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	HCS12Z
Core Size	16-Bit
Speed	32MHz
Connectivity	CANbus, I²C, LINbus, SCI, SPI
Peripherals	DMA, LCD, POR, PWM, WDT
Number of I/O	100
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	5.5V ~ 18V
Data Converters	A/D 8x10b SAR
Oscillator Type	External, Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvhy64f1clq

Port	Pin Name	Pin Function & Priority[1]	I/O	Description	Routing Register	Pin Function after Reset
C	PC7	TXD1	O	TXD of SCI1		GPIO
		PTC[7]	I/O	General-purpose		
	PC6	RXD1	I	RXD of SCI1		
		PTC[6]	I/O	General-purpose		
	PC5	IOC0_7	O	TIM0 channel 7		
		SGA0	O	SGA of SSG0		
		PTC[5]	I/O	General-purpose		
	PC4	IOC0_6	O	TIM0 channel 6		
		SGT0	O	SGT of SSG0		
		PTC[4]	I/O	General-purpose		
	PC3	IOC0_5	O	TIM0 channel 5		
		LPTXD0	I	TXD of LINPHY0	S0L00RR2-0	
		PTC[3]	I/O	General-purpose		
	PC2	IOC0_4	O	TIM0 channel 4		
		LPRXD0	O	RXD of LINPHY0	S0L00RR2-0	
		PTC[2]	I/O	General-purpose		
	PC1	TXCAN0	O	TX of MSCAN0	C0RR	
		PTC[1]	I/O	General-purpose		
	PC0	RXCAN0	I	RX of MSCAN0	C0RR	
		PTC[0]	I/O	General-purpose		
D	PD7	FP15	O	LCD FP15 signal		GPIO
		PTD[7]	I/O	General-purpose		
	PD6	FP14	O	LCD FP14 signal		
		PTD[6]	I/O	General-purpose		
	PD5	FP13	O	LCD FP13 signal		
		PTD[5]	I/O	General-purpose		
	PD4	FP12	O	LCD FP12 signal		
		PTD[4]	I/O	General-purpose		
	PD3	FP11	O	LCD FP11 signal		
		PTD[3]	I/O	General-purpose		
	PD2	FP10	O	LCD FP10 signal		
		PTD[2]	I/O	General-purpose		
	PD1	FP9	O	LCD FP9 signal		
		PTD[1]	I/O	General-purpose		
	PD0	FP8	O	LCD FP8 signal		
		PTD[0]	I/O	General-purpose		

Port	Pin Name	Pin Function & Priority[1]	I/O	Description	Routing Register	Pin Function after Reset
T	PT7	IOC1_7	O	TIM1 channel 7		GPIO
		PTT[7]/KWT[7]	I/O	General-purpose; with interrupt and wakeup		
	PT6	IOC1_6	I/O	TIM1 channel 6		
		ECLK	O	Free running clock output		
		PTT[6]/KWT[6]	I/O	General-purpose; with interrupt and wakeup		
	PT5	IOC1_5	I/O	TIM1 channel 5		
		PTT[5]/KWT[5]	I/O	General-purpose; with interrupt and wakeup		
	PT4	IOC1_4	I/O	TIM1 channel 4		
		PDO	O	DBG profiling data output		
		PTT[4]/KWT[4]	I/O	General-purpose; with interrupt and wakeup		
	PT3	IOC1_3	I/O	TIM1 channel 3		
		PDOCLK	O	DBG profiling clock		
		PTT[3]/KWT[3]	I/O	General-purpose; with interrupt and wakeup		
	PT2	IOC1_2	I/O	TIM1 channel 2		
		DBGEEV	I	DBG external event input		
		PTT[2]/KWT[2]	I/O	General-purpose; with interrupt and wakeup		
	PT1	IOC1_1	I/O	TIM1 channel 1		
		RTC_CAL	I/O	RTC CALCLK output or external 1HZ input		
		PTT[1]/KWT[1]	I/O	General-purpose; with interrupt and wakeup		
	PT0	(IOC1_0)	I/O	TIM1 channel 0	T1IC0RR1-0	
		API_EXTCLK	O	API clock output		
		PTT[0]/KWT[0]	I/O	General-purpose; with interrupt and wakeup		

