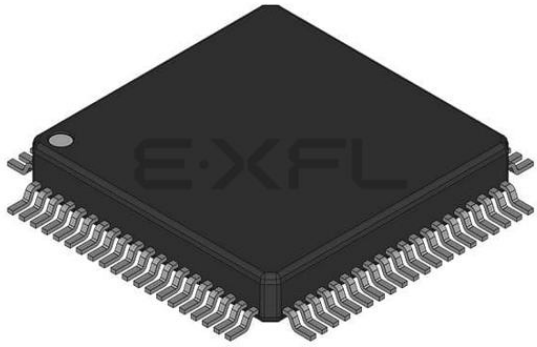


Welcome to E-XFL.COM

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	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	EBI/EMI, I²C, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	60
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 2.75V
Data Converters	A/D 16x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12e64mfue

1.4 Detailed Signal Descriptions

1.4.1 EXTAL, XTAL — Oscillator Pins

EXTAL and XTAL are the external clock and crystal driver pins. On reset all the device clocks are derived from the EXTAL input frequency. XTAL is the crystal output.

1.4.2 $\overline{\text{RESET}}$ — External Reset Pin

RESET is an active low bidirectional control signal that acts as an input to initialize the MCU to a known start-up state. It also acts as an open-drain output to indicate that an internal failure has been detected in either the clock monitor or COP watchdog circuit. External circuitry connected to the RESET pin should not include a large capacitance that would interfere with the ability of this signal to rise to a valid logic one within 32 ECLK cycles after the low drive is released. Upon detection of any reset, an internal circuit drives the RESET pin low and a clocked reset sequence controls when the MCU can begin normal processing.

1.4.3 TEST — Test Pin

The TEST pin is reserved for test and must be tied to VSS in all applications.

1.4.4 XFC — PLL Loop Filter Pin

Dedicated pin used to create the PLL loop filter. See the CRG block description chapter for more detailed information.

1.4.5 BKGD / $\overline{\text{TAGHI}}$ / MODC — Background Debug, Tag High & Mode Pin

The BKGD / $\overline{\text{TAGHI}}$ / MODC pin is used as a pseudo-open-drain pin for the background debug communication. It is used as a MCU operating mode select pin during reset. The state of this pin is latched to the MODC bit at the rising edge of RESET. In MCU expanded modes of operation, when instruction tagging is on, an input low on this pin during the falling edge of E-clock tags the high half of the instruction word being read into the instruction queue. This pin always has an internal pull up.

1.4.6 PA[7:0] / ADDR[15:8] / DATA[15:8] — Port A I/O Pins

PA[7:0] are general purpose input or output pins. In MCU expanded modes of operation, these pins are used for the multiplexed external address and data bus. PA[7:0] pins are not available in the 80 pin package version.

1.4.7 PB[7:0] / ADDR[7:0] / DATA[7:0] — Port B I/O Pins

PB[7:0] are general purpose input or output pins. In MCU expanded modes of operation, these pins are used for the multiplexed external address and data bus. PB[7:0] pins are not available in the 80 pin package version.

1.7 Security

The device will make available a security feature preventing the unauthorized read and write of the memory contents. This feature allows:

- Protection of the contents of FLASH,
- Operation in single-chip mode,
- Operation from external memory with internal FLASH disabled.

The user must be reminded that part of the security must lie with the user's code. An extreme example would be user's code that dumps the contents of the internal program. This code would defeat the purpose of security. At the same time the user may also wish to put a back door in the user's program. An example of this is the user downloads a key through the SCI which allows access to a programming routine that updates parameters.

1.7.1 Securing the Microcontroller

Once the user has programmed the FLASH, the part can be secured by programming the security bits located in the FLASH module. These non-volatile bits will keep the part secured through resetting the part and through powering down the part.

The security byte resides in a portion of the Flash array.

Check the Flash block description chapter for more details on the security configuration.

1.7.2 Operation of the Secured Microcontroller

1.7.2.1 Normal Single Chip Mode

This will be the most common usage of the secured part. Everything will appear the same as if the part was not secured with the exception of BDM operation. The BDM operation will be blocked.

1.7.2.2 Executing from External Memory

The user may wish to execute from external space with a secured microcontroller. This is accomplished by resetting directly into expanded mode. The internal FLASH will be disabled. BDM operations will be blocked.

1.7.3 Unsecuring the Microcontroller

In order to unsecure the microcontroller, the internal FLASH must be erased. This can be done through an external program in expanded mode.

Once the user has erased the FLASH, the part can be reset into special single chip mode. This invokes a program that verifies the erasure of the internal FLASH. Once this program completes, the user can erase and program the FLASH security bits to the unsecured state. This is generally done through the BDM, but the user could also change to expanded mode (by writing the mode bits through the BDM) and jumping to

3.3.4 Port Q

Port Q is associated with the Pulse Width Modulator (PMF) modules. Each pin is assigned according to the following priority: PMF > general-purpose I/O.

When a current status or fault function is enabled, the corresponding pin becomes an input. PQ[3:0] are connected to FAULT[3:0] inputs and PQ[6:4] are connected to $\overline{IS}$[2:0] inputs of the PMF module. Refer to the PMF block description chapter for information on enabling and disabling these PMF functions.

During reset, port Q pins are configured as high-impedance inputs.

3.3.4.1 Port Q I/O Register (PTQ)

	7	6	5	4	3	2	1	0
R	0	PTQ5	PTQ5	PTQ4	PTQ3	PTQ2	PTQ1	PTQ0
W								
PMF:		$\overline{IS2}$	$\overline{IS1}$	$\overline{IS0}$	FAULT3	FAULT2	FAULT1	FAULT0
Reset	0	0	0	0	0	0	0	0

= Reserved or Unimplemented

Figure 3-23. Port Q I/O Register (PTQ)

Read: Anytime. Write: Anytime.

If the associated data direction bit (DDRQx) is set to 1 (output), a read returns the value of the I/O register bit. If the associated data direction bit (DDRQx) is set to 0 (input), a read returns the value of the pin.

3.3.4.2 Port Q Input Register (PTIQ)

	7	6	5	4	3	2	1	0
R	0	PTIQ6	PTIQ5	PTIQ4	PTIQ3	PTIQ2	PTIQ1	PTIQ0
W								
Reset	0	u	u	u	u	u	u	u

= Reserved or Unimplemented u = Unaffected by reset

Figure 3-24. Port Q Input Register (PTIQ)

Read: Anytime. Write: Never, writes to this register have no effect.

