request Overrun:
If a legal Restart Request is detected and no Restart Event is in progress, the RSTA bit is set due to the request. The set RSTA bit indicates that a Restart Request was detected and the Restart Event is in process. In case further Restart Requests occur while the RSTA bit is set, this is defined a overrun situation. This scenario is likely to occur when bit STR_SEQA is set or when a Restart Event causes a Sequence Abort Event. The request overrun is captured in a background register that always stores the last detected overrun request. Hence if the overrun situation occurs more than once while a Restart Event is in progress, only the latest overrun request is pending. When the RSTA bit is cleared, the latest overrun request is processed and RSTA is set again one cycle later.

LoadOK Overrun:
Simultaneously at any Restart Request overrun situation the LoadOK input is evaluated and the status is captured in a background register which is alternated anytime a Restart Request Overrun occurs while Load OK Request is asserted. The Load OK background register is cleared as soon as the pending Restart Request gets processed.

Trigger Overrun:
If a Trigger occurs whilst bit TRIG is already set, this is defined as a Trigger overrun situation and causes the ADC to cease conversion at the next conversion boundary and to set bit TRIG_EIF. A overrun is also detected if the Trigger Event occurs automatically generated by hardware in “Trigger Mode” due to a Restart Event and simultaneously a Trigger Event is generated via data bus or internal interface. In this case the ADC ceases operation before conversion begins to sample. In “Trigger Mode” a Restart Request Overrun does not cause a Trigger Overrun (bit TRIG_EIF not set).

Sequence Abort Request Overrun:
If a Sequence Abort Request occurs whilst bit SEQA is already set, this is defined as a Sequence Abort Request Overrun situation and the overrun request is ignored.

2. CPU stop mode

   During stop mode the PGA module is disabled. From the module PGA side no special handling to enter test mode is required. The content of the configuration registers is unchanged.

NOTE

After enabling and after return from CPU stop mode, the PGA module needs a settling time $t_{PGA\_settling}$ to get fully operational.

12.2.3 Block Diagram

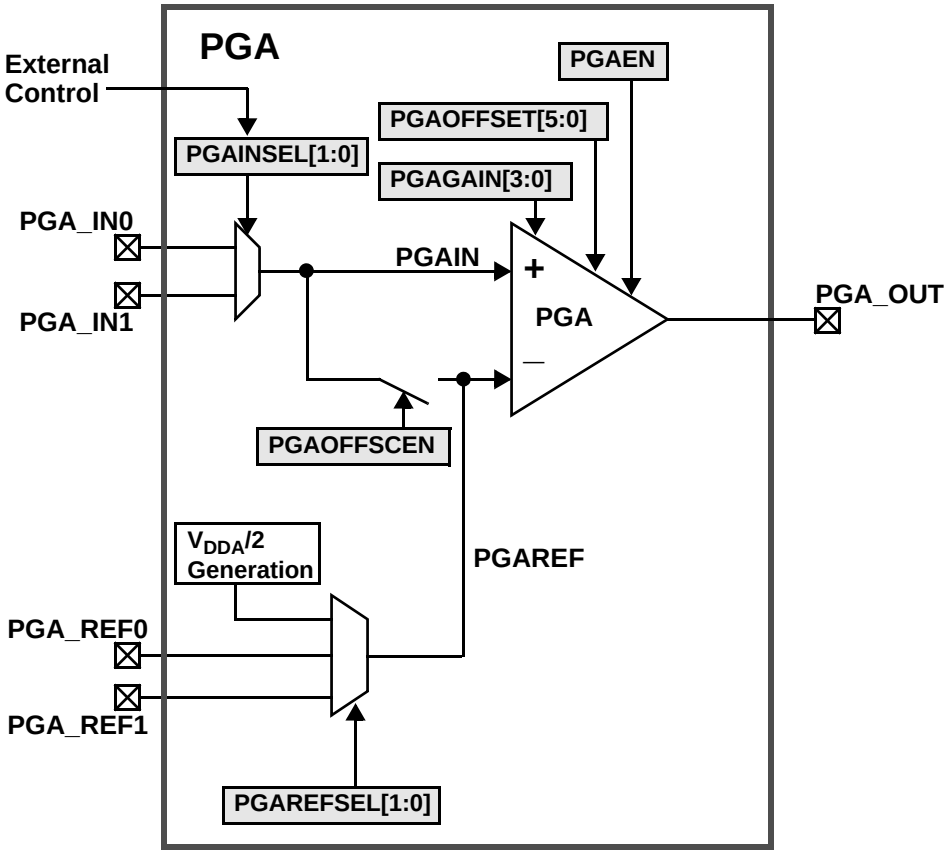


Figure 12-1. PGA Block Diagram

12.2.4 External Signal Description

This section lists the name and description of all external ports.