2.3.2.14 Digital Input Enable Register

Address 0x0299 DIENADL

Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	DIENx7	DIENx6	DIENx5	DIENx4	DIENx3	DIENx2	DIENx1	DIENx0
Reset	0	0	0	0	0	0	0	0

Figure 2-16. Digital Input Enable Register

1. Read: Anytime
   Write: Anytime

Table 2-14. Digital Input Enable Register Field Descriptions

Field	Description
7-0 DIENx	Digital Input Enable — Input buffer control This bit controls the digital input function. If set to 1 the input buffers are enabled and the pin can be used with the digital function. If the pin is used with an analog function this bit shall be cleared to avoid shoot-through current. 1 Associated pin is configured as digital input 0 Associated pin digital input is disabled

Upon detecting the sync request from the host (which is a much longer low time than would ever occur during normal BDC communications), the target:

1. Discards any incomplete command
2. Waits for BKGD to return to a logic high.
3. Delays 16 cycles to allow the host to stop driving the high speed-up pulse.
4. Drives BKGD low for 128 BDCSI clock cycles.
5. Drives a 1-cycle high speed-up pulse to force a fast rise time on BKGD.
6. Removes all drive to the BKGD pin so it reverts to high impedance.
7. Clears the OVRRUN flag (if set).

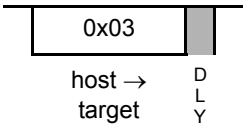
The host measures the low time of this 128-cycle SYNC response pulse and determines the correct speed for subsequent BDC communications. Typically, the host can determine the correct communication speed within a few percent of the actual target speed and the serial protocol can easily tolerate this speed error.

If the SYNC request is detected by the target, any partially executed command is discarded. This is referred to as a soft-reset, equivalent to a timeout in the serial communication. After the SYNC response, the target interprets the next negative edge (issued by the host) as the start of a new BDC command or the start of new SYNC request.

A SYNC command can also be used to abort a pending ACK pulse. This is explained in Section 5.4.8.

5.4.4.2 ACK_DISABLE

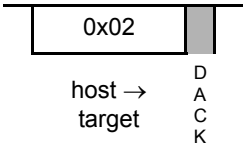
Disable host/target handshake protocol **Always Available**



Disables the serial communication handshake protocol. The subsequent commands, issued after the ACK_DISABLE command, do not execute the hardware handshake protocol. This command is not followed by an ACK pulse.

5.4.4.3 ACK_ENABLE

Enable host/target handshake protocol **Always Available**



Enables the hardware handshake protocol in the serial communication. The hardware handshake is implemented by an acknowledge (ACK) pulse issued by the target MCU in response to a host command. The ACK_ENABLE command is interpreted and executed in the BDC logic without the need to interface

Table 6-24. SSF[2:0] — State Sequence Flag Bit Encoding

SSF[2:0]	Current State
101,110,111	Reserved

6.3.2.12 Debug Comparator A Control Register (DBGACTL)

Address: 0x0110

	7	6	5	4	3	2	1	0
R	0	NDB	INST	0	RW	RWE	reserved	COMPE
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 6-14. Debug Comparator A Control Register

Read: Anytime.

Write: If DBG not armed and PTACT is clear.

Table 6-25. DBGACTL Field Descriptions

Field	Description
6 NDB	**Not Data Bus** — The NDB bit controls whether the match occurs when the data bus matches the comparator register value or when the data bus differs from the register value. This bit is ignored if the INST bit in the same register is set. 0 Match on data bus equivalence to comparator register contents 1 Match on data bus difference to comparator register contents
5 INST	**Instruction Select** — This bit configures the comparator to compare PC or data access addresses. 0 Comparator compares addresses of data accesses 1 Comparator compares PC address
3 RW	**Read/Write Comparator Value Bit** — The RW bit controls whether read or write is used in compare for the associated comparator. The RW bit is ignored if RWE is clear or INST is set. 0 Write cycle is matched 1 Read cycle is matched
2 RWE	**Read/Write Enable Bit** — The RWE bit controls whether read or write comparison is enabled for the associated comparator. This bit is ignored when INST is set. 0 Read/Write is not used in comparison 1 Read/Write is used in comparison
0 COMPE	**Enable Bit** — Determines if comparator is enabled 0 The comparator is not enabled 1 The comparator is enabled

Table 6-26 shows the effect for RWE and RW on the comparison conditions. These bits are ignored if INST is set, because matches based on opcodes reaching the execution stage are data independent.