This register always reads back the status of the associated pins.

3.3.5.3 Port S Data Direction Register (DDRS)

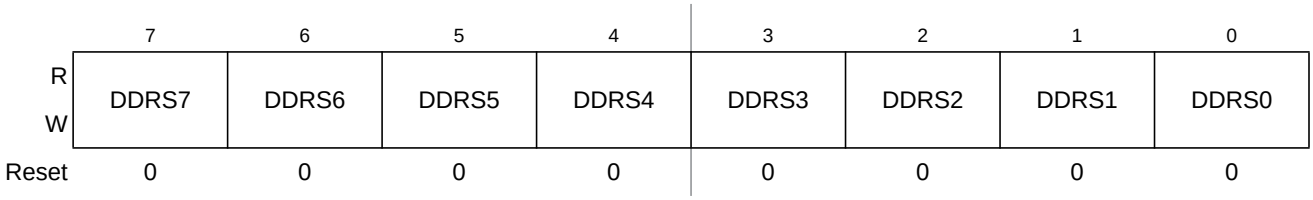


Figure 3-31. Port S Data Direction Register (DDRS)

Read: Anytime. Write: Anytime.

This register configures port pins PS[7:4] and PS[2:0] as either input or output.

When the SPI is enabled, the PS[7:4] pins become the SPI bidirectional pins. The associated Data Direction Register bits have no effect.

When the SCI1 transmitter is enabled, the PS[3] pin becomes the TXD1 output pin and the associated Data Direction Register bit has no effect. When the SCI1 receiver is enabled, the PS[2] pin becomes the RXD1 input pin and the associated Data Direction Register bit has no effect.

When the SCI0 transmitter is enabled, the PS[1] pin becomes the TXD0 output pin and the associated Data Direction Register bit has no effect. When the SCI0 receiver is enabled, the PS[0] pin becomes the RXD0 input pin and the associated Data Direction Register bit has no effect.

If the SPI, SCI1 and SCI0 functions are disabled, the corresponding Data Direction Register bit reverts to control the I/O direction of the associated pin.

Table 3-22. DDRS Field Descriptions

Field	Description
7:0 DDRS[7:0]	**Data Direction Port S** 0 Associated pin is configured as input. 1 Associated pin is configured as output.

The VCO has a minimum operating frequency, which corresponds to the self-clock mode frequency f_{SCM}.

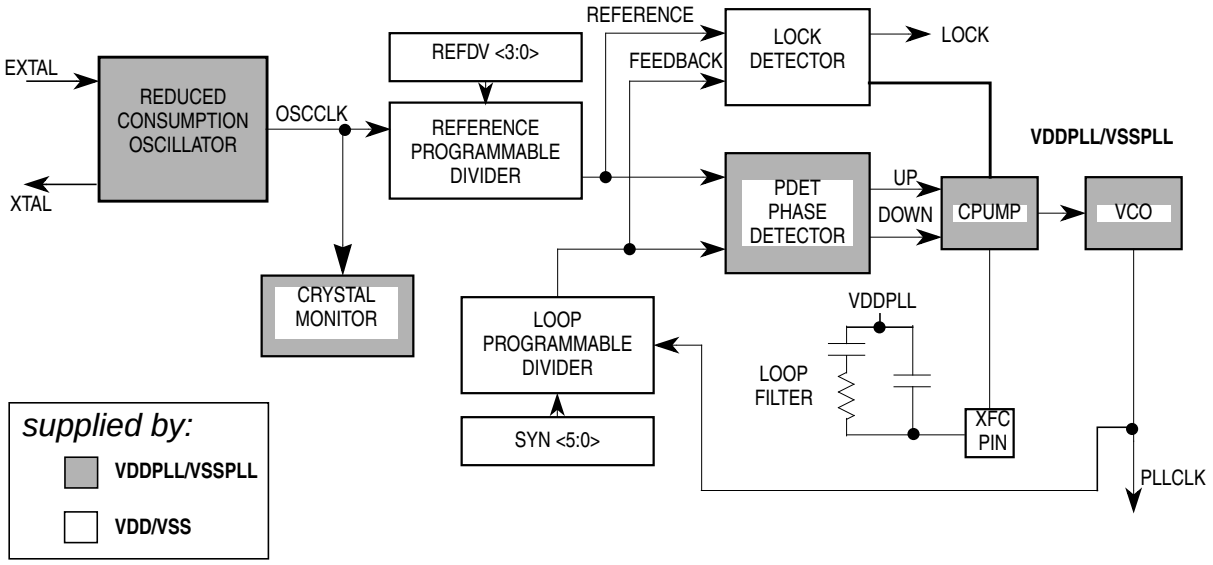


Figure 4-16. PLL Functional Diagram

4.4.1.1 PLL Operation

The oscillator output clock signal (OSCCLK) is fed through the reference programmable divider and is divided in a range of 1 to 16 (REFDV+1) to output the reference clock. The VCO output clock, (PLLCLK) is fed back through the programmable loop divider and is divided in a range of 2 to 128 in increments of [2 x (SYNR +1)] to output the feedback clock. See Figure 4-16.

The phase detector then compares the feedback clock, with the reference clock. Correction pulses are generated based on the phase difference between the two signals. The loop filter then slightly alters the DC voltage on the external filter capacitor connected to XFC pin, based on the width and direction of the correction pulse. The filter can make fast or slow corrections depending on its mode, as described in the next subsection. The values of the external filter network and the reference frequency determine the speed of the corrections and the stability of the PLL.

4.4.1.2 Acquisition and Tracking Modes

The lock detector compares the frequencies of the feedback clock, and the reference clock. Therefore, the speed of the lock detector is directly proportional to the final reference frequency. The circuit determines the mode of the PLL and the lock condition based on this comparison.

Table 4-11. Outcome of Clock Loss in Wait Mode (continued)

CME	SCME	SCMIE	CRG Actions
1	1	1	Clock failure --> – VREG enabled, – PLL enabled, – SCM activated, – Start Clock Quality Check, – SCMIF set. SCMIF generates Self-Clock Mode wakeup interrupt. – Exit Wait Mode in SCM using PLL clock (f_{SCM}) as system clock, – Continue to perform a additional Clock Quality Checks until OSCCLK is o.k. again.

