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	HCS12X
Core Size	16-Bit
Speed	80MHz
Connectivity	CANbus, EBI/EMI, I²C, IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	119
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 24x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 105°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/mc9s12xdt256vag

Revision History

Date	Revision Level	Description
April, 2005	02.07	New Book
May, 2005	02.08	Minor corrections
May, 2005	02.09	removed ESD Machine Model from electrical characteristics added thermal characteristics added more details to run current measurement configurations VDDA supply voltage range 3.15V - 3.6V fot ATD Operating Characteristics I/O Chararcteristics for alll pins except EXTAL, XTAL .... corrected VREG electrical spec IDD wait max 95mA
May 2005	02.10	Improvements to NVM reliabity spec, added part numbers
July 2005	02.11	Added ROM parts to App.
October 2005	02.12	Single Souce S12XD Fam. Document, New Memory Map Figures,
May 2006	2.13	SPI electricals updated Voltage Regulator electricals updated Added Partnumbers and 1L15Y maskset Updated App. E 6SCI's on 112 pin DT/P512 and 3 SPI's on all D256 parts
June 2006	2.14	Data Sheet covers S12XD/B & A Family Included differnt pull device specification for differnt masksets
July 2006	2.15	Minor Corrections and Improvments
June 2007	2.16	Added 2M42E and 1M84E masksets
July 2007	2.17	Modified Appendix
April 2008	2.18	Better explanation of ATD0/1 for S12XD-Family see page 1305 S12XB256 ATD specification changed see Appendix E.6 added M23S maskset
August 2008	2.19	Corrected XGRAMSIZE of S12XD256 on page 44 Corrected 17.4.2.4 XGATE Memory Map Scheme Corrected 18.4.2.4 XGATE Memory Map Scheme
September 2009	2.20	Corrected Table E-6 , 30K flash memory available for XGATE on B256
October 2009	2.21	Corrected Footnote in Appendix E3 regarding Shared XGATE/CPU area

2.4.3.3.1 Wake-up from Pseudo Stop Mode (PSTP=1)

Wake-up from pseudo stop mode is the same as wake-up from wait mode. There are also four different scenarios for the CRG to restart the MCU from pseudo stop mode:

- External reset
- Clock monitor fail
- COP reset
- Wake-up interrupt

If the MCU gets an external reset or COP reset during pseudo stop mode active, the CRG asynchronously restores all configuration bits in the register space to its default settings and starts the reset generator. After completing the reset sequence processing begins by fetching the normal or COP reset vector. pseudo stop mode is left and the MCU is in run mode again.

If the clock monitor is enabled (CME = 1), the MCU is able to leave pseudo stop mode when loss of oscillator/external clock is detected by a clock monitor fail. If the SCME bit is not asserted the CRG generates a clock monitor fail reset (CMRESET). The CRG's behavior for CMRESET is the same compared to external reset, but another reset vector is fetched after completion of the reset sequence. If the SCME bit is asserted the CRG generates a SCM interrupt if enabled (SCMIE = 1). After generating the interrupt the CRG enters self-clock mode and starts the clock quality checker (Section 2.4.1.4, "Clock Quality Checker"). Then the MCU continues with normal operation. If the SCM interrupt is blocked by SCMIE=0, the SCMIF flag will be asserted but the CRG will not wake-up from pseudo stop mode.

If any other interrupt source (e.g., RTI) triggers exit from pseudo stop mode, the MCU immediately continues with normal operation. Because the PLL has been powered-down during stop mode, the PLLSEL bit is cleared and the MCU runs on OSCCLK after leaving stop mode. The software must set the PLLSEL bit again, in order to switch system and core clocks to the PLLCLK.

Table 2-13 summarizes the outcome of a clock loss while in pseudo stop mode.

4.3.2.6 ATD Control Register 5 (ATDCTL5)

This register selects the type of conversion sequence and the analog input channels sampled. Writes to this register will abort current conversion sequence and start a new conversion sequence. If external trigger is enabled (ETRIGE = 1) an initial write to ATDCTL5 is required to allow starting of a conversion sequence which will then occur on each trigger event. Start of conversion means the beginning of the sampling phase.

	7	6	5	4	3	2	1	0
R W	DJM	DSGN	SCAN	MULT	CD	CC	CB	CA
Reset	0	0	0	0	0	0	0	0

Figure 4-8. ATD Control Register 5 (ATDCTL5)

Read: Anytime

Write: Anytime

Table 4-14. ATDCTL5 Field Descriptions

Field	Description
7 DJM	**Result Register Data Justification** — This bit controls justification of conversion data in the result registers. See Section 4.3.2.16, “ATD Conversion Result Registers (ATDDRx)” for details. 0 Left justified data in the result registers. 1 Right justified data in the result registers.
6 DSGN	**Result Register Data Signed or Unsigned Representation** — This bit selects between signed and unsigned conversion data representation in the result registers. Signed data is represented as 2’s complement. Signed data is not available in right justification. See <st-bold>4.3.2.16 ATD Conversion Result Registers (ATDDRx) for details. 0 Unsigned data representation in the result registers. 1 Signed data representation in the result registers. Table 4-15 summarizes the result data formats available and how they are set up using the control bits. Table 4-16 illustrates the difference between the signed and unsigned, left justified output codes for an input signal range between 0 and 5.12 Volts.
5 SCAN	**Continuous Conversion Sequence Mode** — This bit selects whether conversion sequences are performed continuously or only once. If external trigger is enabled (ETRIGE=1) setting this bit has no effect, that means each trigger event starts a single conversion sequence. 0 Single conversion sequence 1 Continuous conversion sequences (scan mode)
4 MULT	**Multi-Channel Sample Mode** — When MULT is 0, the ATD sequence controller samples only from the specified analog input channel for an entire conversion sequence. The analog channel is selected by channel selection code (control bits CD/CC/CB/CA located in ATDCTL5). When MULT is 1, the ATD sequence controller samples across channels. The number of channels sampled is determined by the sequence length value (S8C, S4C, S2C, S1C). The first analog channel examined is determined by channel selection code (CC, CB, CA control bits); subsequent channels sampled in the sequence are determined by incrementing the channel selection code or wrapping around to AN0 (channel 0. 0 Sample only one channel 1 Sample across several channels

5.3.2.6 ATD Control Register 5 (ATDCTL5)

This register selects the type of conversion sequence and the analog input channels sampled. Writes to this register will abort current conversion sequence and start a new conversion sequence.