NOTE

When a CPU indexed jump instruction is executed, the destination address is stored to the trace buffer on instruction completion, indicating the COF has taken place. If an interrupt occurs simultaneously then the next instruction carried out is actually from the interrupt service routine. The instruction at the destination address of the original program flow gets executed after the interrupt service routine.

In the following example an IRQ interrupt occurs during execution of the indexed JMP at address MARK1. The NOP at the destination (SUB_1) is not executed until after the IRQ service routine but the destination address is entered into the trace buffer to indicate that the indexed JMP COF has taken place.

```
         LD      X,#SUB_1
MARK1:   JMP     (0,X)                  ; IRQ interrupt occurs during execution of this
MARK2:   NOP                            ;

SUB_1:   NOP                            ; JMP Destination address TRACE BUFFER ENTRY 1
                                        ; RTI Destination address TRACE BUFFER ENTRY 3
         NOP                            ;
ADDR1:   DBNE    D0,PART5               ; Source address TRACE BUFFER ENTRY 4

IRQ_ISR: LD      D1,#$F0                ; IRQ Vector $FFF2 = TRACE BUFFER ENTRY 2
         ST      D1,VAR_C1
         RTI                            ;
```

The execution flow taking into account the IRQ is as follows

```
         LD      X,#SUB_1
MARK1:   JMP     (0,X)                  ;
IRQ_ISR: LD      D1,#$F0                ;
         ST      D1,VAR_C1
         RTI                            ;
SUB_1:   NOP
         NOP                            ;
ADDR1:   DBNE    D0,PART5               ;
```

The Normal Mode trace buffer format is shown in the following tables. Whilst tracing in Normal or Loop1 modes each array line contains 2 data entries, thus in this case the DBGCNT[0] is incremented after each separate entry. Information byte bits indicate if an entry is a source, destination or vector address.

The external event input can force trace buffer entries independent of COF occurrences, in which case the EEVI bit is set and the PC value of the last instruction is stored to the trace buffer. If the external event coincides with a COF buffer entry a single entry is made with the EEVI bit set.

Normal mode profiling with timestamp is possible when tracing from a single source by setting the STAMP bit in DBGTCRL. This results in a different format (see Table 6-48).

Table 6-47. Normal and Loop1 Mode Trace Buffer Format without Timestamp

Mode	8-Byte Wide Trace Buffer Line							
	7	6	5	4	3	2	1	0

7.3.2.24 Reserved Register CPMUTEST2

NOTE

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in Special Mode can alter the S12CPMU_UHV_V5's functionality.

Module Base + 0x001C

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

= Unimplemented or Reserved

Figure 7-33. Reserved Register CPMUTEST2

Read: Anytime

Write: Only in Special Mode

8.3.2.15 16-Bit Pulse Accumulator Control Register (PACTL)

Module Base + 0x0020

	7	6	5	4	3	2	1	0
R	0	PAEN	PAMOD	PEDGE	CLK1	CLK0	PAOVI	PAI
W								
Reset	0	0	0	0	0	0	0	0

Unimplemented or Reserved

Figure 8-24. 16-Bit Pulse Accumulator Control Register (PACTL)

Read: Any time

Write: Any time

When PAEN is set, the Pulse Accumulator counter is enabled. The Pulse Accumulator counter shares the input pin with IOC7.

Table 8-18. PACTL Field Descriptions

Field	Description
6 PAEN	**Pulse Accumulator System Enable** — PAEN is independent from TEN. With timer disabled, the pulse accumulator can function unless pulse accumulator is disabled. 0 16-Bit Pulse Accumulator system disabled. 1 Pulse Accumulator system enabled.
5 PAMOD	**Pulse Accumulator Mode** — This bit is active only when the Pulse Accumulator is enabled (PAEN = 1). See Table 8-19. 0 Event counter mode. 1 Gated time accumulation mode.
4 PEDGE	**Pulse Accumulator Edge Control** — This bit is active only when the Pulse Accumulator is enabled (PAEN = 1). For PAMOD bit = 0 (event counter mode). See Table 8-19. 0 Falling edges on IOC7 pin cause the count to be increased. 1 Rising edges on IOC7 pin cause the count to be increased. For PAMOD bit = 1 (gated time accumulation mode). 0 IOC7 input pin high enables M (Bus clock) divided by 64 clock to Pulse Accumulator and the trailing falling edge on IOC7 sets the PAIF flag. 1 IOC7 input pin low enables M (Bus clock) divided by 64 clock to Pulse Accumulator and the trailing rising edge on IOC7 sets the PAIF flag.
3:2 CLK[1:0]	**Clock Select Bits —** Refer to Table 8-20.
1 PAOVI	**Pulse Accumulator Overflow Interrupt Enable** 0 Interrupt inhibited. 1 Interrupt requested if PAOVF is set.
0 PAI	**Pulse Accumulator Input Interrupt Enable** 0 Interrupt inhibited. 1 Interrupt requested if PAIF is set.

