



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](#)"

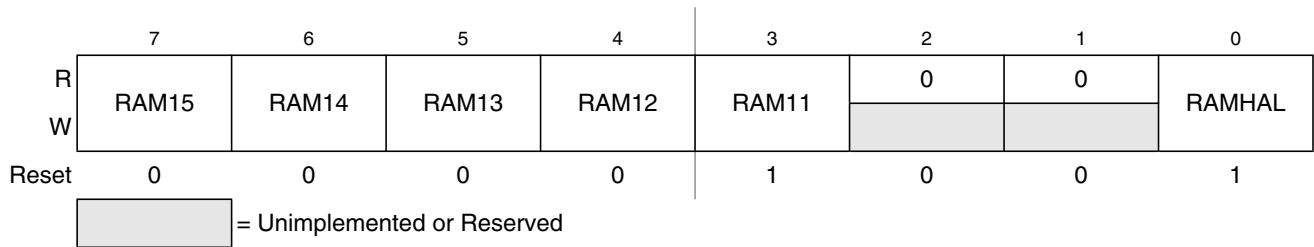
#### Details

Product Status	Obsolete
Core Processor	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	EBI/EMI, 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 ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/mcs12gc64vfue">https://www.e-xfl.com/product-detail/nxp-semiconductors/mcs12gc64vfue</a>

### 3.3.2.1 Initialization of Internal RAM Position Register (INITRM)

Module Base + 0x0010

Starting address location affected by INITRG register setting.



**Figure 3-3. Initialization of Internal RAM Position Register (INITRM)**

Read: Anytime

Write: Once in normal and emulation modes, anytime in special modes

#### NOTE

Writes to this register take one cycle to go into effect.

This register initializes the position of the internal RAM within the on-chip system memory map.

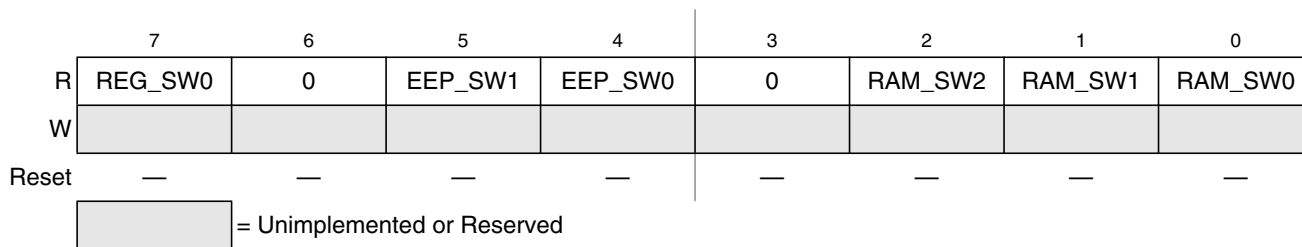
**Table 3-2. INITRM Field Descriptions**

Field	Description
7:3 RAM[15:11]	<b>Internal RAM Map Position</b> — These bits determine the upper five bits of the base address for the system's internal RAM array.
0 RAMHAL	<b>RAM High-Align</b> — RAMHAL specifies the alignment of the internal RAM array. 0 Aligns the RAM to the lowest address (0x0000) of the mappable space 1 Aligns the RAM to the higher address (0xFFFF) of the mappable space

### 3.3.2.7 Memory Size Register 0 (MEMSIZ0)

Module Base + 0x001C

Starting address location affected by INITRG register setting.



**Figure 3-9. Memory Size Register 0 (MEMSIZ0)**

Read: Anytime

Write: Writes have no effect

Reset: Defined at chip integration, see device overview section.

The MEMSIZ0 register reflects the state of the register, EEPROM and RAM memory space configuration switches at the core boundary which are configured at system integration. This register allows read visibility to the state of these switches.

**Table 3-7. MEMSIZ0 Field Descriptions**

Field	Description
7 REG_SW0	<b>Allocated System Register Space</b> 0 Allocated system register space size is 1K byte 1 Allocated system register space size is 2K byte
5:4 EEP_SW[1:0]	<b>Allocated System EEPROM Memory Space</b> — The allocated system EEPROM memory space size is as given in <a href="#">Table 3-8</a> .
2 RAM_SW[2:0]	<b>Allocated System RAM Memory Space</b> — The allocated system RAM memory space size is as given in <a href="#">Table 3-9</a> .

**Table 3-8. Allocated EEPROM Memory Space**

eep_sw1:eep_sw0	Allocated EEPROM Space
00	0K byte
01	2K bytes
10	4K bytes
11	8K bytes

**Table 3-9. Allocated RAM Memory Space**

ram_sw2:ram_sw0	Allocated RAM Space	RAM Mappable Region	INITRM Bits Used	RAM Reset Base Address <sup>(1)</sup>
000	2K bytes	2K bytes	RAM[15:11]	0x0800
001	4K bytes	4K bytes	RAM[15:12]	0x0000
010	6K bytes	8K bytes <sup>(2)</sup>	RAM[15:13]	0x0800

Table 4-10. RDRIV Field Descriptions

Field	Description
7 RDRK	<b>Reduced Drive of Port K</b> 0 All port K output pins have full drive enabled. 1 All port K output pins have reduced drive enabled.
4 RDPE	<b>Reduced Drive of Port E</b> 0 All port E output pins have full drive enabled. 1 All port E output pins have reduced drive enabled.
1 RDPB	<b>Reduced Drive of Port B</b> 0 All port B output pins have full drive enabled. 1 All port B output pins have reduced drive enabled.
0 RDPA	<b>Reduced Drive of Ports A</b> 0 All port A output pins have full drive enabled. 1 All port A output pins have reduced drive enabled.

#### 4.3.2.12 External Bus Interface Control Register (EBICTL)

Module Base + 0x000E

Starting address location affected by INITRG register setting.