4.4.10 Low-Power Operation in Stop Mode

All clocks are stopped in STOP mode, dependent of the setting of the PCE, PRE and PSTP bit. The oscillator is disabled in STOP mode unless the PSTP bit is set. All counters and dividers remain frozen but do not initialize. If the PRE or PCE bits are set, the RTI or COP continues to run in pseudo-stop mode. In addition to disabling system and core clocks the CRG requests other functional units of the MCU (e.g. voltage-regulator) to enter their individual power-saving modes (if available). This is the main difference between pseudo-stop mode and wait mode.

After executing the STOP instruction the core requests the CRG to switch the MCU into stop mode. If the PLLSEL bit remains set when entering stop mode, the CRG will switch the system and core clocks to OSCCLK by clearing the PLLSEL bit. Then the CRG disables the PLL, disables the core clock and finally disables the remaining system clocks. As soon as all clocks are switched off, stop mode is active.

If pseudo-stop mode (PSTP = 1) is entered from self-clock mode the CRG will continue to check the clock quality until clock check is successful. The PLL and the voltage regulator (VREG) will remain enabled. If full stop mode (PSTP = 0) is entered from self-clock mode an ongoing clock quality check will be stopped. A complete timeout window check will be started when stop mode is exited again.

Wake-up from stop mode also depends on the setting of the PSTP bit.

Table 6-1. ATD10B16CV2 Memory Map

Address Offset	Use	Access
0x0000	ATD Control Register 0 (ATDCTL0)[1]	R/W
0x0001	ATD Control Register 1 (ATDCTL1)[2]	R/W
0x0002	ATD Control Register 2 (ATDCTL2)	R/W
0x0003	ATD Control Register 3 (ATDCTL3)	R/W
0x0004	ATD Control Register 4 (ATDCTL4)	R/W
0x0005	ATD Control Register 5 (ATDCTL5)	R/W
0x0006	ATD Status Register 0 (ATDSTAT0)	R/W
0x0007	Unimplemented	
0x0008	ATD Test Register 0 (ATDTEST0)[3]	R
0x0009	ATD Test Register 1 (ATDTEST1)	R/W
0x000A	ATD Status Register 2 (ATDSTAT2)	R
0x000B	ATD Status Register 1 (ATDSTAT1)	R
0x000C	ATD Input Enable Register 0 (ATDDIEN0)	R/W
0x000D	ATD Input Enable Register 1 (ATDDIEN1)	R/W
0x000E	Port Data Register 0 (PORTAD0)	R
0x000F	Port Data Register 1 (PORTAD1)	R
0x0010, 0x0011	ATD Result Register 0 (ATDDR0H, ATDDR0L)	R/W
0x0012, 0x0013	ATD Result Register 1 (ATDDR1H, ATDDR1L)	R/W
0x0014, 0x0015	ATD Result Register 2 (ATDDR2H, ATDDR2L)	R/W
0x0016, 0x0017	ATD Result Register 3 (ATDDR3H, ATDDR3L)	R/W
0x0018, 0x0019	ATD Result Register 4 (ATDDR4H, ATDDR4L)	R/W
0x001A, 0x001B	ATD Result Register 5 (ATDDR5H, ATDDR5L)	R/W
0x001C, 0x001D	ATD Result Register 6 (ATDDR6H, ATDDR6L)	R/W
0x001E, 0x001F	ATD Result Register 7 (ATDDR7H, ATDDR7L)	R/W
0x0020, 0x0021	ATD Result Register 8 (ATDDR8H, ATDDR8L)	R/W
0x0022, 0x0023	ATD Result Register 9 (ATDDR9H, ATDDR9L)	R/W
0x0024, 0x0025	ATD Result Register 10 (ATDDR10H, ATDDR10L)	R/W
0x0026, 0x0027	ATD Result Register 11 (ATDDR11H, ATDDR11L)	R/W
0x0028, 0x0029	ATD Result Register 12 (ATDDR12H, ATDDR12L)	R/W
0x002A, 0x002B	ATD Result Register 13 (ATDDR13H, ATDDR13L)	R/W
0x002C, 0x002D	ATD Result Register 14 (ATDDR14H, ATDDR14L)	R/W
0x002E, 0x002F	ATD Result Register 15 (ATDDR15H, ATDDR15L)	R/W

1 ATDCTL0 is intended for factory test purposes only.

2 ATDCTL1 is intended for factory test purposes only.

3 ATDTEST0 is intended for factory test purposes only.

NOTE

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

Table 6-10. ATDCTL5 Field Descriptions (continued)

Field	Description
3:0 C[D:A}	**Analog Input Channel Select Code** — These bits select the analog input channel(s) whose signals are sampled and converted to digital codes. Table 6-13 lists the coding used to select the various analog input channels. In the case of single channel conversions (MULT = 0), this selection code specified the channel to be examined. In the case of multiple channel conversions (MULT = 1), this selection code represents the first channel to be examined in the conversion sequence. Subsequent channels are determined by incrementing the channel selection code or wrapping around to AN0 (after converting the channel defined by the Wrap Around Channel Select Bits WRAP[3:0] in ATDCTL0). In case starting with a channel number higher than the one defined by WRAP[3:0] the first wrap around will be AN15 to AN0.

Table 6-11. Available Result Data Formats.

SRES8	DJM	DSGN	Result Data Formats Description and Bus Bit Mapping
1	0	0	8-bit / left justified / unsigned — bits 15:8
1	0	1	8-bit / left justified / signed — bits 15:8
1	1	X	8-bit / right justified / unsigned — bits 7:0
0	0	0	10-bit / left justified / unsigned — bits 15:6
0	0	1	10-bit / left justified / signed -— bits 15:6
0	1	X	10-bit / right justified / unsigned — bits 9:0

Table 6-12. Left Justified, Signed and Unsigned ATD Output Codes.

Input Signal V_{RL} = 0 Volts V_{RH} = 5.12 Volts	Signed 8-Bit Codes	Unsigned 8-Bit Codes	Signed 10-Bit Codes	Unsigned 10-Bit Codes
5.120 Volts	7F	FF	7FC0	FFC0
5.100	7F	FF	7F00	FF00
5.080	7E	FE	7E00	FE00
2.580	01	81	0100	8100
2.560	00	80	0000	8000
2.540	FF	7F	FF00	7F00
0.020	81	01	8100	0100
0.000	80	00	8000	0000

Table 10-5. Multiplier Factor

IBC7-6	MUL
00	01
01	02
10	04
11	RESERVED

The number of clocks from the falling edge of SCL to the first tap (Tap[1]) is defined by the values shown in the scl2tap column of Table 10-4, all subsequent tap points are separated by 2^{IBC5-3} as shown in the tap2tap column in Table 10-4. The SCL Tap is used to generated the SCL period and the SDA Tap is used to determine the delay from the falling edge of SCL to SDA changing, the SDA hold time.