Table 10-10. Summary of Conversion Flow Control Bit Scenarios

RSTA	TRIG	SEQA	LDOK	Conversion Flow Control Mode	Conversion Flow Control Scenario
0	0	0	0	Both Modes	Valid
0	0	0	1	Both Modes	Can Not Occur
0	0	1	0	Both Modes	Valid[5.]
0	0	1	1	Both Modes	Can Not Occur
0	1	0	0	Both Modes	Valid[2.]
0	1	0	1	Both Modes	Can Not Occur
0	1	1	0	Both Modes	Can Not Occur
0	1	1	1	Both Modes	Can Not Occur
1	0	0	0	Both Modes	Valid[4.]
1	0	0	1	Both Modes	Valid[1. 4.]
1	0	1	0	Both Modes	Valid[3. 4. 5.]
1	0	1	1	Both Modes	Valid[1. 3. 4. 5.]
1	1	0	0	“Restart Mode”	Error flag TRIG_EIF set
				“Trigger Mode”	Valid[2. 4. 6.]
1	1	0	1	“Restart Mode”	Error flag TRIG_EIF set
				“Trigger Mode”	Valid[1. 2. 4. 6.]
1	1	1	0	“Restart Mode”	Error flag TRIG_EIF set
				“Trigger Mode”	Valid[2. 3. 4. 5. 6.]
1	1	1	1	“Restart Mode”	Error flag TRIG_EIF set
				“Trigger Mode”	Valid[(1) (2) (3) (4) (5) (6)]

1. Swap CSL buffer
2. Start conversion sequence
3. Prevent RSTA_EIF and LDOK_EIF
4. Load conversion command from top of CSL
5. Abort any ongoing conversion, conversion sequence and CSL
6. Bit TRIG set automatically in Trigger Mode

For a detailed description of all conversion flow control bit scenarios please see also Section 10.5.3.2.4, “The two conversion flow control Mode Configurations, Section 10.5.3.2.5, “The four ADC conversion flow control bits and Section 10.5.3.2.6, “Conversion flow control in case of conversion sequence control bit overrun scenarios

10.5.3.2.4 The two conversion flow control Mode Configurations

The ADC provides two modes (“Trigger Mode” and “Restart Mode”) which are different in the conversion control flow. The “Restart Mode” provides precise timing control about the sample start point but is more complex from the flow control perspective, while the “Trigger Mode” is more simple from flow control point of view but is less controllable regarding conversion sample start.

Following are the key differences:

In “Trigger Mode” configuration, when conversion flow control bit RSTA gets set the bit TRIG gets set automatically. Hence in “Trigger Mode” the applications should not set the bit TRIG and bit RSTA simultaneously (via data bus or internal interface), because it is a flow control failure and the ADC will cease operation.

In “Trigger Mode” configuration, after the execution of the initial Restart Event the current CSL can be executed and controlled via Trigger Events only. Hence, if the “End Of List” command is reached a restart of conversion flow from top of current CSL does not require to set bit RSTA because returning to the top of current CSL is done automatically. Therefore the current CSL can be executed again after the “End Of List” command type is executed by a Trigger Event only.

In “Restart Mode” configuration, the execution of a CSL is controlled via Trigger Events and Restart Events. After execution of the “End Of List” command the conversion flow must be continued by a Restart Event followed by a Trigger Event and the Trigger Event must not occur before the Restart Event has finished.

For more details and examples regarding flow control and application use cases please see following section and Section 10.8.7, “Conversion flow control application information.