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	ESTR
W								
Reset:								
Peripheral	0	0	0	0	0	0	0	0
All other modes	0	0	0	0	0	0	0	1


 = Unimplemented or Reserved

Figure 4-16. External Bus Interface Control Register (EBICTL)

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

Write: Refer to individual bit descriptions below

The EBICTL register is used to control miscellaneous functions (i.e., stretching of external E clock).

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

Table 4-11. EBICTL Field Descriptions

Field	Description
0 ESTR	<b>E Clock Stretches</b> — This control bit determines whether the E clock behaves as a simple free-running clock or as a bus control signal that is active only for external bus cycles. Normal and Emulation: write once Special: write anytime 0 E never stretches (always free running). 1 E stretches high during stretched external accesses and remains low during non-visible internal accesses. This bit has no effect in single-chip modes.

#### 4.4.3.1.2 Normal Expanded Wide Mode

In expanded wide modes, Ports A and B are configured as a 16-bit multiplexed address and data bus and Port E bit 4 is configured as the E clock output signal. These signals allow external memory and peripheral devices to be interfaced to the MCU.

Port E pins other than PE4/ECLK are configured as general purpose I/O pins (initially high-impedance inputs with internal pull resistors enabled). Control bits PIPOE, NECLK, LSTRE, and RDWE in the PEAR register can be used to configure Port E pins to act as bus control outputs instead of general purpose I/O pins.

It is possible to enable the pipe status signals on Port E bits 6 and 5 by setting the PIPOE bit in PEAR, but it would be unusual to do so in this mode. Development systems where pipe status signals are monitored would typically use the special variation of this mode.

The Port E bit 2 pin can be reconfigured as the  $R/\overline{W}$  bus control signal by writing “1” to the RDWE bit in PEAR. If the expanded system includes external devices that can be written, such as RAM, the RDWE bit would need to be set before any attempt to write to an external location. If there are no writable resources in the external system, PE2 can be left as a general purpose I/O pin.

The Port E bit 3 pin can be reconfigured as the  $\overline{LSTRB}$  bus control signal by writing “1” to the LSTRE bit in PEAR. The default condition of this pin is a general purpose input because the  $\overline{LSTRB}$  function is not needed in all expanded wide applications.

The Port E bit 4 pin is initially configured as ECLK output with stretch. The E clock output function depends upon the settings of the NECLK bit in the PEAR register, the IVIS bit in the MODE register and the ESTR bit in the EBICTL register. The E clock is available for use in external select decode logic or as a constant speed clock for use in the external application system.

#### 4.4.3.1.3 Normal Expanded Narrow Mode

This mode is used for lower cost production systems that use 8-bit wide external EPROMs or RAMs. Such systems take extra bus cycles to access 16-bit locations but this may be preferred over the extra cost of additional external memory devices.

Ports A and B are configured as a 16-bit address bus and Port A is multiplexed with data. Internal visibility is not available in this mode because the internal cycles would need to be split into two 8-bit cycles.

Since the PEAR register can only be written one time in this mode, use care to set all bits to the desired states during the single allowed write.

The PE3/ $\overline{LSTRB}$  pin is always a general purpose I/O pin in normal expanded narrow mode. Although it is possible to write the LSTRE bit in PEAR to “1” in this mode, the state of LSTRE is overridden and Port E bit 3 cannot be reconfigured as the  $\overline{LSTRB}$  output.

It is possible to enable the pipe status signals on Port E bits 6 and 5 by setting the PIPOE bit in PEAR, but it would be unusual to do so in this mode. LSTRB would also be needed to fully understand system activity. Development systems where pipe status signals are monitored would typically use special expanded wide mode or occasionally special expanded narrow mode.

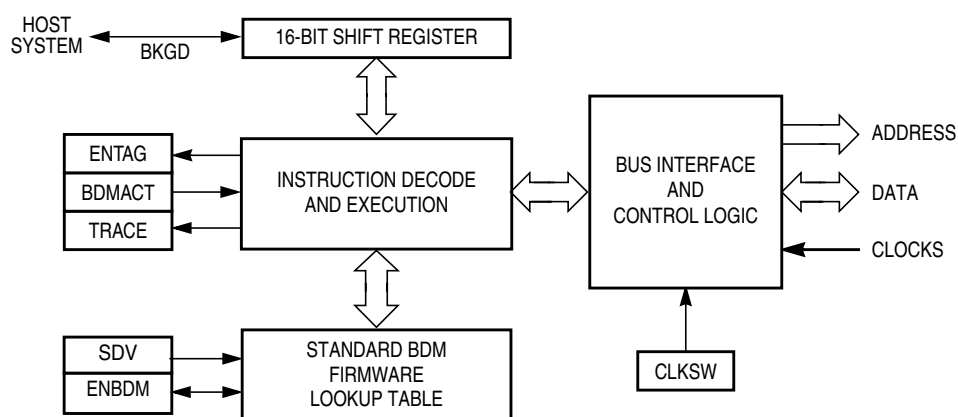
## Chapter 6

# Background Debug Module (BDMV4) Block Description

### 6.1 Introduction

This section describes the functionality of the background debug module (BDM) sub-block of the HCS12 core platform.

A block diagram of the BDM is shown in [Figure 6-1](#).



**Figure 6-1. BDM Block Diagram**

The background debug module (BDM) sub-block is a single-wire, background debug system implemented in on-chip hardware for minimal CPU intervention. All interfacing with the BDM is done via the BKGD pin.

BDMV4 has enhanced capability for maintaining synchronization between the target and host while allowing more flexibility in clock rates. This includes a sync signal to show the clock rate and a handshake signal to indicate when an operation is complete. The system is backwards compatible with older external interfaces.