	7	6	5	4	3	2	1	0
R	DJM	DSGN	SCAN	MULT	0	CC	CB	CA
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 5-8. ATD Control Register 5 (ATDCTL5)

Read: Anytime

Write: Anytime

Table 5-13. ATDCTL5 Field Descriptions

Field	Description
7 DJM	**Result Register Data Justification** — This bit controls justification of conversion data in the result registers. See Section 5.3.2.13, “ATD Conversion Result Registers (ATDDRx),” for details. 0 Left justified data in the result registers 1 Right justified data in the result registers
6 DSGN	**Result Register Data Signed or Unsigned Representation** — This bit selects between signed and unsigned conversion data representation in the result registers. Signed data is represented as 2’s complement. Signed data is not available in right justification. See Section 5.3.2.13, “ATD Conversion Result Registers (ATDDRx),” for details. 0 Unsigned data representation in the result registers 1 Signed data representation in the result registers Table 5-14 summarizes the result data formats available and how they are set up using the control bits. Table 5-15 illustrates the difference between the signed and unsigned, left justified output codes for an input signal range between 0 and 5.12 Volts.
5 SCAN	**Continuous Conversion Sequence Mode** — This bit selects whether conversion sequences are performed continuously or only once. 0 Single conversion sequence 1 Continuous conversion sequences (scan mode)
4 MULT	**Multi-Channel Sample Mode** — When MULT is 0, the ATD sequence controller samples only from the specified analog input channel for an entire conversion sequence. The analog channel is selected by channel selection code (control bits CC/CB/CA located in ATDCTL5). When MULT is 1, the ATD sequence controller samples across channels. The number of channels sampled is determined by the sequence length value (S8C, S4C, S2C, S1C). The first analog channel examined is determined by channel selection code (CC, CB, CA control bits); subsequent channels sampled in the sequence are determined by incrementing the channel selection code. 0 Sample only one channel 1 Sample across several channels
2–0 CC, CB, CA	**Analog Input Channel Select Code** — These bits select the analog input channel(s) whose signals are sampled and converted to digital codes. Table 5-16 lists the coding used to select the various analog input channels. In the case of single channel scans (MULT = 0), this selection code specified the channel examined. In the case of multi-channel scans (MULT = 1), this selection code represents the first channel to be examined in the conversion sequence. Subsequent channels are determined by incrementing channel selection code; selection codes that reach the maximum value wrap around to the minimum value.

11.4.6.5.2 Fast Data Tolerance

Figure 11-29 shows how much a fast received frame can be misaligned. The fast stop bit ends at RT10 instead of RT16 but is still sampled at RT8, RT9, and RT10.

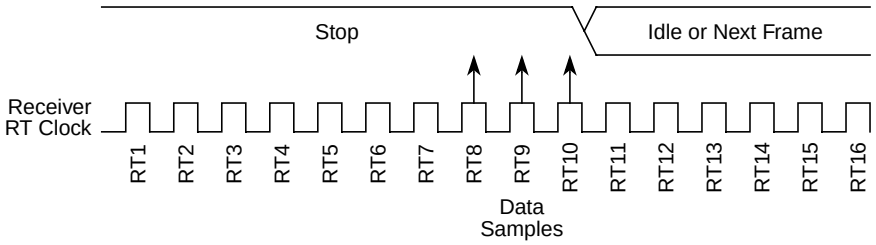


Figure 11-29. Fast Data

For an 8-bit data character, it takes the receiver 9 bit times x 16 RTr cycles + 10 RTr cycles = 154 RTr cycles to finish data sampling of the stop bit.

With the misaligned character shown in Figure 11-29, the receiver counts 154 RTr cycles at the point when the count of the transmitting device is 10 bit times x 16 RTt cycles = 160 RTt cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 8-bit character with no errors is:

((160 – 154) / 160) x 100 = 3.75%

For a 9-bit data character, it takes the receiver 10 bit times x 16 RTr cycles + 10 RTr cycles = 170 RTr cycles to finish data sampling of the stop bit.

With the misaligned character shown in Figure 11-29, the receiver counts 170 RTr cycles at the point when the count of the transmitting device is 11 bit times x 16 RTt cycles = 176 RTt cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 9-bit character with no errors is:

((176 – 170) /176) x 100 = 3.40%

11.4.6.6 Receiver Wakeup

To enable the SCI to ignore transmissions intended only for other receivers in multiple-receiver systems, the receiver can be put into a standby state. Setting the receiver wakeup bit, RWU, in SCI control register 2 (SCICR2) puts the receiver into standby state during which receiver interrupts are disabled.The SCI will still load the receive data into the SCIDRH/L registers, but it will not set the RDRF flag.

The transmitting device can address messages to selected receivers by including addressing information in the initial frame or frames of each message.