10.5.3.2.5 The four ADC conversion flow control bits

There are four bits to control conversion flow (execution of a CSL and CSL exchange in double buffer mode). Each bit is controllable via the data bus and internal interface depending on the setting of ACC_CFG[1:0] bits (see also Figure 10-2). In the following the conversion control event to control the conversion flow is given with the related internal interface signal and corresponding register bit name together with information regarding:

- Function of the conversion control event
- How to request the event
- When is the event finished
- Mandatory requirements to executed the event

A summary of all event combinations is provided by Table 10-10.

- **Trigger Event**
  Internal Interface Signal: Trigger
  Corresponding Bit Name: TRIG

Module Base + 0x000E — Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	RXERR7	RXERR6	RXERR5	RXERR4	RXERR3	RXERR2	RXERR1	RXERR0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 11-18. MSCAN Receive Error Counter (CANRXERR)

1. Read: Only when in sleep mode (SLPRQ = 1 and SLPAK = 1) or initialization mode (INITRQ = 1 and INITAK = 1)
   Write: Unimplemented

NOTE

Reading this register when in any other mode other than sleep or initialization mode may return an incorrect value. For MCUs with dual CPUs, this may result in a CPU fault condition.

11.3.2.16 MSCAN Transmit Error Counter (CANTXERR)

This register reflects the status of the MSCAN transmit error counter.

Module Base + 0x000F — Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	TXERR7	TXERR6	TXERR5	TXERR4	TXERR3	TXERR2	TXERR1	TXERR0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 11-19. MSCAN Transmit Error Counter (CANTXERR)

1. Read: Only when in sleep mode (SLPRQ = 1 and SLPAK = 1) or initialization mode (INITRQ = 1 and INITAK = 1)
   Write: Unimplemented

NOTE

Reading this register when in any other mode other than sleep or initialization mode, may return an incorrect value. For MCUs with dual CPUs, this may result in a CPU fault condition.

For microcontrollers without a clock and reset generator (CRG), CANCLK is driven from the crystal oscillator (oscillator clock).

A programmable prescaler generates the time quanta (Tq) clock from CANCLK. A time quantum is the atomic unit of time handled by the MSCAN.

Eqn. 11-2

$$\Gamma q = \frac{f_{CANCLK}}{(Prescaler\ \ value}$$

A bit time is subdivided into three segments as described in the Bosch CAN 2.0A/B specification. (see Figure 11-44):

- SYNC_SEG: This segment has a fixed length of one time quantum. Signal edges are expected to happen within this section.
- Time Segment 1: This segment includes the PROP_SEG and the PHASE_SEG1 of the CAN standard. It can be programmed by setting the parameter TSEG1 to consist of 4 to 16 time quanta.
- Time Segment 2: This segment represents the PHASE_SEG2 of the CAN standard. It can be programmed by setting the TSEG2 parameter to be 2 to 8 time quanta long.

Eqn. 11-3

$$Bit\ \ Rate = \frac{f_{Tq}}{(number\ \ of\ \ Time\ \ Quanta)}$$

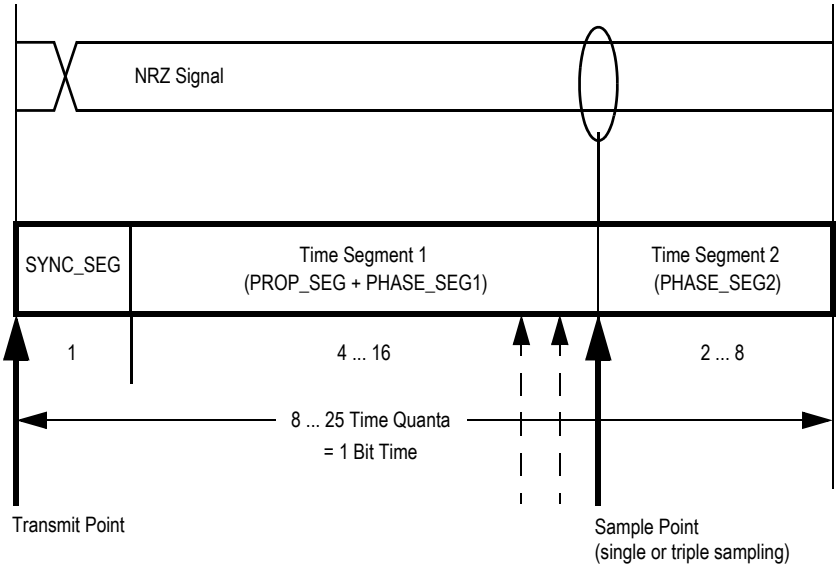


Figure 11-44. Segments within the Bit Time

- Acknowledge bit generation/detection
- Bus busy detection
- General Call Address detection
- Compliant to ten-bit address

15.3.2 Register Descriptions

This section consists of register descriptions. Each description includes a standard register diagram. Details of register bit and field function follow the register diagrams, in bit order.