#### 6.1.1 Features

- Single-wire communication with host development system
- BDMV4 (and BDM2): Enhanced capability for allowing more flexibility in clock rates
- BDMV4: SYNC command to determine communication rate
- BDMV4: GO\_UNTIL command
- BDMV4: Hardware handshake protocol to increase the performance of the serial communication
- Active out of reset in special single-chip mode

## 6.4 Functional Description

The BDM receives and executes commands from a host via a single wire serial interface. There are two types of BDM commands, namely, hardware commands and firmware commands.

Hardware commands are used to read and write target system memory locations and to enter active background debug mode, see [Section 6.4.3, “BDM Hardware Commands.”](#) Target system memory includes all memory that is accessible by the CPU.

Firmware commands are used to read and write CPU resources and to exit from active background debug mode, see [Section 6.4.4, “Standard BDM Firmware Commands.”](#) The CPU resources referred to are the accumulator (D), X index register (X), Y index register (Y), stack pointer (SP), and program counter (PC).

Hardware commands can be executed at any time and in any mode excluding a few exceptions as highlighted, see [Section 6.4.3, “BDM Hardware Commands.”](#) Firmware commands can only be executed when the system is in active background debug mode (BDM).

### 6.4.1 Security

If the user resets into special single-chip mode with the system secured, a secured mode BDM firmware lookup table is brought into the map overlapping a portion of the standard BDM firmware lookup table. The secure BDM firmware verifies that the on-chip EEPROM and FLASH EEPROM are erased. This being the case, the UNSEC bit will get set. The BDM program jumps to the start of the standard BDM firmware and the secured mode BDM firmware is turned off and all BDM commands are allowed. If the EEPROM or FLASH do not verify as erased, the BDM firmware sets the ENBDM bit, without asserting UNSEC, and the firmware enters a loop. This causes the BDM hardware commands to become enabled, but does not enable the firmware commands. This allows the BDM hardware to be used to erase the EEPROM and FLASH. After execution of the secure firmware, regardless of the results of the erase tests, the CPU registers, INITEE and PPAGE, will no longer be in their reset state.

### 6.4.2 Enabling and Activating BDM

The system must be in active BDM to execute standard BDM firmware commands. BDM can be activated only after being enabled. BDM is enabled by setting the ENBDM bit in the BDM status (BDMSTS) register. The ENBDM bit is set by writing to the BDM status (BDMSTS) register, via the single-wire interface, using a hardware command such as WRITE\_BD\_BYTE.

After being enabled, BDM is activated by one of the following<sup>1</sup>:

- Hardware BACKGROUND command
- BDM external instruction tagging mechanism
- CPU BGND instruction
- Breakpoint sub-block’s force or tag mechanism<sup>2</sup>

When BDM is activated, the CPU finishes executing the current instruction and then begins executing the firmware in the standard BDM firmware lookup table. When BDM is activated by the breakpoint sub-

1. BDM is enabled and active immediately out of special single-chip reset.

2. This method is only available on systems that have a breakpoint or a debug sub-block.

### 9.3.2.6 CRG Clock Select Register (CLKSEL)

This register controls CRG clock selection. Refer to [Figure 9-17](#) for details on the effect of each bit.

Module Base + 0x0005

	7	6	5	4	3	2	1	0
R	PLLSEL	PSTP	SYSWAI	ROAWAI	PLLWAI	CWAI	RTIWAI	COPWAI
W								
Reset	0	0	0	0	0	0	0	0

**Figure 9-9. CRG Clock Select Register (CLKSEL)**

Read: anytime

Write: refer to each bit for individual write conditions

**Table 9-4. CLKSEL Field Descriptions**

Field	Description
7 PLLSEL	<p><b>PLL Select Bit</b> — Write anytime. Writing a 1 when LOCK = 0 and AUTO = 1, or TRACK = 0 and AUTO = 0 has no effect. This prevents the selection of an unstable PLLCLK as SYSCLK. PLLSEL bit is cleared when the MCU enters self-clock mode, stop mode or wait mode with PLLWAI bit set.</p> <p>0 System clocks are derived from OSCCLK (Bus Clock = OSCCLK / 2).</p> <p>1 System clocks are derived from PLLCLK (Bus Clock = PLLCLK / 2).</p>
6 PSTP	<p><b>Pseudo-Stop Bit</b> — Write: anytime — This bit controls the functionality of the oscillator during stop mode.</p> <p>0 Oscillator is disabled in stop mode.</p> <p>1 Oscillator continues to run in stop mode (pseudo-stop). The oscillator amplitude is reduced. Refer to oscillator block description for availability of a reduced oscillator amplitude.</p> <p><b>Note:</b> Pseudo-stop allows for faster stop recovery and reduces the mechanical stress and aging of the resonator in case of frequent stop conditions at the expense of a slightly increased power consumption.</p> <p><b>Note:</b> Lower oscillator amplitude exhibits lower power consumption but could have adverse effects during any electro-magnetic susceptibility (EMS) tests.</p>
5 SYSWAI	<p><b>System Clocks Stop in Wait Mode Bit</b> — Write: anytime</p> <p>0 In wait mode, the system clocks continue to run.</p> <p>1 In wait mode, the system clocks stop.</p> <p><b>Note:</b> RTI and COP are not affected by SYSWAI bit.</p>
4 ROAWAI	<p><b>Reduced Oscillator Amplitude in Wait Mode Bit</b> — Write: anytime — Refer to oscillator block description chapter for availability of a reduced oscillator amplitude. If no such feature exists in the oscillator block then setting this bit to 1 will not have any effect on power consumption.</p> <p>0 Normal oscillator amplitude in wait mode.</p> <p>1 Reduced oscillator amplitude in wait mode.</p> <p><b>Note:</b> Lower oscillator amplitude exhibits lower power consumption but could have adverse effects during any electro-magnetic susceptibility (EMS) tests.</p>
3 PLLWAI	<p><b>PLL Stops in Wait Mode Bit</b> — Write: anytime — If PLLWAI is set, the CRGV4 will clear the PLLSEL bit before entering wait mode. The PLLON bit remains set during wait mode but the PLL is powered down. Upon exiting wait mode, the PLLSEL bit has to be set manually if PLL clock is required.</p> <p>While the PLLWAI bit is set the AUTO bit is set to 1 in order to allow the PLL to automatically lock on the selected target frequency after exiting wait mode.</p> <p>0 PLL keeps running in wait mode.</p> <p>1 PLL stops in wait mode.</p>