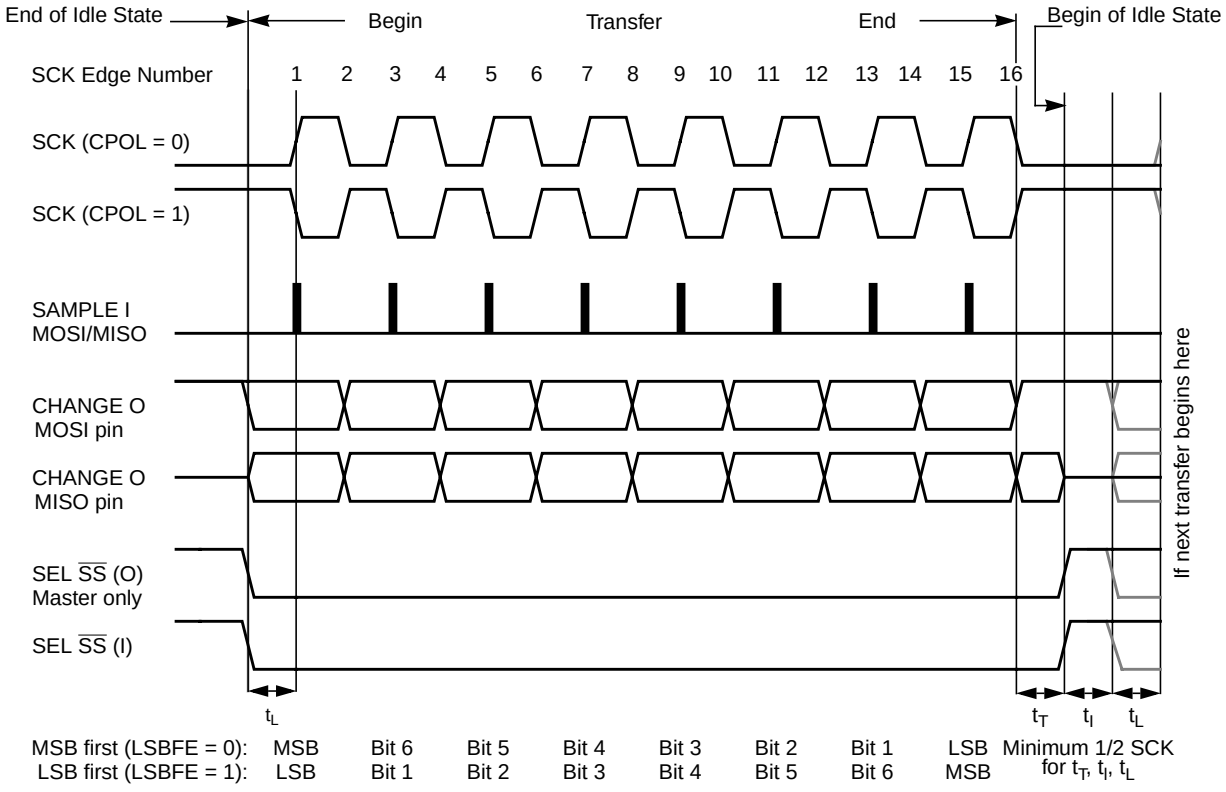
The WAKE bit in SCI control register 1 (SCICR1) determines how the SCI is brought out of the standby state to process an incoming message. The WAKE bit enables either idle line wakeup or address mark wakeup.

Data reception is double buffered. Data is shifted serially into the SPI shift register during the transfer and is transferred to the parallel SPI data register after the last bit is shifted in.

After the 16th (last) SCK edge:

- Data that was previously in the master SPI data register should now be in the slave data register and the data that was in the slave data register should be in the master.
- The SPIF flag in the SPI status register is set, indicating that the transfer is complete.

Figure 12-11 is a timing diagram of an SPI transfer where CPHA = 0. SCK waveforms are shown for CPOL = 0 and CPOL = 1. The diagram may be interpreted as a master or slave timing diagram because the SCK, MISO, and MOSI pins are connected directly between the master and the slave. The MISO signal is the output from the slave and the MOSI signal is the output from the master. The $\overline{SS}$ pin of the master must be either high or reconfigured as a general-purpose output not affecting the SPI.



t_L = Minimum leading time before the first SCK edge
t_T = Minimum trailing time after the last SCK edge
t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time)
t_L, t_T, and t_I are guaranteed for the master mode and required for the slave mode.

Figure 12-11. SPI Clock Format 0 (CPHA = 0)

15.4.5 BDM Command Structure

Hardware and firmware BDM commands start with an 8-bit opcode followed by a 16-bit address and/or a 16-bit data word depending on the command. All the read commands return 16 bits of data despite the byte or word implication in the command name.

8-bit reads return 16-bits of data, of which, only one byte will contain valid data. If reading an even address, the valid data will appear in the MSB. If reading an odd address, the valid data will appear in the LSB.

16-bit misaligned reads and writes are generally not allowed. If attempted by BDM hardware command, the BDM will ignore the least significant bit of the address and will assume an even address from the remaining bits.

The following cycle count information is only valid when the external wait function is not used (see wait bit of EBI sub-block). During an external wait the BDM can not steal a cycle. Hence be careful with the external wait function if the BDM serial interface is much faster than the bus, because of the BDM soft-reset after time-out (see Section 15.4.11, “Serial Communication Time Out”).

For hardware data read commands, the external host must wait at least 150 bus 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 BDM shift register, ready to be shifted out. For hardware write commands, the external host must wait 150 bus clock cycles after sending the data to be written before attempting to send a new command. This is to avoid disturbing the BDM shift register before the write has been completed. The 150 bus clock cycle delay in both cases includes the maximum 128 cycle delay that can be incurred as the BDM waits for a free cycle before stealing a cycle.

For firmware read commands, the external host should wait at least 48 bus clock cycles after sending the command opcode and before attempting to obtain the read data. This includes the potential of extra cycles when the access is external and stretched (+1 to maximum +7 cycles) or to registers of the PRU (port replacement unit) in emulation mode. The 48 cycle wait allows enough time for the requested data to be made available in the BDM shift register, ready to be shifted out.

NOTE

This timing has increased from previous BDM modules due to the new capability in which the BDM serial interface can potentially run faster than the bus. On previous BDM modules this extra time could be hidden within the serial time.

For firmware write commands, the external host must wait 36 bus clock cycles after sending the data to be written before attempting to send a new command. This is to avoid disturbing the BDM shift register before the write has been completed.

20.3.2.8.1 Debug Comparator Control Register (DBGXCTL)

The contents of this register bits 7 and 6 differ depending upon which comparator registers are visible in the 8-byte window of the DBG module register address map.