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0000	LCDCR0	R	LCDEN	0	0	LCLK1	LCLK0	BIAS	DUTY1	DUTY0
		W								
0x0001	LCDCR1	R	0	0	0	0	0	0	LCDSWAI	LCDRSTP
		W								
0x0002	LCDFPENR0	R	FP7EN	FP6EN	FP5EN	FP4EN	FP3EN	FP2EN	FP1EN	FP0EN
		W								
0x0003	LCDFPENR1	R	FP15EN	FP14EN	FP13EN	FP12EN	FP11EN	FP10EN	FP9EN	FP8EN
		W								
0x0004	LCDFPENR2	R	FP23EN	FP22EN	FP21EN	FP20EN	FP19EN	FP18EN	FP17EN	FP16EN
		W								
0x0005	LCDFPENR3	R	FP31EN	FP30EN	FP29EN	FP28EN	FP27EN	FP26EN	FP25EN	FP24EN
		W								
0x0006	LCDFPENR4	R	FP39EN	FP38EN	FP37EN	FP36EN	FP35EN	FP34EN	FP33EN	FP32EN
		W								
0x0007	Reserved	R								
		W								
0x0008	LCDRAM0	R	FP1BP3	FP1BP2	FP1BP1	FP1BP0	FP0BP3	FP0BP2	FP0BP1	FP0BP0
		W								
0x0009	LCDRAM1	R	FP3BP3	FP3BP2	FP3BP1	FP3BP0	FP2BP3	FP2BP2	FP2BP1	FP2BP0
		W								
0x000A	LCDRAM2	R	FP5BP3	FP5BP2	FP5BP1	FP5BP0	FP4BP3	FP4BP2	FP4BP1	FP4BP0
		W								
0x000B	LCDRAM3	R	FP7BP3	FP7BP2	FP7BP1	FP7BP0	FP6BP3	FP6BP2	FP6BP1	FP6BP0
		W								
0x000C	LCDRAM4	R	FP9BP3	FP9BP2	FP9BP1	FP9BP0	FP8BP3	FP8BP2	FP8BP1	FP8BP0
		W								
0x000D	LCDRAM5	R	FP11BP3	FP11BP2	FP11BP1	FP11BP0	FP10BP3	FP10BP2	FP10BP1	FP10BP0
		W								
0x000E	LCDRAM6	R	FP13BP3	FP13BP2	FP13BP1	FP13BP0	FP12BP3	FP12BP2	FP12BP1	FP12BP0
		W								
0x000F	LCDRAM7	R	FP15BP3	FP15BP2	FP15BP1	FP15BP0	FP14BP3	FP14BP2	FP14BP1	FP14BP0
		W								

Figure 15-2. LCD40F4BV3 Register Summary

17.4.3 Operation in Low Power Modes

The SSD block can be configured for lower MCU power consumption in three different ways.

- Stop mode powers down the sigma-delta converter and halts clock to the modulus counter. Exit from Stop enables the sigma-delta converter and the clock to the modulus counter but due to the converter recovery time, the integration result should be ignored.
- Wait mode with SSDWAI bit set powers down the sigma-delta converter and halts the clock to the modulus counter. Exit from Wait enables the sigma-delta converter and clock to the modulus counter but due to the converter recovery time, the integration result should be ignored.
- Clearing SDCPU bit powers down the sigma-delta converter.

17.4.4 Stall Detection Flow

Figure 17-15 shows a flowchart and software setup for stall detection of a stepper motor.

Chapter 20
ECC Generation module (SRAM_ECCV1)

20.1 Introduction

The purpose of ECC logic is to detect and correct as much as possible memory data bit errors. These soft errors can occur randomly during operation, mainly generated by alpha radiation. Soft Error means, that only the information inside the memory cell is corrupt, the memory cell itself is not damaged. A write access with correct data solves the issue. If the ECC algorithm is able to correct the data, then the system can use this corrected data without any issues. If the ECC algorithm is able to detect, but not correct the error, then the system is able to ignore the memory read data to avoid system malfunction.