IBC7–6 defines the multiplier factor MUL. The values of MUL are shown in the Table 10-5.

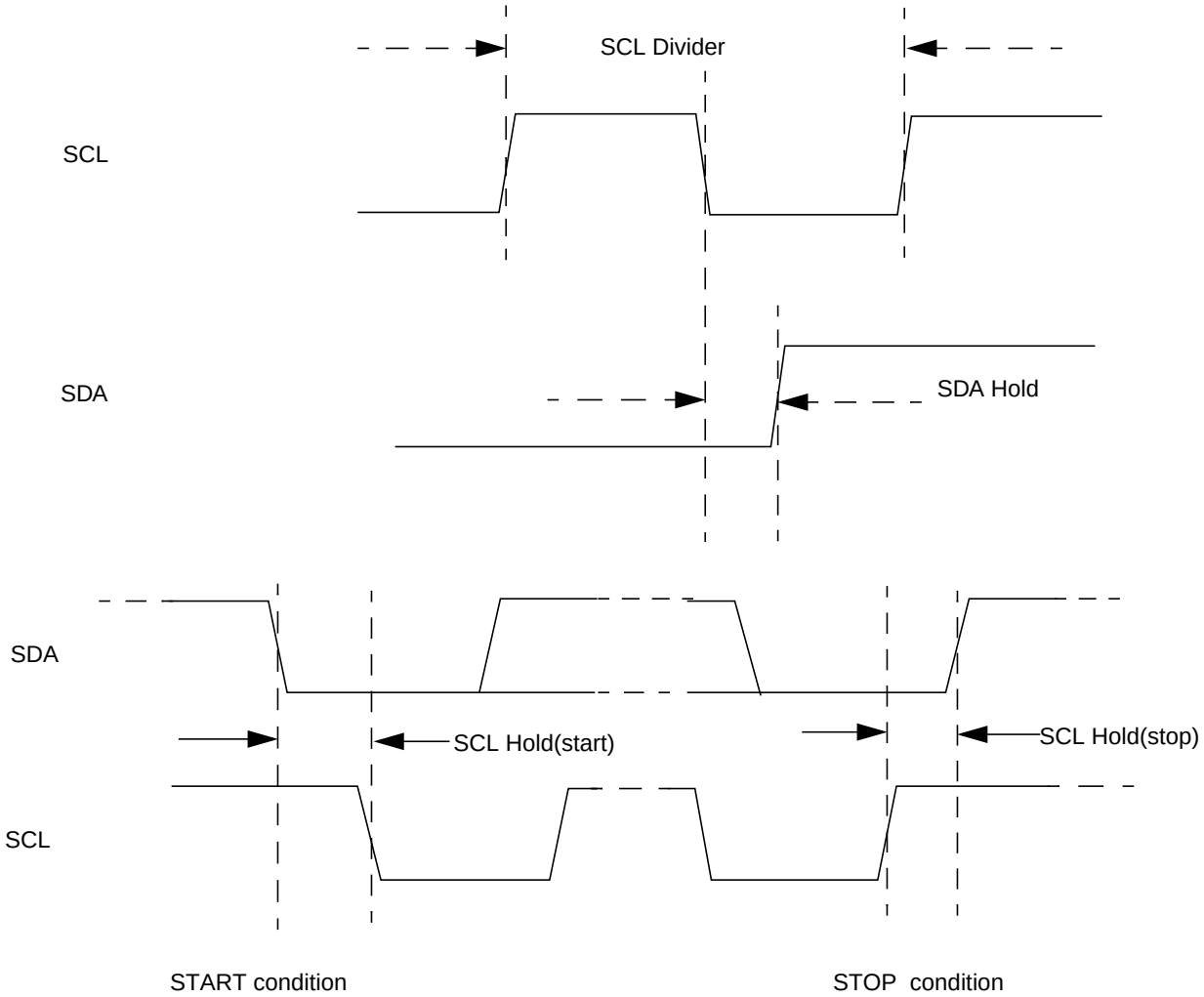


Figure 10-4. SCL Divider and SDA Hold

The equation used to generate the divider values from the IBFD bits is:

SCL Divider = MUL x {2 x (scl2tap + [(SCL_Tap -1) x tap2tap] + 2)}

Table 10-6. IIC Divider and Hold Values (Sheet 2 of 5)

IBC[7:0] (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
23	256	33	126	129
24	288	49	142	145
25	320	49	158	161
26	384	65	190	193
27	480	65	238	241
28	320	33	158	161
29	384	33	190	193
2A	448	65	222	225
2B	512	65	254	257
2C	576	97	286	289
2D	640	97	318	321
2E	768	129	382	385
2F	960	129	478	481
30	640	65	318	321
31	768	65	382	385
32	896	129	446	449
33	1024	129	510	513
34	1152	193	574	577
35	1280	193	638	641
36	1536	257	766	769
37	1920	257	958	961
38	1280	129	638	641
39	1536	129	766	769
3A	1792	257	894	897
3B	2048	257	1022	1025
3C	2304	385	1150	1153
3D	2560	385	1278	1281
3E	3072	513	1534	1537
3F	3840	513	1918	1921
MUL=2				
40	40	14	12	22
41	44	14	14	24
42	48	16	16	26
43	52	16	18	28
44	56	18	20	30
45	60	18	22	32
46	68	20	26	36
47	80	20	32	42
48	56	14	20	30
49	64	14	24	34
4A	72	18	28	38
4B	80	18	32	42
4C	88	22	36	46
4D	96	22	40	50
4E	112	26	48	58

from where was during the previous transmission. It is not possible for the IIC to wake up the CPU when its internal clocks are stopped.

If it were the case that the IBSWAI bit was cleared when the WAI instruction was executed, the IIC internal clocks and interface would remain alive, continuing the operation which was currently underway. It is also possible to configure the IIC such that it will wake up the CPU via an interrupt at the conclusion of the current operation. See the discussion on the IBIF and IBIE bits in the IBSR and IBCR, respectively.