Address: 0x0028

	7	6	5	4	3	2	1	0
R	0	NDB	TAG	BRK	RW	RWE	SRC	COMPE
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 20-13. Debug Comparator Control Register (Comparators A and C)

Address: 0x0028

	7	6	5	4	3	2	1	0
R	SZE	SZ	TAG	BRK	RW	RWE	SRC	COMPE
W								
Reset	0	0	0	0	0	0	0	0

Figure 20-14. Debug Comparator Control Register (Comparators B and D)

Read: Anytime

Write: Anytime when S12XDBG not armed.

Table 20-27. DBGXCTL Field Descriptions

Field	Description
7 SZE (Comparators B nd D)	**Size Comparator Enable Bit** — The SZE bit controls whether access size comparison is enabled for the associated comparator. This bit is ignored if the TAG bit in the same register is set. 0 Word/Byte access size is not used in comparison 1 Word/Byte access size is used in comparison
6 NDB (Comparators A and C	**Not Data Bus Compare** — 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. Furthermore data bus bits can be individually masked using the comparator data mask registers. This bit is only available for comparators A and C. This bit is ignored if the TAG bit in the same register is set. This bit position has an SZ functionality for comparators B and D. 0 Match on data bus equivalence to comparator register contents 1 Match on data bus difference to comparator register contents
6 SZ (Comparators B and D)	**Size Comparator Value Bit** — The SZ bit selects either word or byte access size in comparison for the associated comparator. This bit is ignored if the SZE bit is cleared or if the TAG bit in the same register is set. This bit position has NDB functionality for comparators A and C 0 Word access size will be compared 1 Byte access size will be compared
5 TAG	**Tag Select** — This bit controls whether the comparator match will cause a trigger or tag the opcode at the matched address. Tagged opcodes trigger only if they reach the execution stage of the instruction queue. 0 Trigger immediately on match 1 On match, tag the opcode. If the opcode is about to be executed a trigger is generated

22.3.2.54 Port J Data Register (PTJ)

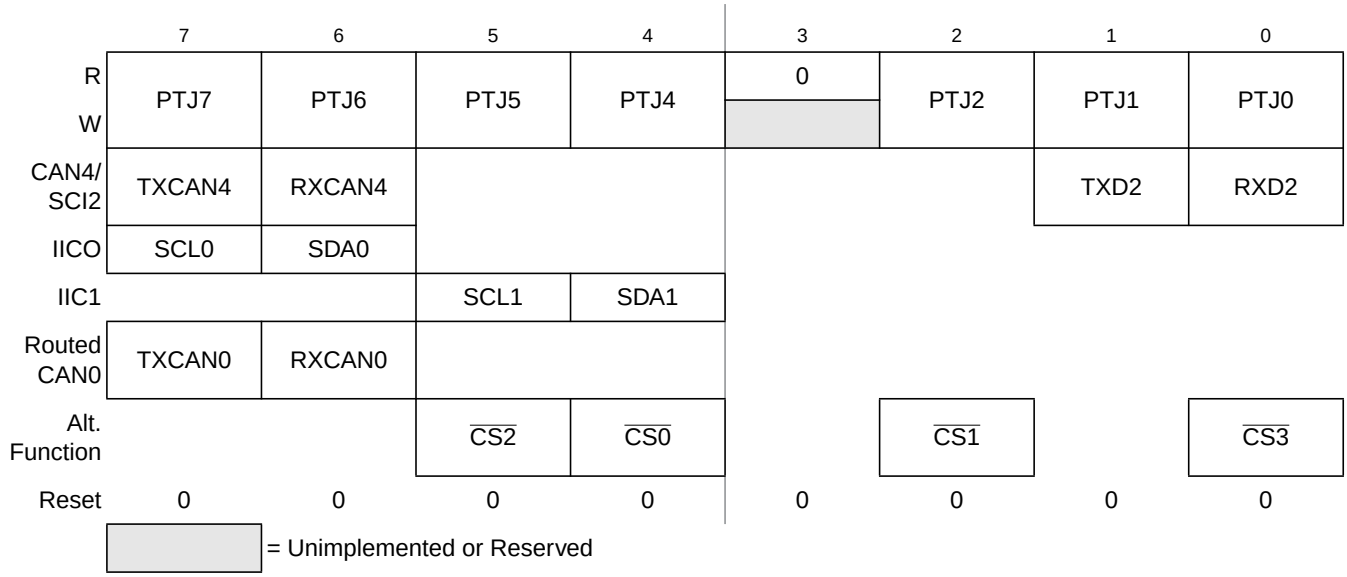


Figure 22-56. Port J Data Register (PTJ)

Read: Anytime.

Write: Anytime.

Port J pins 7–4 and 2–0 are associated with the CAN4, SCI2, IIC0 and IIC1, the routed CAN0 modules and chip select signals ($\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$). These pins can be used as general purpose I/O when not used with any of the peripherals.

If the data direction bits of the associated I/O pins are set to logic level “1”, a read returns the value of the port register, otherwise the buffered pin input state is read.

Table 22-51. PTJ Field Descriptions

Field	Description
7–6 PJ[7:0]	The CAN4 function (TXCAN4 and RXCAN4) takes precedence over the IIC0, the routed CAN0 and the general purpose I/O function if the CAN4 module is enabled. The IIC0 function (SCL0 and SDA0) takes precedence over the routed CAN0 and the general purpose I/O function if the IIC0 is enabled. If the IIC0 module takes precedence the SDA0 and SCL0 outputs are configured as open drain outputs. *Refer to IIC section for details.* The routed CAN0 function (TXCAN0 and RXCAN0) takes precedence over the general purpose I/O function if the routed CAN0 module is enabled. *Refer to MSCAN section for details.*
5-4 PJ[5:4]	The IIC1 function (SCL1 and SDA1) takes precedence over the chip select ($\overline{CS0}$, $\overline{CS2}$) and general purpose I/O function if the IIC1 is enabled. The chip selects ($\overline{CS0}$, $\overline{CS2}$) take precedence over the general purpose I/O. If the IIC1 module takes precedence the SDA1 and SCL1 outputs are configured as open drain outputs. *Refer to IIC section for details.*
2 PJ2	The chip select function ($\overline{CS1}$) takes precedence over the general purpose I/O.