12.2.5 Amplifier Inputs

For correct operation all amplifier input pins must be within the common voltage input range V_{CM}.

12.2.5.1 PGA_REF0 Pin

This analog pin is used as reference voltage and amplifier minus input voltage if the associated control register bit is set.

12.2.5.2 PGA_REF1 Pin

This analog pin is used as reference voltage and amplifier minus input voltage if the associated control register bit is set.

12.2.5.3 PGA_IN0 Pin

This analog pin is used as amplifier plus input voltage if the associated control register bit is set.

12.2.5.4 PGA_IN1 Pin

This analog pin is used as amplifier plus input voltage if the associated control register bit is set.

12.2.6 PGA_OUT Pin

This analog pin provides the analog amplifier output voltage of the PGA as a function of the gain, offset and the reference voltage.

12.3 Memory Map and Register Definition

This sections provides the detailed information of all registers for the PGA module.

12.3.1 Register Summary

Figure 12-2 shows the summary of all implemented registers inside the PGA module.

NOTE

Register Address = Module Base Address + Address Offset, where the Module Base Address is defined at the MCU level and the Address Offset is defined at the module level.

Address Offset Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 PGAEN	R	0	0	0	0	0		PGAOFFSCEN	PGAEN
	W								
0x0001 PGACNTL	R	0	0	PGAREFSEL[1:0]		0	0	PGAINSEL[1:0]	
	W								

= Unimplemented

Figure 12-2. PGA Register Summary

Table 17-7. PWMPRCLK Field Descriptions

Field	Description
6–4 PCKB[2:0]	**Prescaler Select for Clock B** — Clock B is one of two clock sources which can be used for all channels. These three bits determine the rate of clock B, as shown in Table 17-8.
2–0 PCKA[2:0]	**Prescaler Select for Clock A** — Clock A is one of two clock sources which can be used for all channels. These three bits determine the rate of clock A, as shown in Table 17-8.

Table 17-8. Clock A or Clock B Prescaler Selects

PCKA/B2	PCKA/B1	PCKA/B0	Value of Clock A/B
0	0	0	bus clock
0	0	1	bus clock / 2
0	1	0	bus clock / 4
0	1	1	bus clock / 8
1	0	0	bus clock / 16
1	0	1	bus clock / 32
1	1	0	bus clock / 64
1	1	1	bus clock / 128

17.3.2.5 PWM Center Align Enable Register (PWMCAE)

The PWMCAE register contains eight control bits for the selection of center aligned outputs or left aligned outputs for each PWM channel. If the CAEx bit is set to a one, the corresponding PWM output will be center aligned. If the CAEx bit is cleared, the corresponding PWM output will be left aligned. See Section 17.4.2.5, “Left Aligned Outputs” and Section 17.4.2.6, “Center Aligned Outputs” for a more detailed description of the PWM output modes.

Module Base + 0x0004

	7	6	5	4	3	2	1	0
R W	CAE7	CAE6	CAE5	CAE4	CAE3	CAE2	CAE1	CAE0
Reset	0	0	0	0	0	0	0	0

Figure 17-7. PWM Center Align Enable Register (PWMCAE)

Read: Anytime

Write: Anytime

NOTE

Write these bits only when the corresponding channel is disabled.

21.4 Functional Description

21.4.1 General

The S12LINPHYV2 module implements the physical layer of the LIN interface. This physical layer can be driven by the SCI (Serial Communication Interface) module or directly through the LPDR register.

21.4.2 Slew Rate and LIN Mode Selection

The slew rate can be selected for Electromagnetic Compatibility (EMC) optimized operation at 10.4 kbit/s and 20 kbit/s as well as at fast baud rate (up to 250 kbit/s) for test and programming. The slew rate can be chosen with the bits LPSLR[1:0] in the LIN Slew Rate Mode Register (LPSLRM). The default slew rate corresponds to 20 kbit/s.

The LIN Physical Layer can also be configured to be used for non-LIN applications (for example, to transmit a PWM pulse) by disabling the TxD-dominant timeout (LPDTDIS=1).

Changing the slew rate (LPSLRM Register) during transmission is not allowed in order to avoid unwanted effects. To change the register, the LIN Physical Layer must first be disabled (LPE=0). Once it is updated the LIN Physical Layer can be enabled again.