10.3.2.4 IIC Status Register (IBSR)

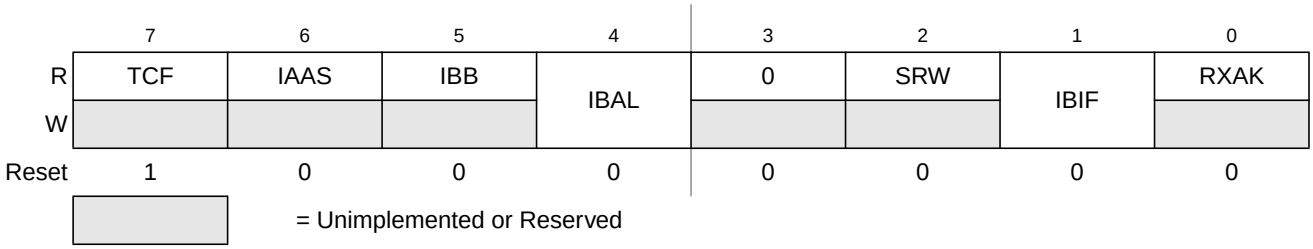


Figure 10-6. IIC Bus Status Register (IBSR)

This status register is read-only with exception of bit 1 (IBIF) and bit 4 (IBAL), which are software clearable.

Table 10-8. IBSR Field Descriptions

Field	Description
7 TCF	**Data Transferring Bit** — While one byte of data is being transferred, this bit is cleared. It is set by the falling edge of the 9th clock of a byte transfer. Note that this bit is only valid during or immediately following a transfer to the IIC module or from the IIC module. 0 Transfer in progress 1 Transfer complete
6 IAAS	**Addressed as a Slave Bit** — When its own specific address (I-bus address register) is matched with the calling address, this bit is set.The CPU is interrupted provided the IBIE is set.Then the CPU needs to check the SRW bit and set its Tx/$\overline{\text{Rx}}$ mode accordingly.Writing to the I-bus control register clears this bit. 0 Not addressed 1 Addressed as a slave
5 IBB	Bus Busy Bit 0 This bit indicates the status of the bus. When a START signal is detected, the IBB is set. If a STOP signal is detected, IBB is cleared and the bus enters idle state. 1 Bus is busy
4 IBAL	**Arbitration Lost** — The arbitration lost bit (IBAL) is set by hardware when the arbitration procedure is lost. Arbitration is lost in the following circumstances: 1. SDA sampled low when the master drives a high during an address or data transmit cycle. 2. SDA sampled low when the master drives a high during the acknowledge bit of a data receive cycle. 3. A start cycle is attempted when the bus is busy. 4. A repeated start cycle is requested in slave mode. 5. A stop condition is detected when the master did not request it. This bit must be cleared by software, by writing a one to it. A write of 0 has no effect on this bit.
3 RESERVED	**Reserved** — Bit 3 of IBSR is reserved for future use. A read operation on this bit will return 0.

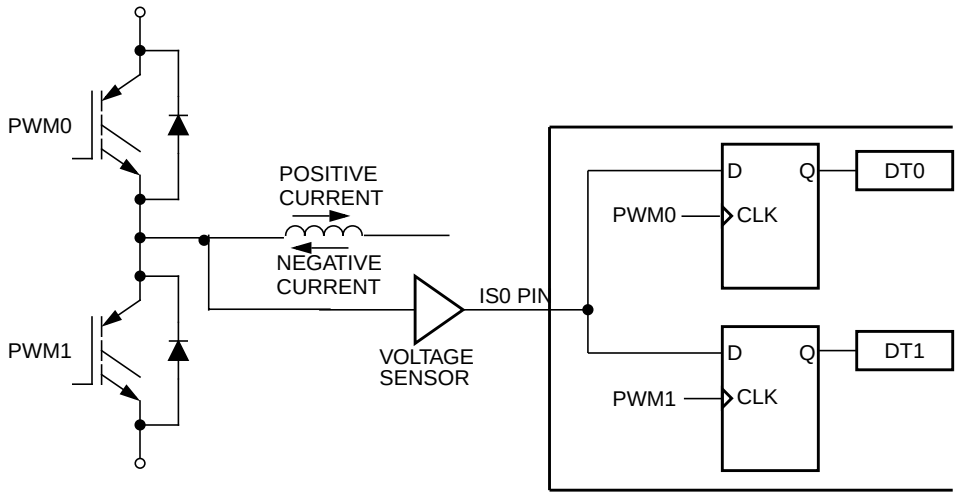
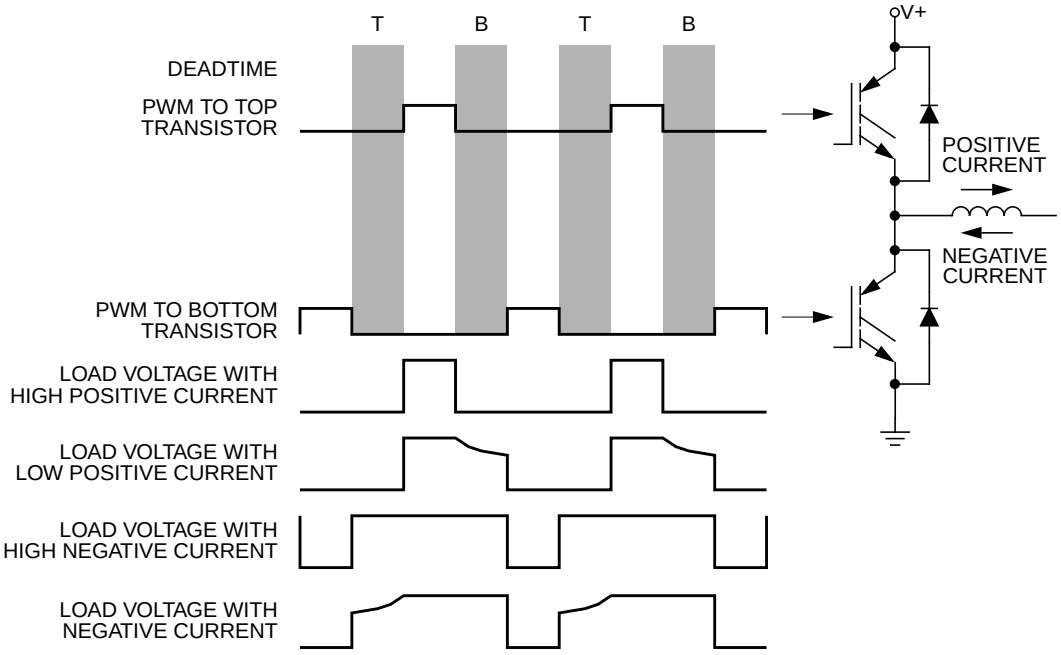