Table 23-1. Pin Functions and Priorities (Sheet 4 of 7)

Port	Pin Name	Pin Function and Priority	I/O	Description	Pin Function after Reset
M	PM7	TXCAN4	O	MSCAN4 transmit pin	GPIO
		GPIO	I/O	General-purpose I/O	
	PM6	RXCAN4	I	MSCAN4 receive pin	
		GPIO	I/O	General-purpose I/O	
	PM5	TXCAN2	O	MSCAN2 transmit pin	
		TXCAN0	O	MSCAN0 transmit pin	
		TXCAN4	O	MSCAN4 transmit pin	
		SCK0	I/O	Serial Peripheral Interface 0 serial clock pin *If CAN0 is routed to PM[3:2] the SPI0 can still be used in bidirectional master mode.*	
		GPIO	I/O	General-purpose I/O	
	PM4	RXCAN2	I	MSCAN2 receive pin	
		RXCAN0	I	MSCAN0 receive pin	
		RXCAN4	I	MSCAN4 receive pin	
		MOSI0	I/O	Serial Peripheral Interface 0 master out/slave in pin *If CAN0 is routed to PM[3:2] the SPI0 can still be used in bidirectional master mode.*	
		GPIO	I/O	General-purpose I/O	
	PM3	TXCAN1	O	MSCAN1 transmit pin	
		TXCAN0	O	MSCAN0 transmit pin	
		$\overline{SS0}$	I/O	Serial Peripheral Interface 0 slave select output in master mode, input for slave mode or master mode.	
		GPIO	I/O	General-purpose I/O	
	PM2	RXCAN1	I	MSCAN1 receive pin	
		RXCAN0	I	MSCAN0 receive pin	
		MISO0	I/O	Serial Peripheral Interface 0 master in/slave out pin	
		GPIO	I/O	General-purpose I/O	
	PM1	TXCAN0	O	MSCAN0 transmit pin	
		GPIO	I/O	General-purpose I/O	
	PM0	RXCAN0	I	MSCAN0 receive pin	
		GPIO	I/O	General-purpose I/O	

23.0.5.37 Module Routing Register (MODRR)

	7	6	5	4	3	2	1	0
R	0	MODRR6	MODRR5	MODRR4	MODRR3	MODRR2	MODRR1	MODRR0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 23-39. Module Routing Register (MODRR)

Read: Anytime.

Write: Anytime.

This register configures the re-routing of CAN0, CAN4, SPI0, SPI1, and SPI2 on alternative ports.

Table 23-38. Module Routing Summary

Module	MODRR							Related Pins			
	6	5	4	3	2	1	0	RXCAN	TXCAN		
CAN0	x	x	x	x	x	0	0	PM0	PM1		
	x	x	x	x	x	0	1	PM2	PM3		
	x	x	x	x	x	1	0	PM4	PM5		
	x	x	x	x	x	1	1	PJ6	PJ7		
CAN4	x	x	x	0	0	x	x	PJ6	PJ7		
	x	x	x	0	1	x	x	PM4	PM5		
	x	x	x	1	0	x	x	PM6	PM7		
	x	x	x	1	1	x	x	Reserved			
								MISO	MOSI	SCK	$\overline{SS}$
SPI0	x	x	0	x	x	x	x	PS4	PS5	PS6	PS7
	x	x	1	x	x	x	x	PM2	PM4	PM5	PM3
SPI1	x	0	x	x	x	x	x	PP0	PP1	PP2	PP3
	x	1	x	x	x	x	x	PH0	PH1	PH2	PH3
SPI2	0	x	x	x	x	x	x	PP4	PP5	PP7	PP6
	1	x	x	x	x	x	x	PH4	PH5	PH6	PH7

24.0.5.65 Port AD1 Pull Up Enable Register 1 (PER1AD1)

	7	6	5	4	3	2	1	0
R W	PER1AD17	PER1AD16	PER1AD15	PER1AD14	PER1AD13	PER1AD12	PER1AD11	PER1AD10
Reset	0	0	0	0	0	0	0	0

Figure 24-67. Port AD1 Pull Up Enable Register 1 (PER1AD1)

Read: Anytime.

Write: Anytime.

This register activates a pull-up device on the respective PAD[7:0] pin if the port is used as input. This bit has no effect if the port is used as output. Out of reset no pull-up device is enabled.

Table 24-59. PER1AD1 Field Descriptions

Field	Description
7–0 PER1AD1[7:0]	**Pull Device Enable Port AD1 Register 1** 0 Pull-up device is disabled. 1 Pull-up device is enabled.

Functional Description

Each pin except PE0, PE1, and BKGD can act as general purpose I/O. In addition each pin can act as an output from the external bus interface module or a peripheral module or an input to the external bus interface module or a peripheral module.

A set of configuration registers is common to all ports with exceptions in the expanded bus interface and ATD ports (Table 24-60). All registers can be written at any time; however a specific configuration might not become active.

Example: Selecting a pull-up device

This device does not become active while the port is used as a push-pull output.

Table 24-60. Register Availability per Port[1]

Port	Data	Data Direction	Input	Reduced Drive	Pull Enable	Polarity Select	Wired-OR Mode	Interrupt Enable	Interrupt Flag
A	yes	yes	—	yes	yes	—	—	—	—
B	yes	yes	—			—	—	—	—
E	yes	yes	—			—	—	—	—
K	yes	yes	—			—	—	—	—
T	yes	yes	yes	yes	yes	—	—	—	—
S	yes	yes	yes	yes	yes	yes	yes	—	—
M	yes	yes	yes	yes	yes	yes	yes	—	—
P	yes	yes	yes	yes	yes	yes	—	yes	yes
H	yes	yes	yes	yes	yes	yes	—	yes	yes

Table 24-60. Register Availability per Port[1]

Port	Data	Data Direction	Input	Reduced Drive	Pull Enable	Polarity Select	Wired-OR Mode	Interrupt Enable	Interrupt Flag
J	yes	yes	yes	yes	yes	yes	—	yes	yes
AD1	yes	yes	—	yes	yes	—	—	—	—

1. Each cell represents one register with individual configuration bits

24.0.6 Registers

24.0.6.1 Data Register

This register holds the value driven out to the pin if the pin is used as a general purpose I/O.