NOTE

For 20 kbit/s and Fast Mode communication speeds, the corresponding slew rate ***MUST*** be set; otherwise, the communication is not guaranteed (violation of the specified LIN duty cycles). For 10.4 kbit/s, the 20 kbit/s slew rate ***can*** be set but the EMC performance is worse. The up to 250 kbit/s slew rate must be chosen ***ONLY*** for fast mode, not for any of the 10.4 kbit/s or 20 kbit/s LIN compliant communication speeds.

21.4.2.1 10.4 kbit/s and 20 kbit/s

When the slew rate is chosen for 10.4 kbit/s or 20 kbit/s communication, a control loop is activated within the module to make the rise and fall times of the LIN bus independent from VLINSUP and the load on the bus.

21.4.2.2 Fast Mode (not LIN compliant)

Choosing this slew rate allows baud rates up to 250 kbit/s by having much steeper edges (please refer to electricals). As for the 10.4 kbit/s and 20 kbit/s modes, the slope control loop is also engaged. This mode is used for fast communication only, and the LIN electricals are not supported (for example, the LIN duty cycles).

A stronger external pullup resistor might be necessary to sustain communication speeds up to 250 kbit/s. The LIN signal (and therefore the receive LPRxD signal) might not be symmetrical for high baud rates with high loads on the bus.

During the reset sequence, fields DPOPEN and DPS of the DFPROT register are loaded with the contents of the EEPROM protection byte in the Flash configuration field at global address 0xFF_FE0D located in P-Flash memory (see Table 22-3) as indicated by reset condition F in Table 22-24. To change the EEPROM protection that will be loaded during the reset sequence, the P-Flash sector containing the EEPROM protection byte must be unprotected, then the EEPROM protection byte must be programmed. If a double bit fault is detected while reading the P-Flash phrase containing the EEPROM protection byte during the reset sequence, the DPOPEN bit will be cleared and DPS bits will be set to leave the EEPROM memory fully protected.

Trying to alter data in any protected area in the EEPROM memory will result in a protection violation error and the FPVIOL bit will be set in the FSTAT register. Block erase of the EEPROM memory is not possible if any of the EEPROM sectors are protected.

Table 22-23. DFPROT Field Descriptions

Field	Description
7 DPOPEN	**EEPROM Protection Control** 0 Enables EEPROM memory protection from program and erase with protected address range defined by DPS bits 1 Disables EEPROM memory protection from program and erase
6–0 DPS[6:0]	**EEPROM Protection Size** — The DPS bits determine the size of the protected area in the EEPROM memory as shown in Table 22-24 .

Table 22-24. EEPROM Protection Address Range

DPS[6:0]	Global Address Range	Protected Size
0000000	0x10_0000 – 0x10_001F	32 bytes
0000001	0x10_0000 – 0x10_003F	64 bytes
0000010	0x10_0000 – 0x10_005F	96 bytes
0000011	0x10_0000 – 0x10_007F	128 bytes
0000100	0x10_0000 – 0x10_009F	160 bytes
The Protection Size goes on enlarging in step of 32 bytes, for each DPS value increment . .		
0001111	0x10_0000 – 0x10_01FF	512 bytes
0011111	0x10_0000 – 0x10_03FF	1K byte
0111111	0x10_0000 – 0x10_07FF	2K bytes
1111111	0x10_0000 – 0x10_0FFF	4K bytes

The number of DPS bits depends on the size of the implemented EEPROM. The whole implemented EEPROM range can always be protected. Each DPS value increment increases the size of the protected range by 32-bytes. Thus to protect a 1 KB range DPS[4:0] must be set (protected range of 32 x 32 bytes).