Module Base + 0x00X3

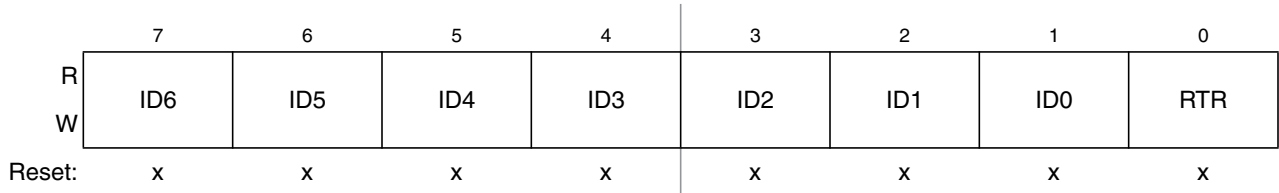


Figure 10-28. Identifier Register 3 (IDR3) — Extended Identifier Mapping

Table 10-27. IDR3 Register Field Descriptions — Extended

Field	Description
7:1 ID[6:0]	<b>Extended Format Identifier</b> — The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number.
0 RTR	<b>Remote Transmission Request</b> — This flag reflects the status of the remote transmission request bit in the CAN frame. In the case of a receive buffer, it indicates the status of the received frame and supports the transmission of an answering frame in software. In the case of a transmit buffer, this flag defines the setting of the RTR bit to be sent. 0 Data frame 1 Remote frame

### 10.3.3.1.2 IDR0–IDR3 for Standard Identifier Mapping

Module Base + 0x00X0

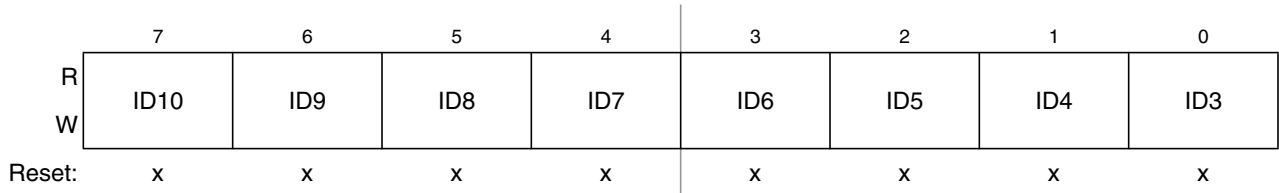


Figure 10-29. Identifier Register 0 — Standard Mapping

Table 10-28. IDR0 Register Field Descriptions — Standard

Field	Description
7:0 ID[10:3]	<b>Standard Format Identifier</b> — The identifiers consist of 11 bits (ID[10:0]) for the standard format. ID10 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number. See also ID bits in <a href="#">Table 10-29</a> .

**Table 12-9. PWMCTL Field Descriptions**

Field	Description
6 CON45	<b>Concatenate Channels 4 and 5</b> 0 Channels 4 and 5 are separate 8-bit PWMs. 1 Channels 4 and 5 are concatenated to create one 16-bit PWM channel. Channel 4 becomes the high-order byte and channel 5 becomes the low-order byte. Channel 5 output pin is used as the output for this 16-bit PWM (bit 5 of port PWMP). Channel 5 clock select control bit determines the clock source, channel 5 polarity bit determines the polarity, channel 5 enable bit enables the output and channel 5 center aligned enable bit determines the output mode.
5 CON23	<b>Concatenate Channels 2 and 3</b> 0 Channels 2 and 3 are separate 8-bit PWMs. 1 Channels 2 and 3 are concatenated to create one 16-bit PWM channel. Channel 2 becomes the high-order byte and channel 3 becomes the low-order byte. Channel 3 output pin is used as the output for this 16-bit PWM (bit 3 of port PWMP). Channel 3 clock select control bit determines the clock source, channel 3 polarity bit determines the polarity, channel 3 enable bit enables the output and channel 3 center aligned enable bit determines the output mode.
4 CON01	<b>Concatenate Channels 0 and 1</b> 0 Channels 0 and 1 are separate 8-bit PWMs. 1 Channels 0 and 1 are concatenated to create one 16-bit PWM channel. Channel 0 becomes the high-order byte and channel 1 becomes the low-order byte. Channel 1 output pin is used as the output for this 16-bit PWM (bit 1 of port PWMP). Channel 1 clock select control bit determines the clock source, channel 1 polarity bit determines the polarity, channel 1 enable bit enables the output and channel 1 center aligned enable bit determines the output mode.
3 PSWAI	<b>PWM Stops in Wait Mode</b> — Enabling this bit allows for lower power consumption in wait mode by disabling the input clock to the prescaler. 0 Allow the clock to the prescaler to continue while in wait mode. 1 Stop the input clock to the prescaler whenever the MCU is in wait mode.
2 PFRZ	<b>PWM Counters Stop in Freeze Mode</b> — In freeze mode, there is an option to disable the input clock to the prescaler by setting the PFRZ bit in the PWMCTL register. If this bit is set, whenever the MCU is in freeze mode the input clock to the prescaler is disabled. This feature is useful during emulation as it allows the PWM function to be suspended. In this way, the counters of the PWM can be stopped while in freeze mode so that after normal program flow is continued, the counters are re-enabled to simulate real-time operations. Because the registers remain accessible in this mode, to re-enable the prescaler clock, either disable the PFRZ bit or exit freeze mode. 0 Allow PWM to continue while in freeze mode. 1 Disable PWM input clock to the prescaler whenever the part is in freeze mode. This is useful for emulation.