Writing to this register has only an effect on the pin if the pin is used as general purpose output. When reading this address, the buffered state of the pin is returned if the associated data direction register bit is set to “0”.

If the data direction register bits are set to logic level “1”, the contents of the data register is returned. This is independent of any other configuration (Figure 24-68).

24.0.6.2 Input Register

This is a read-only register and always returns the buffered state of the pin (Figure 24-68).

24.0.6.3 Data Direction Register

This register defines whether the pin is used as an input or an output.

If a peripheral module controls the pin the contents of the data direction register is ignored (Figure 24-68).

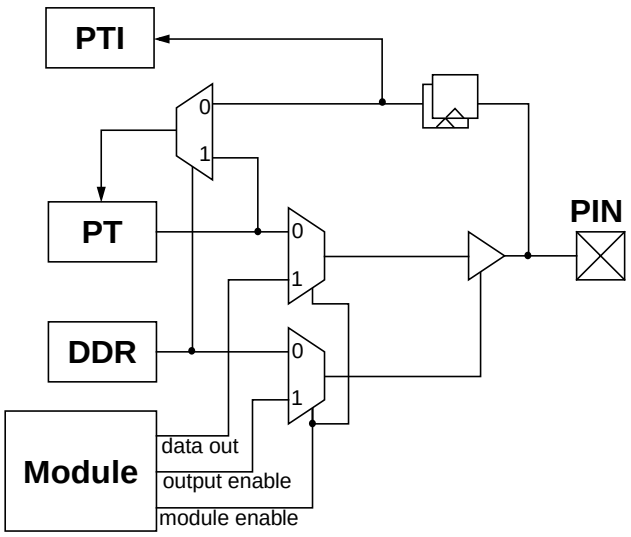


Figure 24-68. Illustration of I/O Pin Functionality

25.5 Operating Modes

25.5.1 Wait Mode

If a command is active (CCIF = 0) when the MCU enters the wait mode, the active command and any buffered command will be completed.

The EEPROM module can recover the MCU from wait mode if the CBEIF and CCIF interrupts are enabled (see Section 25.8, “Interrupts”).

25.5.2 Stop Mode

If a command is active (CCIF = 0) when the MCU enters the stop mode, the operation will be aborted and, if the operation is program, sector erase, mass erase, or sector modify, the EEPROM array data being programmed or erased may be corrupted and the CCIF and ACCERR flags will be set. If active, the high voltage circuitry to the EEPROM memory will immediately be switched off when entering stop mode. Upon exit from stop mode, the CBEIF flag is set and any buffered command will not be launched. The ACCERR flag must be cleared before starting a command write sequence (see Section 25.4.1.2, “Command Write Sequence”).

NOTE

As active commands are immediately aborted when the MCU enters stop mode, it is strongly recommended that the user does not use the STOP instruction during program, sector erase, mass erase, or sector modify operations.

25.5.3 Background Debug Mode

In background debug mode (BDM), the EPROT register is writable. If the MCU is unsecured, then all EEPROM commands listed in Table 25-9 can be executed. If the MCU is secured and is in special single chip mode, the only command available to execute is mass erase.

25.6 EEPROM Module Security

The EEPROM module does not provide any security information to the MCU. After each reset, the security state of the MCU is a function of information provided by the Flash module (see the specific FTX Block Guide).

If the PVIOL flag is set in the FSTAT register, the user must clear the PVIOL flag before starting another command write sequence (see Section 27.3.2.6, “Flash Status Register (FSTAT)”).

27.5 Operating Modes

27.5.1 Wait Mode

If a command is active (CCIF = 0) when the MCU enters wait mode, the active command and any buffered command will be completed.

The Flash module can recover the MCU from wait mode if the CBEIF and CCIF interrupts are enabled (see Section 27.8, “Interrupts”).

27.5.2 Stop Mode

If a command is active (CCIF = 0) when the MCU enters stop mode, the operation will be aborted and, if the operation is program or erase, the Flash array data being programmed or erased may be corrupted and the CCIF and ACCERR flags will be set. If active, the high voltage circuitry to the Flash memory will immediately be switched off when entering stop mode. Upon exit from stop mode, the CBEIF flag is set and any buffered command will not be launched. The ACCERR flag must be cleared before starting a command write sequence (see Section 27.4.1.2, “Command Write Sequence”).

NOTE

As active commands are immediately aborted when the MCU enters stop mode, it is strongly recommended that the user does not use the STOP instruction during program or erase operations.

27.5.3 Background Debug Mode

In background debug mode (BDM), the FPROT register is writable. If the MCU is unsecured, then all Flash commands listed in Table 27-20 can be executed. If the MCU is secured and is in special single chip mode, only mass erase can be executed.

27.6 Flash Module Security

The Flash module provides the necessary security information to the MCU. After each reset, the Flash module determines the security state of the MCU as defined in Section 27.3.2.2, “Flash Security Register (FSEC)”.