Appendix J
PIM Electrical Specifications

J.1 High-Voltage Inputs (HVI) Electrical Characteristics

Table J-1. Static Electrical Characteristics - High Voltage Input Pins - Port L

Characteristics are $5.5V \le V_{SUP} \le 18V$, $-40°C \le T_J \le 175°C$[1] unless otherwise noted. Typical values noted reflect the approximate parameter mean at $T_A = 25°C$[2] under nominal conditions unless otherwise noted.						
Num	**Ratings**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	Digital Input Threshold • $V_{SUP} > 6.5V$, V_{DDX}=5V selected • $5.5V \le V_{SUP} \le 6.5V$, V_{DDX}=5V selected • $V_{SUP} > 5.5V$, V_{DDX}=3.3V selected	V_{TH_HVI}	 2.8 2.0 0.8	 3.5 2.5 1.8	 4.5 3.8 2.7	 V V V
2	Input Hysteresis	V_{HYS_HVI}	_	250	_	mV
3	Pin Input Divider Ratio with external series R_{EXT_HVI} Ratio = V_{HVI} / $V_{Internal(ADC)}$	 $Ratio_{L\_HVI}$ $Ratio_{H\_HVI}$ $Ratio_{12\_HVI}$	 – – –	 2 6 12	 – – –	
4	Analog Input Matching Absolute Error on V_{ADC}[3] • Compared to V_{HVI} / $Ratio_{L\_HVI}$ ($1V < V_{HVI} < 7V$), V_{DDX}=5V selected ($1V < V_{HVI} < 2.8V$), V_{DDX}=3.3V selected • Compared to V_{HVI} / $Ratio_{H\_HVI}$ ($3V < V_{HVI} < 21V$), V_{DDX}=5V selected ($5V < V_{HVI} < 18V$), V_{DDX}=5V selected ($3V < V_{HVI} < 8.4V$), V_{DDX}=3.3V selected • Compared to V_{HVI} / $Ratio_{12\_HVI}$ ($6V < V_{HVI} < 28V$), V_{DDX}=5V selected ($6V < V_{HVI} < 14.8V$), V_{DDX}=3.3V selected • Direct Mode (PTADIRL=1) ($0.5V < V_{HVI} < 3.5V$),V_{DDX}=5V selected ($0.5V < V_{HVI} < 1.4V$),V_{DDX}=3.3V selected	 AIM_{L_HVI} AIM_{H_HVI} AIM_{12_HVI} AIM_{D_HVI} 	 – – – – – – – – –	 ± 2 ± 3 ± 2 ± 2 ± 3 ± 3 ± 3 ± 2 ± 2	 ± 5 ± 7 ± 5 ± 3 ± 7 ± 7 ± 7 ± 5 ± 5	 % % % % % % % % %
5	High Voltage Input Series Resistor **Note:** Always required externally at HVI pins.	R_{EXT_HVI}	–	10	–	kΩ
6	Enable Uncertainty Time	t_{UNC_HVI}	–	1	–	µs
7	Input capacitance	C_{IN_HVI}	–	8	–	pF
8	Current Injection	I_{IC_HVI}	see Footnote[4]			–

1 T_J: Junction Temperature

2 T_A: Ambient Temperature

3 Outside of the given V_{HVI} range the error is significant. The ratio can be changed, if outside of the given range

O.9 0x0480-x04AF PWM0 (continued)

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x048A - 0x048B	RESERVED	R	0	0	0	0	0	0	0	0
		W								
0x048C	PWMCNT0	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x048D	PWMCNT1	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x048E	PWMCNT2	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x048F	PWMCNT3	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x0490	PWMCNT4	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x0491	PWMCNT5	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x0492	PWMCNT6	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x0493	PWMCNT7	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x0494	PWMPER0	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0495	PWMPER1	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0496	PWMPER2	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0497	PWMPER3	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0498	PWMPER4	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0499	PWMPER5	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x049A	PWMPER6	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x049B	PWMPER7	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								

O.9 0x0480-x04AF PWM0 (continued)

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x049C	PWMDTY0	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x049D	PWMDTY1	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x049E	PWMDTY2	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x049F	PWMDTY32	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x04A0	PWMDTY42	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x04A1	PWMDTY52	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x04A2	PWMDTY62	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x04A3	PWMDTY72	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x04A4 - 0x04AF	RESERVED	R W	0	0	0	0	0	0	0	0

O.10 0x0500-x052F PWM1

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0500	PWME	R W	PWME7	PWME6	PWME5	PWME4	PWME3	PWME2	PWME1	PWME0
0x0501	PWMPOL	R W	PPOL7	PPOL6	PPOL5	PPOL4	PPOL3	PPOL2	PPOL1	PPOL0
0x05022	PWMCLK	R W	PCLK7	PCLKL6	PCLK5	PCLK4	PCLK3	PCLK2	PCLK1	PCLK0
0x0503	PWMPRCLK	R W	0	PCKB2	PCKB1	PCKB0	0	PCKA2	PCKA1	PCKA0
0x0504	PWMCAE	R W	CAE7	CAE6	CAE5	CAE4	CAE3	CAE2	CAE1	CAE0
0x0505	PWMCTL	R W	CON67	CON45	CON23	CON01	PSWAI	PFRZ	0	0
0x0506	PWMCLKAB	R W	PCLKAB7	PCLKAB6	PCLKAB5	PCLKAB4	PCLKAB3	PCLKAB2	PCLKAB1	PCLKAB0