With the misaligned character shown in Figure 13-20, the receiver counts 167 RTr cycles at the point when the count of the transmitting device is 10 bit times  $\times$  16 RTt cycles = 160 RTt cycles.

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

$$((167 - 160) / 167) \times 100 = 4.19\%$$

### 13.4.4.5.2 Fast Data Tolerance

Figure 13-21 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.

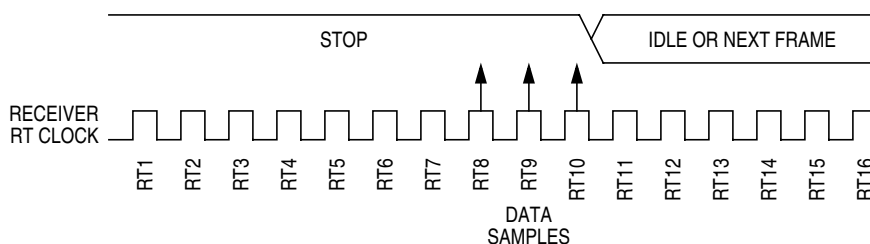


Figure 13-21. Fast Data

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

With the misaligned character shown in Figure 13-21, the receiver counts 154 RTr cycles at the point when the count of the transmitting device is 10 bit times  $\times$  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:

$$((160 - 154) / 160) \times 100 = 3.75\%$$

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

With the misaligned character shown in Figure 13-21, the receiver counts 170 RTr cycles at the point when the count of the transmitting device is 11 bit times  $\times$  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:

$$((176 - 170) / 176) \times 100 = 3.40\%$$

### 13.4.4.6 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.

Table 15-2. TIM16B8CV1 Memory Map

Address Offset	Use	Access
0x0000	Timer Input Capture/Output Compare Select (TIOS)	R/W
0x0001	Timer Compare Force Register (CFORC)	R/W <sup>(1)</sup>
0x0002	Output Compare 7 Mask Register (OC7M)	R/W
0x0003	Output Compare 7 Data Register (OC7D)	R/W
0x0004	Timer Count Register (TCNT(hi))	R/W <sup>(2)</sup>
0x0005	Timer Count Register (TCNT(lo))	R/W <sup>2</sup>
0x0006	Timer System Control Register1 (TSCR1)	R/W
0x0007	Timer Toggle Overflow Register (TTOV)	R/W
0x0008	Timer Control Register1 (TCTL1)	R/W
0x0009	Timer Control Register2 (TCTL2)	R/W
0x000A	Timer Control Register3 (TCTL3)	R/W
0x000B	Timer Control Register4 (TCTL4)	R/W
0x000C	Timer Interrupt Enable Register (TIE)	R/W
0x000D	Timer System Control Register2 (TSCR2)	R/W
0x000E	Main Timer Interrupt Flag1 (TFLG1)	R/W
0x000F	Main Timer Interrupt Flag2 (TFLG2)	R/W
0x0010	Timer Input Capture/Output Compare Register 0 (TC0(hi))	R/W <sup>(3)</sup>
0x0011	Timer Input Capture/Output Compare Register 0 (TC0(lo))	R/W <sup>3</sup>
0x0012	Timer Input Capture/Output Compare Register 1 (TC1(hi))	R/W <sup>3</sup>
0x0013	Timer Input Capture/Output Compare Register 1 (TC1(lo))	R/W <sup>3</sup>
0x0014	Timer Input Capture/Output Compare Register 2 (TC2(hi))	R/W <sup>3</sup>
0x0015	Timer Input Capture/Output Compare Register 2 (TC2(lo))	R/W <sup>3</sup>
0x0016	Timer Input Capture/Output Compare Register 3 (TC3(hi))	R/W <sup>3</sup>
0x0017	Timer Input Capture/Output Compare Register 3 (TC3(lo))	R/W <sup>3</sup>
0x0018	Timer Input Capture/Output Compare Register4 (TC4(hi))	R/W <sup>3</sup>
0x0019	Timer Input Capture/Output Compare Register 4 (TC4(lo))	R/W <sup>3</sup>
0x001A	Timer Input Capture/Output Compare Register 5 (TC5(hi))	R/W <sup>3</sup>
0x001B	Timer Input Capture/Output Compare Register 5 (TC5(lo))	R/W <sup>3</sup>
0x001C	Timer Input Capture/Output Compare Register 6 (TC6(hi))	R/W <sup>3</sup>
0x001D	Timer Input Capture/Output Compare Register 6 (TC6(lo))	R/W <sup>3</sup>
0x001E	Timer Input Capture/Output Compare Register 7 (TC7(hi))	R/W <sup>3</sup>
0x001F	Timer Input Capture/Output Compare Register 7 (TC7(lo))	R/W <sup>3</sup>
0x0020	16-Bit Pulse Accumulator Control Register (PACTL)	R/W
0x0021	Pulse Accumulator Flag Register (PAFLG)	R/W
0x0022	Pulse Accumulator Count Register (PACNT(hi))	R/W
0x0023	Pulse Accumulator Count Register (PACNT(lo))	R/W
0x0024 – 0x002C	Reserved	— <sup>(4)</sup>
0x002D	Timer Test Register (TIMTST)	R/W <sup>2</sup>
0x002E – 0x002F	Reserved	— <sup>4</sup>

1. Always read 0x0000.

2. Only writable in special modes (test\_mode = 1).

3. Write to these registers have no meaning or effect during input capture.

4. Write has no effect; return 0 on read

### 15.6.2 Pulse Accumulator Input Interrupt (PAOVI)

This active high output will be asserted by the module to request a timer pulse accumulator input interrupt to be serviced by the system controller.