The contents of the Flash security byte at 0x7F_FF0F in the Flash Configuration Field must be changed directly by programming 0x7F_FF0F when the MCU is unsecured and the higher address sector is unprotected. If the Flash security byte is left in a secured state, any reset will cause the MCU to initialize to a secure operating mode.

$$f_C < \frac{2 \cdot \zeta \cdot f_{ref}}{\pi \cdot \left(\zeta + \sqrt{1 + \zeta^2}\right)} \cdot \frac{1}{10} \rightarrow f_C < \frac{f_{ref}}{4 \cdot 10}; (\zeta = 0.9)$$

$f_C < 100kHz$

And finally the frequency relationship is defined as

$$n = \frac{f_{VCO}}{f_{ref}} = 2 \cdot (synr + 1) \quad = 20$$

With the above values the resistance can be calculated. The example is shown for a loop bandwidth $f_C = 20$ kHz:

$$R = \frac{2 \cdot \pi \cdot n \cdot f_C}{K_\Phi} = \frac{2 \cdot \pi \cdot 20 \cdot 20kHz}{(539.1Hz)/\Omega} = 4.7k\Omega$$

The capacitance C_s can now be calculated as:

$$C_s = \frac{2 \cdot \zeta^2}{\pi \cdot f_C \cdot R} = \frac{0.516}{f_C \cdot R}; (\zeta = 0.9) \quad = 5.5nF = \sim 4.7nF$$

The capacitance C_p should be chosen in the range of:

$$\frac{C_s}{20} \leq C_p \leq \frac{C_s}{10} \quad C_P = 470pF$$

A.5.3.2 Jitter Information

The basic functionality of the PLL is shown in Figure A-3. With each transition of the clock f_{cmp}, the deviation from the reference clock f_{ref} is measured and input voltage to the VCO is adjusted accordingly.The adjustment is done continuously with no abrupt changes in the clock output frequency. Noise, voltage, temperature and other factors cause slight variations in the control loop resulting in a clock jitter. This jitter affects the real minimum and maximum clock periods as illustrated in Figure A-4.

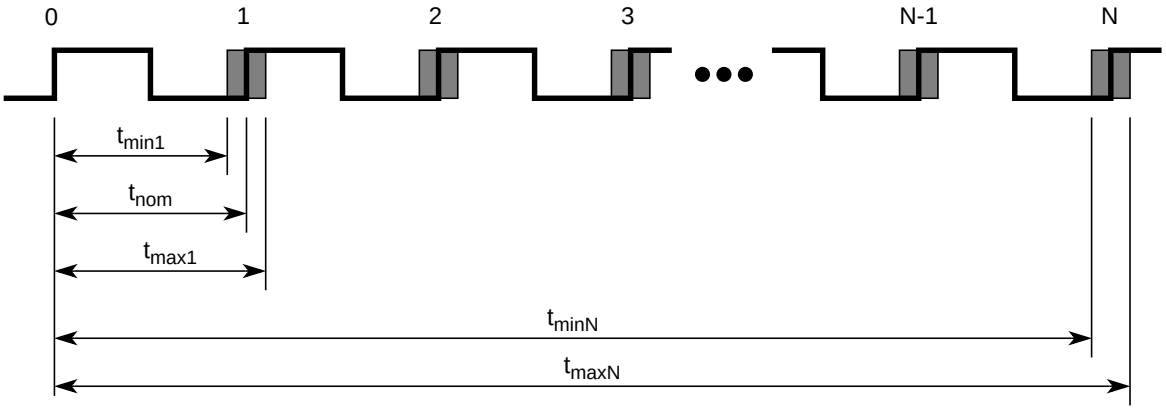


Figure A-4. Jitter Definitions

A.8.2 Normal Expanded Mode (External Wait Feature Enabled)