The ECC value is calculated based on an aligned 2 byte memory data word. The ECC algorithm is able to detect and correct single bit ECC errors. Double bit ECC errors will be detected but the system is not able to correct these errors. This kind of ECC code is called SECDED code. This ECC code requires 6 additional parity bits for each 2 byte data word.

20.1.1 Features

The SRAM_ECC module provides the ECC logic for the system memory based on a SECDED algorithm. Main features of the SRAM_ECC module:

- SECDED ECC code
  - single bit error detection and correction per 2 byte data word
  - double bit error detection per 2 byte data word
- memory initialization function
- byte wide system memory write access
- automatic single bit ECC error correction for read and write accesses
- debug logic to read and write raw use data and ECC values

20.2 Memory Map and Register Definition

This section provides a detailed description of all memory and registers for the SRAM_ECC module.

20.2.1 Register Summary

Figure 20-1 shows the summary of all implemented registers inside the SRAM_ECC module.

21.4.5.4 P-Flash Commands

Table 21-30 summarizes the valid P-Flash commands along with the effects of the commands on the P-Flash block and other resources within the Flash module.

Table 21-30. P-Flash Commands

FCMD	Command	Function on P-Flash Memory
0x01	Erase Verify All Blocks	Verify that all P-Flash (and EEPROM) blocks are erased.
0x02	Erase Verify Block	Verify that a P-Flash block is erased.
0x03	Erase Verify P-Flash Section	Verify that a given number of words starting at the address provided are erased.
0x04	Read Once	Read a dedicated 64 byte field in the nonvolatile information register in P-Flash block that was previously programmed using the Program Once command.
0x06	Program P-Flash	Program a phrase in a P-Flash block.
0x07	Program Once	Program a dedicated 64 byte field in the nonvolatile information register in P-Flash block that is allowed to be programmed only once.
0x08	Erase All Blocks	Erase all P-Flash (and EEPROM) blocks. An erase of all Flash blocks is only possible when the FPLDIS, FPHDIS, and FPOPEN bits in the FPROT register and the DPOPEN bit in the DFPROT register are set prior to launching the command.
0x09	Erase Flash Block	Erase a P-Flash (or EEPROM) block. An erase of the full P-Flash block is only possible when FPLDIS, FPHDIS and FPOPEN bits in the FPROT register are set prior to launching the command.
0x0A	Erase P-Flash Sector	Erase all bytes in a P-Flash sector.
0x0B	Unsecure Flash	Supports a method of releasing MCU security by erasing all P-Flash (and EEPROM) blocks and verifying that all P-Flash (and EEPROM) blocks are erased.
0x0C	Verify Backdoor Access Key	Supports a method of releasing MCU security by verifying a set of security keys.
0x0D	Set User Margin Level	Specifies a user margin read level for all P-Flash blocks.
0x0E	Set Field Margin Level	Specifies a field margin read level for all P-Flash blocks (special modes only).
0x13	Protection Override	Supports a mode to temporarily override Protection configuration (for P-Flash and/or EEPROM) by verifying a key.

21.4.5.5 EEPROM Commands

Table 21-31 summarizes the valid EEPROM commands along with the effects of the commands on the EEPROM block.

0x0004–0x000F Reserved

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0004-0x000F	Reserved	R	0	0	0	0	0	0	0	0
		W								

0x0010–0x001F Interrupt Control module(INT)

0x0010	IVBR	R	IVB_ADDR[15:8]							
		W								
0x0011		R	IVB_ADDR[7:1]							0
		W								
0x0012-0x0015	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0016	INT_XGPRIO	R	0	0	0	0	0	XILVL[2:0]		
		W								
0x0017	INT_CFADDR	R	0	INT_CFADDR[6:3]				0	0	0
		W								
0x0018	INT_CFDATA0	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x0019	INT_CFDATA1	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x001A	INT_CFDATA2	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x001B	INT_CFDATA3	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x001C	INT_CFDATA4	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x001D	INT_CFDATA5	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x001E	INT_CFDATA6	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x001F	INT_CFDATA7	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								

0x0020–0x006F Reserved

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0020-0x006F	Reserved	R	0	0	0	0	0	0	0	0
		W								