Figure 11-55. Current-Status Sense Scheme for Deadtime Correction

If both D flip-flops latch low, DT0 = 0, DT1 = 0, during deadtime periods if current is large and flowing out of the complementary circuit. See Figure 11-55. If both D flip-flops latch the high, DT0 = 1, DT1 = 1, during deadtime periods if current is also large and flowing into the complementary circuit. However, under low-current, the output voltage of the complementary circuit during deadtime is somewhere between the high and low levels. The current cannot free-wheel throughout the opposition anti-body diode, regardless of polarity, giving additional distortion when the current crosses zero. Sampled results will be DT0 = 0 and DT1 = 1. Thus, the best time to change one PWM value register to another is just before the current zero crossing.



T = DEADTIME INTERVAL BEFORE ASSERTION OF TOP PWM
B = DEADTIME INTERVAL BEFORE ASSERTION OF BOTTOM PWM

Figure 11-56. Output Voltage Waveforms

14.2 External Signal Description

Due to the nature of VREG3V3V2 being a voltage regulator providing the chip internal power supply voltages most signals are power supply signals connected to pads.

Table 14-1 shows all signals of VREG3V3V2 associated with pins.

Table 14-1. VREG3V3V2 — Signal Properties

Name	Port	Function	Reset State	Pull Up
V_{DDR}	—	VREG3V3V2 power input (positive supply)	—	—
V_{DDA}	—	VREG3V3V2 quiet input (positive supply)	—	—
V_{SSA}	—	VREG3V3V2 quiet input (ground)	—	—
V_{DD}	—	VREG3V3V2 primary output (positive supply)	—	—
V_{SS}	—	VREG3V3V2 primary output (ground)	—	—
V_{DDPLL}	—	VREG3V3V2 secondary output (positive supply)	—	—
V_{SSPLL}	—	VREG3V3V2 secondary output (ground)	—	—
V_{REGEN} (optional)	—	VREG3V3V2 (Optional) Regulator Enable	—	—

NOTE

Check device overview chapter for connectivity of the signals.

14.2.1 V_{DDR} — Regulator Power Input

Signal V_{DDR} is the power input of VREG3V3V2. All currents sourced into the regulator loads flow through this pin. A chip external decoupling capacitor (100 nF...220 nF, X7R ceramic) between V_{DDR} and V_{SSR} can smoothen ripple on V_{DDR}.

For entering shutdown mode, pin V_{DDR} should also be tied to ground on devices without a V_{REGEN} pin.

14.2.2 V_{DDA}, V_{SSA} — Regulator Reference Supply

Signals V_{DDA}/V_{SSA} which are supposed to be relatively quiet are used to supply the analog parts of the regulator. Internal precision reference circuits are supplied from these signals. A chip external decoupling capacitor (100 nF...220 nF, X7R ceramic) between V_{DDA} and V_{SSA} can further improve the quality of this supply.

16.3.2.9 Debug Comparator A Extended Register (DBGCAX)

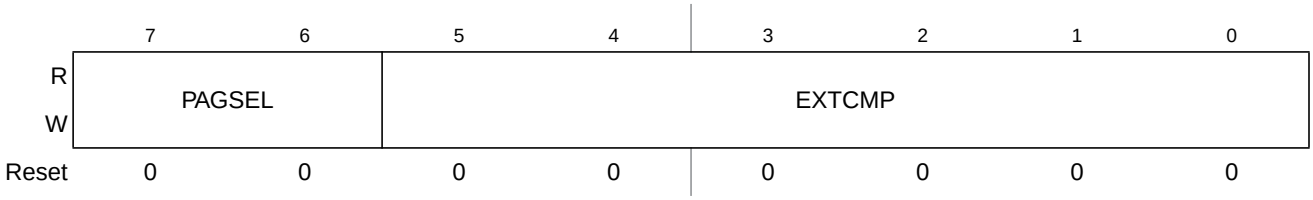


Figure 16-15. Debug Comparator A Extended Register (DBGCAX)

Table 16-19. DBGCAX Field Descriptions

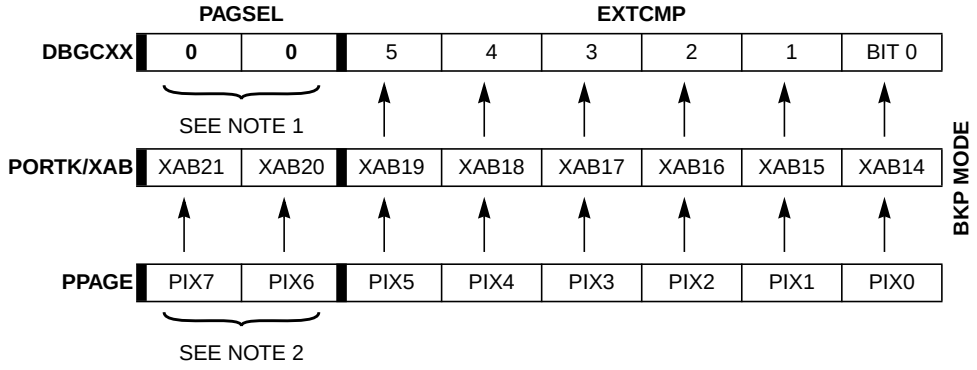
Field	Description
7:6 PAGSEL	**Page Selector Field** — If DBGEN is set in DBGC1, then PAGSEL selects the type of paging as shown in Table 16-20. DPAGE and EPAGE are not yet implemented so the value in bit 7 will be ignored (i.e., PAGSEL values of 10 and 11 will be interpreted as values of 00 and 01, respectively). In BKP mode, PAGSEL has no meaning and EXTCMP[5:0] are compared to address bits [19:14] if the address is in the FLASH/ROM memory space.
5:0 EXTCMP	**Comparator A Extended Compare Bits** — The EXTCMP bits are used as comparison address bits as shown in Table 16-20 along with the appropriate PPAGE, DPAGE, or EPAGE signal from the core.