### 15.6.3 Pulse Accumulator Overflow Interrupt (PAOVF)

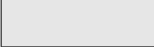
This active high output will be asserted by the module to request a timer pulse accumulator overflow interrupt to be serviced by the system controller.

### 15.6.4 Timer Overflow Interrupt (TOF)

This active high output will be asserted by the module to request a timer overflow interrupt to be serviced by the system controller.

Module Base + 0x000F

	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 17-20. RESERVED6**

All bits read 0 and are not writable.

## 17.4 Functional Description

### 17.4.1 Flash Command Operations

Write operations are used for the program, erase, and erase verify algorithms described in this section. The program and erase algorithms are controlled by a state machine whose timebase FCLK is derived from the oscillator clock via a programmable divider. The FCMD register as well as the associated FADDR and FDATA registers operate as a buffer and a register (2-stage FIFO) so that a new command along with the necessary data and address can be stored to the buffer while the previous command is still in progress. This pipelined operation allows a time optimization when programming more than one word on a specific row, as the high voltage generation can be kept active in between two programming commands. The pipelined operation also allows a simplification of command launching. Buffer empty as well as command completion are signalled by flags in the FSTAT register with corresponding interrupts generated, if enabled.

The next sections describe:

- How to write the FCLKDIV register
- Command write sequence used to program, erase or erase verify the Flash array
- Valid Flash commands
- Errors resulting from illegal Flash operations

#### 17.4.1.1 Writing the FCLKDIV Register

Prior to issuing any Flash command after a reset, it is first necessary to write the FCLKDIV register to divide the oscillator clock down to within the 150-kHz to 200-kHz range. Since the program and erase timings are also a function of the bus clock, the FCLKDIV determination must take this information into account.

If we define:

- FCLK as the clock of the Flash timing control block
- Tbus as the period of the bus clock
- INT(x) as taking the integer part of x (e.g., INT(4.323) = 4),

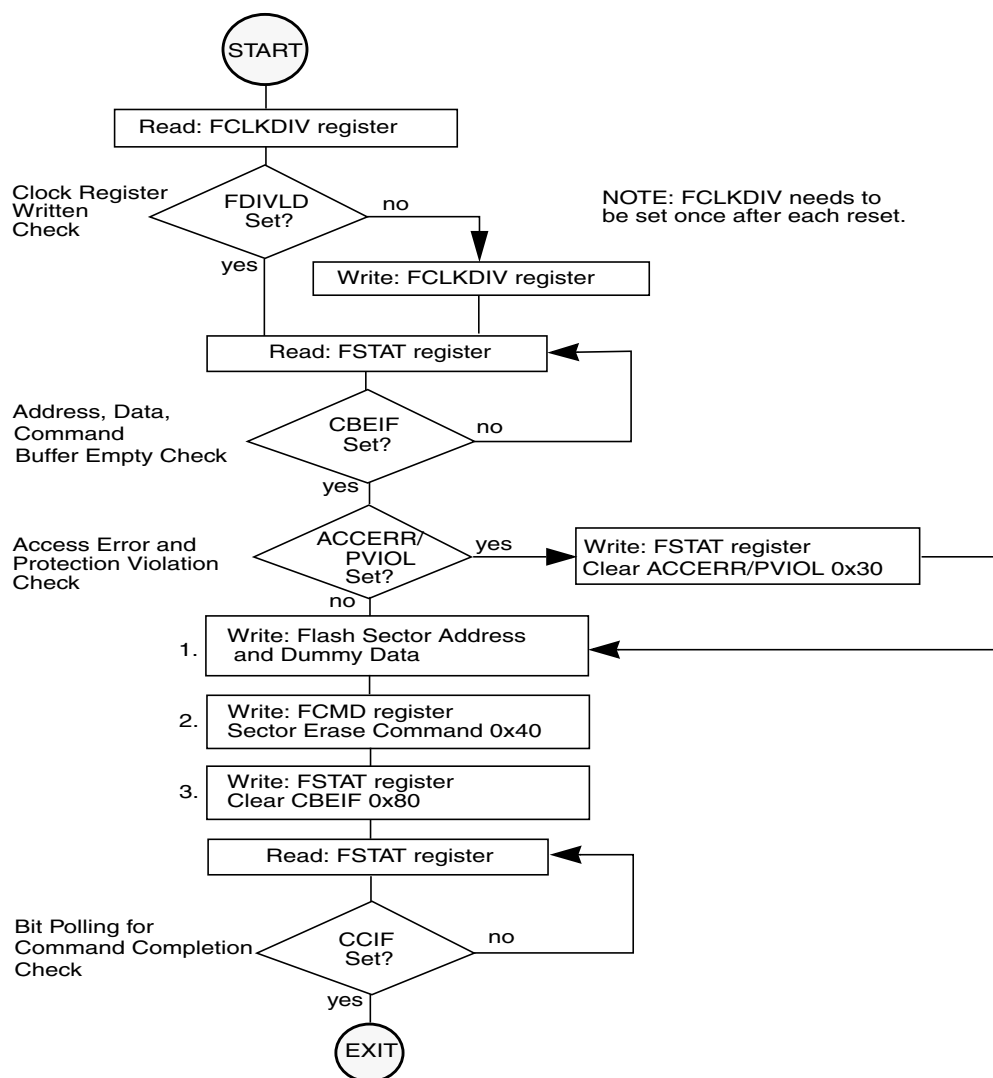


Figure 17-24. Example Sector Erase Command Flow

**Table 18-4. FSEC Field Descriptions**

Field	Description
7–6 KEYEN[1:0]	<b>Backdoor Key Security Enable Bits</b> — The KEYEN[1:0] bits define the enabling of the backdoor key access to the Flash module as shown in <a href="#">Table 18-5</a> .
5–2 NV[5:2]	<b>Nonvolatile Flag Bits</b> — The NV[5:2] bits are available to the user as nonvolatile flags.
1–0 SEC[1:0]	<b>Flash Security Bits</b> — The SEC[1:0] bits define the security state of the MCU as shown in <a href="#">Table 18-6</a> . If the Flash module is unsecured using backdoor key access, the SEC[1:0] bits are forced to 1:0.