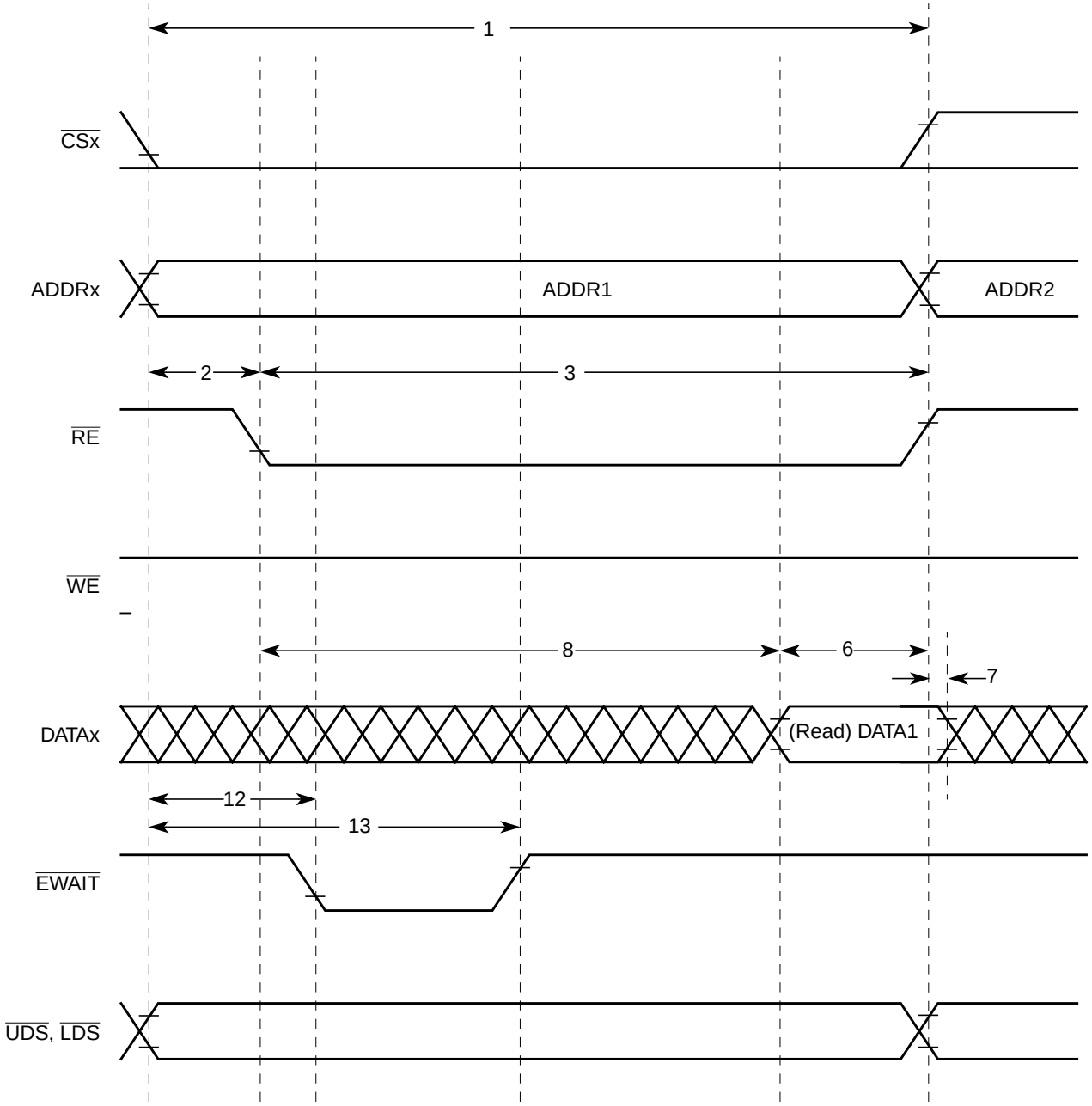


Figure A-12. Example 1b: Normal Expanded Mode — Stretched Read Access

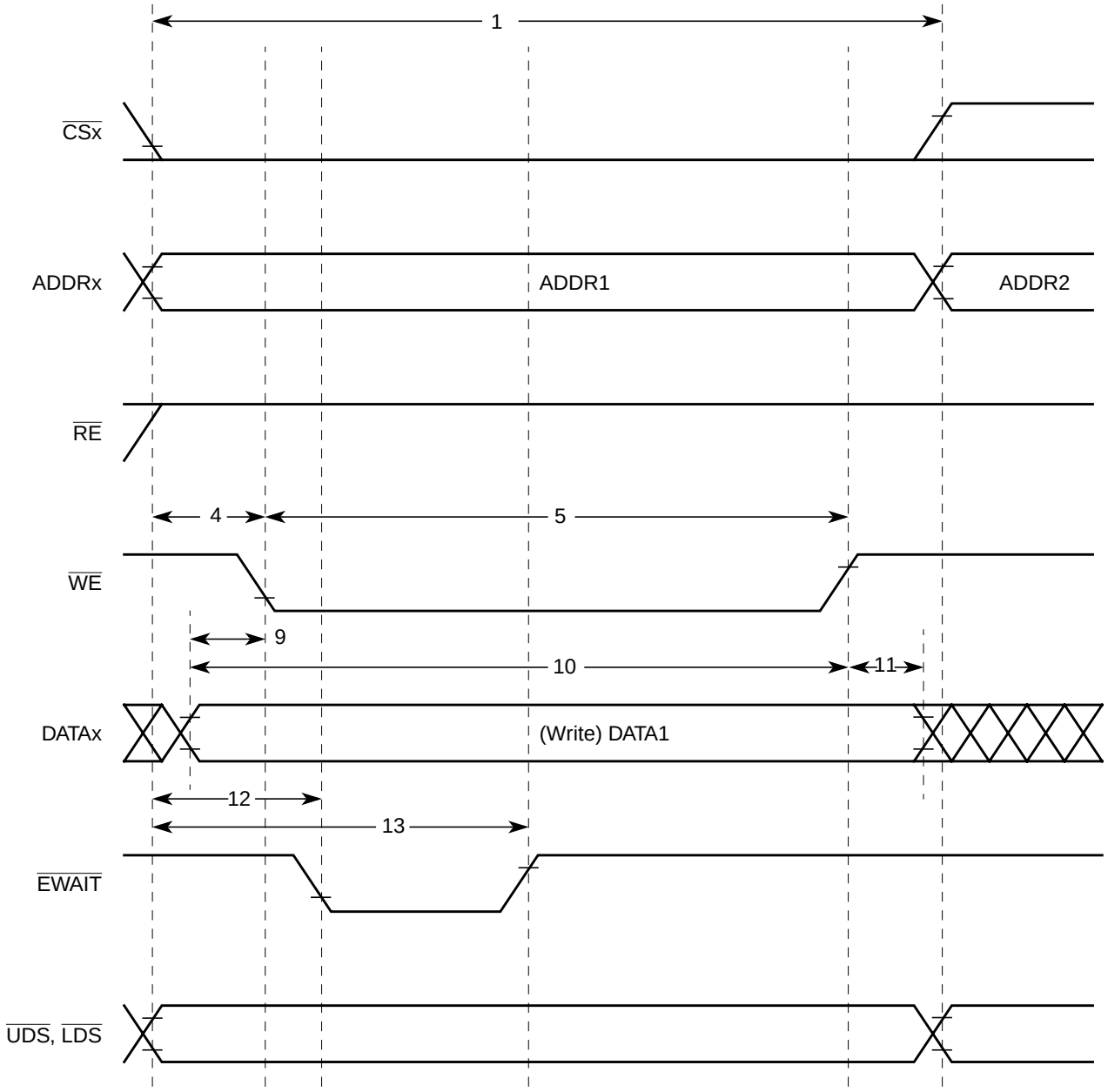


Figure A-13. Example 1b: Normal Expanded Mode — Stretched Write Access

0x0080–0x00AF Analog-to-Digital Converter 10-bit 16-Channels (ATD1) Map (Sheet 2 of 3)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0096	ATD1DR3H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x0097	ATD1DR3L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x0098	ATD1DR4H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x0099	ATD1DR4L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x009A	ATD1DR5H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x009B	ATD1DR5L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x009C	ATD1DR6H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x009D	ATD1DR6L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x009E	ATD1DR7H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x009F	ATD1DR7L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x00A0	ATD1DR8H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x00A1	ATD1DR8L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x00A2	ATD1DR9H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x00A3	ATD1DR9L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x00A4	ATD1DR10H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x00A5	ATD1DR10L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x00A6	ATD1DR11H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x00A7	ATD1DR11L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x00A8	ATD1DR12H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x00A9	ATD1DR12L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x00AA	ATD1DR13H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x00AB	ATD1DR13L	R	Bit7	Bit6	0	0	0	0	0	0
		W								