Table 16-20. Comparator A or B Compares

Mode		EXTCMP Compare	High-Byte Compare
BKP[1]	Not FLASH/ROM access	No compare	DBGCxH[7:0] = AB[15:8]
	FLASH/ROM access	EXTCMP[5:0] = XAB[19:14]	DBGCxH[5:0] = AB[13:8]
DBG[2]	PAGSEL = 00	No compare	DBGCxH[7:0] = AB[15:8]
	PAGSEL = 01	EXTCMP[5:0] = XAB[21:16]	DBGCxH[7:0] = XAB[15:14], AB[13:8]

1 See Figure 16-16.

2 See Figure 16-10 (note that while this figure provides extended comparisons for comparator C, the figure also pertains to comparators A and B in DBG mode only).



NOTES:
1. In BKP mode, PAGSEL has no functionality. Therefore, set PAGSEL to 00 (reset state).
2. Current HCS12 implementations are limited to six PPAGE bits, PIX[5:0].

Figure 16-16. Comparators A and B Extended Comparison in BKP Mode

NOTE

BDM should not be entered from a breakpoint unless the ENABLE bit is set in the BDM. Even if the ENABLE bit in the BDM is cleared, the CPU actually executes the BDM firmware code. It checks the ENABLE and returns if ENABLE is not set. If the BDM is not serviced by the monitor then the breakpoint would be re-asserted when the BDM returns to normal CPU flow.

There is no hardware to enforce restriction of breakpoint operation if the BDM is not enabled.

When program control returns from a tagged breakpoint through an RTI or a BDM GO command, it will return to the instruction whose tag generated the breakpoint. Unless breakpoints are disabled or modified in the service routine or active BDM session, the instruction will be tagged again and the breakpoint will be repeated. In the case of BDM breakpoints, this situation can also be avoided by executing a TRACE1 command before the GO to increment the program flow past the tagged instruction.

16.4.1.4 Using Comparator C in BKP Mode

The original BKP_ST12_A module supports two breakpoints. The DBG_ST12_A module can be used in BKP mode and allow a third breakpoint using comparator C. Four additional bits, BKCEN, TAGC, RWCEN, and RWC in DBGC2 in conjunction with additional comparator C address registers, DBGCCX, DBGCCH, and DBGCCL allow the user to set up a third breakpoint. Using PAGSEL in DBGCCX for expanded memory will work differently than the way paged memory is done using comparator A and B in BKP mode. See Section 16.3.2.5, “Debug Comparator C Extended Register (DBGCCX),” for more information on using comparator C.

16.4.2 DBG Operating in DBG Mode

Enabling the DBG module in DBG mode, allows the arming, triggering, and storing of data in the trace buffer and can be used to cause CPU breakpoints. The DBG module is made up of three main blocks, the comparators, trace buffer control logic, and the trace buffer.

NOTE

In general, there is a latency between the triggering event appearing on the bus and being detected by the DBG circuitry. In general, tagged triggers will be more predictable than forced triggers.

16.4.2.1 Comparators

The DBG contains three comparators, A, B, and C. Comparator A compares the core address bus with the address stored in DBGCAH and DBGCAL. Comparator B compares the core address bus with the address stored in DBGCBH and DBGCBL except in full mode, where it compares the data buses to the data stored in DBGCBH and DBGCBL. Comparator C can be used as a breakpoint generator or as the address comparison unit in the loop1 mode. Matches on comparator A, B, and C are signaled to the trace buffer

16.4.2.6 Capture Modes

The DBG in DBG mode can operate in four capture modes. These modes are described in the following subsections.

16.4.2.6.1 Normal Mode

In normal mode, the DBG module uses comparator A and B as triggering devices. Change-of-flow information or data will be stored depending on TRG in DBGSC.

16.4.2.6.2 Loop1 Mode

The intent of loop1 mode is to prevent the trace buffer from being filled entirely with duplicate information from a looping construct such as delays using the DBNE instruction or polling loops using BRSET/BRCLR instructions. Immediately after address information is placed in the trace buffer, the DBG module writes this value into the C comparator and the C comparator is placed in ignore address mode. This will prevent duplicate address entries in the trace buffer resulting from repeated bit-conditional branches. Comparator C will be cleared when the ARM bit is set in loop1 mode to prevent the previous contents of the register from interfering with loop1 mode operation. Breakpoints based on comparator C are disabled.

Loop1 mode only inhibits duplicate source address entries that would typically be stored in most tight looping constructs. It will not inhibit repeated entries of destination addresses or vector addresses, because repeated entries of these would most likely indicate a bug in the user's code that the DBG module is designed to help find.

NOTE

In certain very tight loops, the source address will have already been fetched again before the C comparator is updated. This results in the source address being stored twice before further duplicate entries are suppressed. This condition occurs with branch-on-bit instructions when the branch is fetched by the first P-cycle of the branch or with loop-construct instructions in which the branch is fetched with the first or second P cycle. See examples below:

```
LOOP   INCX                    ; 1-byte instruction fetched by 1st P-cycle of BRCLR
       BRCLR CMPTMP,#$0c,LOOP  ; the BRCLR instruction also will be fetched by 1st P-cycle of BRCLR

LOOP2  BRN   *                 ; 2-byte instruction fetched by 1st P-cycle of DBNE
       NOP                     ; 1-byte instruction fetched by 2nd P-cycle of DBNE
       DBNE  A,LOOP2           ; this instruction also fetched by 2nd P-cycle of DBNE
```

NOTE

Loop1 mode does not support paged memory, and inhibits duplicate entries in the trace buffer based solely on the CPU address. There is a remote possibility of an erroneous address match if program flow alternates between paged and unpaged memory space.

18.3.2.9 Mode Register (MODE)

	7	6	5	4	3	2	1	0
R	MODC	MODB	MODA	0	IVIS	0	EMK	EME
W								
Reset								
Special Single Chip	0	0	0	0	0	0	0	0
Emulation Expanded Narrow	0	0	1	0	1	0	1	1
Special Test	0	1	0	0	1	0	0	0
Emulation Expanded Wide	0	1	1	0	1	0	1	1
Normal Single Chip	1	0	0	0	0	0	0	0
Normal Expanded Narrow	1	0	1	0	0	0	0	0
Peripheral	1	1	0	0	0	0	0	0
Normal Expanded Wide	1	1	1	0	0	0	0	0

= Unimplemented or Reserved

Figure 18-13. Mode Register (MODE)

Read: Anytime (provided this register is in the map).