**Table 18-5. Flash KEYEN States**

KEYEN[1:0]	Status of Backdoor Key Access
00	DISABLED
01 <sup>(1)</sup>	DISABLED
10	ENABLED
11	DISABLED

1. Preferred KEYEN state to disable Backdoor Key Access.

**Table 18-6. Flash Security States**

SEC[1:0]	Status of Security
00	Secured
01 <sup>(1)</sup>	Secured
10	Unsecured
11	Secured

1. Preferred SEC state to set MCU to secured state.


The security function in the Flash module is described in [Section 18.4.3, “Flash Module Security”](#).

### 18.3.2.3 RESERVED1

This register is reserved for factory testing and is not accessible to the user.

Module Base + 0x0002

	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 18-6. RESERVED1**

All bits read 0 and are not writable.



### 19.4.1.3 Valid Flash Commands

Table 19-17 summarizes the valid Flash commands along with the effects of the commands on the Flash array.

**Table 19-17. Valid Flash Commands**

FCMD	Meaning	Function on Flash Array
0x05	Erase Verify	Verify all bytes in the Flash array are erased. If the Flash array is erased, the BLANK bit will set in the FSTAT register upon command completion.
0x20	Program	Program a word (2 bytes) in the Flash array.
0x40	Sector Erase	Erase all 1024 bytes in a sector of the Flash array.
0x41	Mass Erase	Erase all bytes in the Flash array. A mass erase of the full Flash array is only possible when FPLDIS, FPHDIS, and FPOPEN bits in the FPROT register are set prior to launching the command.

#### CAUTION

A Flash word must be in the erased state before being programmed.  
Cumulative programming of bits within a Flash word is not allowed.

## 19.4.4 Flash Reset Sequence

On each reset, the Flash module executes a reset sequence to hold CPU activity while loading the following registers from the Flash array memory according to [Table 19-1](#):

- FPROT — Flash Protection Register (see [Section 19.3.2.5](#))
- FSEC — Flash Security Register (see [Section 19.3.2.2](#))

### 19.4.4.1 Reset While Flash Command Active

If a reset occurs while any Flash command is in progress, that command will be immediately aborted. The state of the word being programmed or the sector/array being erased is not guaranteed.

## 19.4.5 Interrupts

The Flash module can generate an interrupt when all Flash commands have completed execution or the Flash address, data, and command buffers are empty.

**Table 19-18. Flash Interrupt Sources**

Interrupt Source	Interrupt Flag	Local Enable	Global (CCR) Mask
Flash Address, Data, and Command Buffers are empty	CBEIF (FSTAT register)	CBEIE	I Bit
All Flash commands have completed execution	CCIF (FSTAT register)	CCIE	I Bit

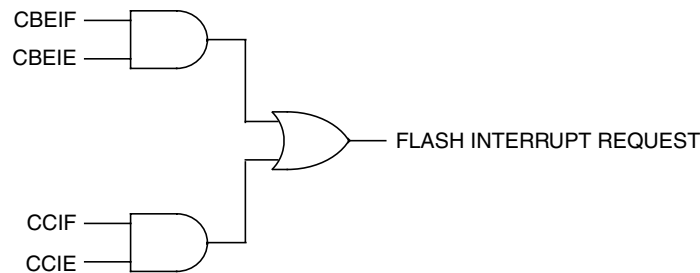
### NOTE

Vector addresses and their relative interrupt priority are determined at the MCU level.

### 19.4.5.1 Description of Interrupt Operation

[Figure 19-29](#) shows the logic used for generating interrupts.

The Flash module uses the CBEIF and CCIF flags in combination with the enable bits CBIE and CCIE to discriminate for the generation of interrupts.



**Figure 19-29. Flash Interrupt Implementation**

For a detailed description of these register bits, refer to [Section 19.3.2.4](#), “Flash Configuration Register (FCNFG)” and [Section 19.3.2.6](#), “Flash Status Register (FSTAT)”.