Write: Each bit has specific write conditions. Please refer to the descriptions of each bit on the following pages.

The MODE register is used to establish the operating mode and other miscellaneous functions (i.e., internal visibility and emulation of port E and K).

In special peripheral mode, this register is not accessible but it is reset as shown to system configuration features. Changes to bits in the MODE register are delayed one cycle after the write.

This register is not in the on-chip memory map in expanded and special peripheral modes. Therefore, these accesses will be echoed externally.

A.1.10.1 Measurement Conditions

All measurements are without output loads. Unless otherwise noted the currents are measured in single chip mode, internal voltage regulator enabled and at 25MHz bus frequency using a 4MHz oscillator.

A.1.10.2 Additional Remarks

In expanded modes the currents flowing in the system are highly dependent on the load at the address, data and control signals as well as on the duty cycle of those signals. No generally applicable numbers can be given. A very good estimate is to take the single chip currents and add the currents due to the external loads.

Table A-8. Supply Current Characteristics

Conditions are shown in Table A-4 unless otherwise noted							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	P	Run supply currents Single Chip, Internal regulator enabled	I_{DD5}	—	—	65	mA
2		Wait Supply current	I_{DDW}				mA
	P	All modules enabled		—	—	40	
	P	only RTI enabled		—	—	5	
3		Pseudo Stop Current (RTI and COP enabled) [1, 2]	I_{DDPS}				µA
	C	–40°C		—	570	—	
	C	27°C		—	600	—	
	C	70°C		—	650	—	
	C	85°C		—	750	—	
	C	105°C		—	850	—	
	C	125°C		—	1200	—	
	C	140°C		—	1500	—	
4		Pseudo Stop Current (RTI and COP disabled) [1,2]	I_{DDPS}				µA
	C	-40°C		—	370	—	
	P	27°C		—	400	500	
	C	70°C		—	450	—	
	C	85°C		—	550	—	
	P	"C" Temp Option 100°C		—	600	1600	
	C	105°C		—	650	—	
	P	"V" Temp Option 120°C		—	800	2100	
	C	125°C		—	850	—	
	P	"M" Temp Option 140°C		—	1200	5000	
5		Stop Current [2]	I_{DDS}				µA
	C	-40°C		—	12	—	
	P	27°C		—	30	100	
	C	70°C		—	100	—	
	C	85°C		—	130	—	
	P	"C" Temp Option 100°C		—	160	1200	
	C	105°C		—	200	—	
	P	"V" Temp Option 120°C		—	350	1700	
	C	125°C		—	400	—	
	P	"M" Temp Option 140°C		—	600	5000	

1 PLL off

2 At those low power dissipation levels $T_J = T_A$ can be assumed

Table A-13. PLL Characteristics

Conditions are shown in Table A-4 unless otherwise noted							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	P	Self Clock Mode frequency	f_{SCM}	1		5.5	MHz
2	D	VCO locking range	f_{VCO}	8		50	MHz
3	D	Lock Detector transition from Acquisition to Tracking mode	$\|\Delta_{trk}\|$	3		4	%[1]
4	D	Lock Detection	$\|\Delta_{Lock}\|$	0		1.5	%[1]
5	D	Un-Lock Detection	$\|\Delta_{unl}\|$	0.5		2.5	%[1]
6	D	Lock Detector transition from Tracking to Acquisition mode	$\|\Delta_{unt}\|$	6		8	%[1]
7	C	PLLON Total Stabilization delay (Auto Mode) [2]	t_{stab}		0.5		ms
8	D	PLLON Acquisition mode stabilization delay [2]	t_{acq}		0.3		ms
9	D	PLLON Tracking mode stabilization delay [2]	t_{al}		0.2		ms
10	D	Fitting parameter VCO loop gain	K_1		-100		MHz/V
11	D	Fitting parameter VCO loop frequency	f_1		60		MHz
12	D	Charge pump current acquisition mode	$\| i_{ch} \|$		38.5		µA
13	D	Charge pump current tracking mode	$\| i_{ch} \|$		3.5		µA
14	C	Jitter fit parameter 1[2]	j_1			1.1	%
15	C	Jitter fit parameter 2[2]	j_2			0.13	%

[1] % deviation from target frequency
[2] f_{OSC} = 4MHz, f_{BUS} = 25MHz equivalent f_{VCO} = 50MHz: REFDV = #$03, SYNR = #$018, Cs = 4.7nF, Cp = 470pF, Rs = 10KΩ.

A.4 Flash NVM

A.4.1 NVM Timing

The time base for all NVM program or erase operations is derived from the oscillator. A minimum oscillator frequency f_{NVMOSC} is required for performing program or erase operations. The NVM modules do not have any means to monitor the frequency and will not prevent program or erase operation at frequencies above or below the specified minimum. Attempting to program or erase the NVM modules at a lower frequency a full program or erase transition is not assured.

The Flash program and erase operations are timed using a clock derived from the oscillator using the FCLKDIV register. The frequency of this clock must be set within the limits specified as f_{NVMOP}.

The minimum program and erase times shown in Table A-14 are calculated for maximum f_{NVMOP} and maximum f_{bus}. The maximum times are calculated for minimum f_{NVMOP} and a f_{bus} of 2MHz.

A.4.1.1 Single Word Programming

The programming time for single word programming is dependent on the bus frequency as a well as on the frequency f_{NVMOP} and can be calculated according to the following formula.

$$t_{swpgm} = 9 \cdot \frac{1}{f_{NVMOP}} + 25 \cdot \frac{1}{f_{bus}}$$

A.4.1.2 Row Programming

Flash programming where up to 64 words in a row can be programmed consecutively by keeping the command pipeline filled. The time to program a consecutive word can be calculated as:

$$t_{bwpgm} = 4 \cdot \frac{1}{f_{NVMOP}} + 9 \cdot \frac{1}{f_{bus}}$$

The time to program a whole row is:

$$t_{brpgm} = t_{swpgm} + 63 \cdot t_{bwpgm}$$

Row programming is more than 2 times faster than single word programming.

A.4.1.3 Sector Erase

Erasing a 1024 byte Flash sector takes:

$$t_{era} \approx 4000 \cdot \frac{1}{f_{NVMOP}}$$

The setup times can be ignored for this operation.