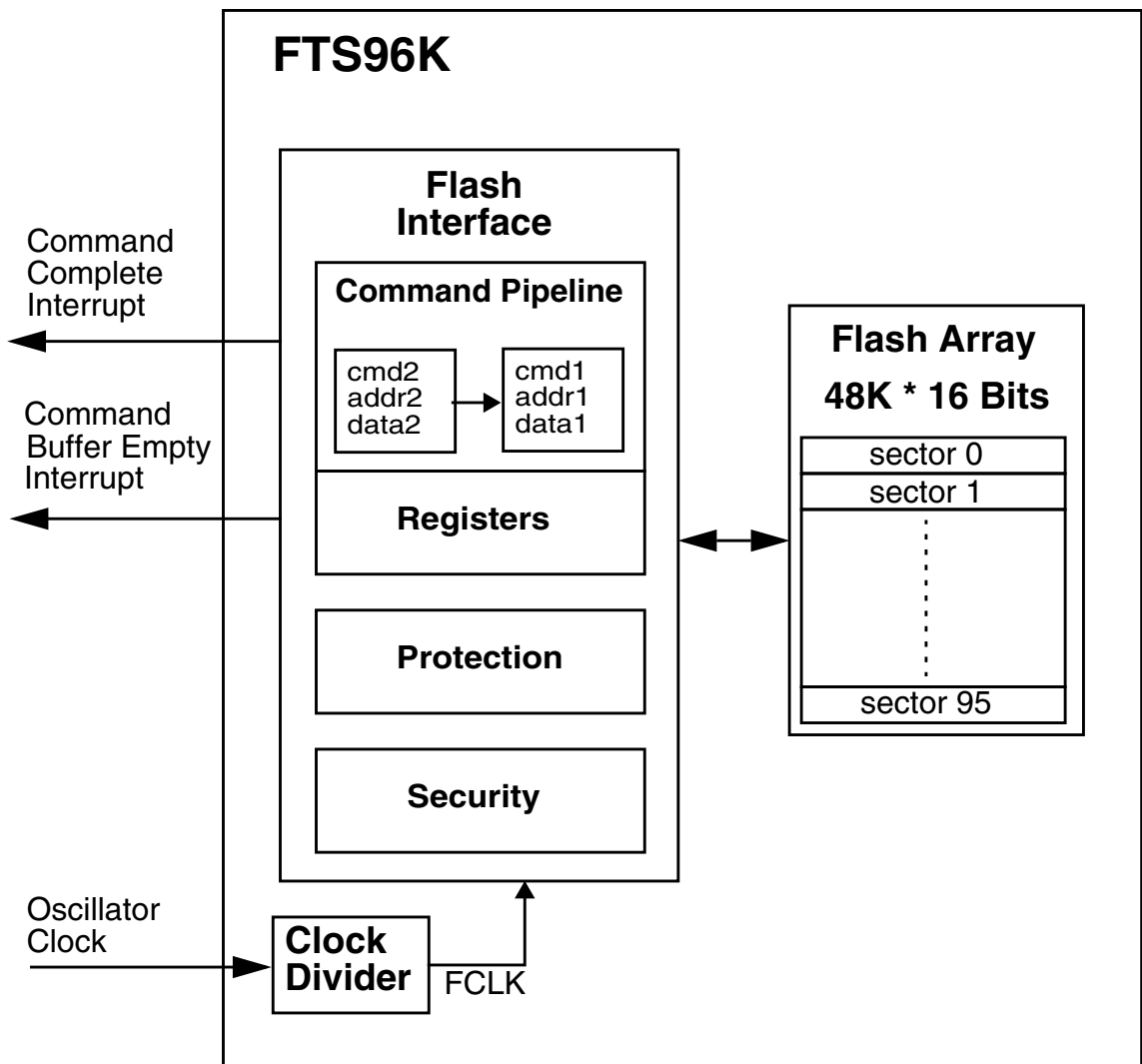


Figure 20-2. FTS96K Block Diagram

## 20.2 External Signal Description

The [FTS128K1FTS96K](#) module contains no signals that connect off-chip.

## 20.3 Memory Map and Registers

This section describes the [FTS128K1FTS96K](#) memory map and registers.

### 20.3.1 Module Memory Map

The [FTS128K1FTS96K](#) memory map is shown in [Figure 20-3](#)[Figure 20-4](#). The HCS12 architecture places the Flash array addresses between [0x40000x4000](#) and [0xFFFF](#), which corresponds to three 16 Kbyte [pages](#). The content of the HCS12 Core PPAGE register is used to map the logical [middle](#) page ranging from

## A.2.2 ATD Operating Characteristics In 3.3V Range

The [Table A-11](#) shows conditions under which the ATD operates.

The following constraints exist to obtain full-scale, full range results:  $V_{SSA} \leq V_{RL} \leq V_{IN} \leq V_{RH} \leq V_{DDA}$ . This constraint exists since the sample buffer amplifier can not drive beyond the power supply levels that it ties to. If the input level goes outside of this range it will effectively be clipped

**Table A-11. ATD Operating Characteristics**

Conditions are shown in <a href="#">Table A-4</a> unless otherwise noted; Supply Voltage $3.3V-10\% \leq V_{DDA} \leq 3.3V+10\%$							
Num	C	Rating	Symbol	Min	Typ	Max	Unit
1	D	Reference Potential Low High	$V_{RL}$ $V_{RH}$	$V_{SSA}$ $V_{DDA}/2$	— —	$V_{DDA}/2$ $V_{DDA}$	V V
2	C	Differential Reference Voltage	$V_{RH}-V_{RL}$	3.0	3.3	3.6	V
3	D	ATD Clock Frequency	$f_{ATDCLK}$	0.5	—	2.0	MHz
4	D	ATD 10-Bit Conversion Period Clock Cycles <sup>(1)</sup> Conv, Time at 2.0MHz ATD Clock $f_{ATDCLK}$	$N_{CONV10}$ $T_{CONV10}$	14 7	— —	28 14	Cycles $\mu s$
5	D	ATD 8-Bit Conversion Period Clock Cycles <sup>1</sup> Conv, Time at 2.0MHz ATD Clock $f_{ATDCLK}$	$N_{CONV8}$ $T_{CONV8}$	12 6	— —	26 13	Cycles $\mu s$
6	D	Recovery Time ( $V_{DDA}=3.3$ Volts)	$t_{REC}$	—	—	20	$\mu s$
7	P	Reference Supply current	$I_{REF}$	—	—	0.250	mA

1. The minimum time assumes a final sample period of 2 ATD clocks cycles while the maximum time assumes a final sample period of 16 ATD clocks.

## A.2.3 Factors Influencing Accuracy

Three factors — source resistance, source capacitance and current injection — have an influence on the accuracy of the ATD.

### A.2.3.1 Source Resistance

Due to the input pin leakage current as specified in [Table A-6](#) in conjunction with the source resistance there will be a voltage drop from the signal source to the ATD input. The maximum source resistance  $R_S$  specifies results in an error of less than 1/2 LSB (2.5mV) at the maximum leakage current. If device or operating conditions are less than worst case or leakage-induced error is acceptable, larger values of source resistance is allowable.

### A.2.3.2 Source Capacitance

When sampling an additional internal capacitor is switched to the input. This can cause a voltage drop due to charge sharing with the external and the pin capacitance. For a maximum sampling error of the input voltage  $\leq 1\text{LSB}$ , then the external filter capacitor,  $C_f \geq 1024 * (C_{INS} - C_{INN})$ .

## C.1.1 80-Pin QFP Package

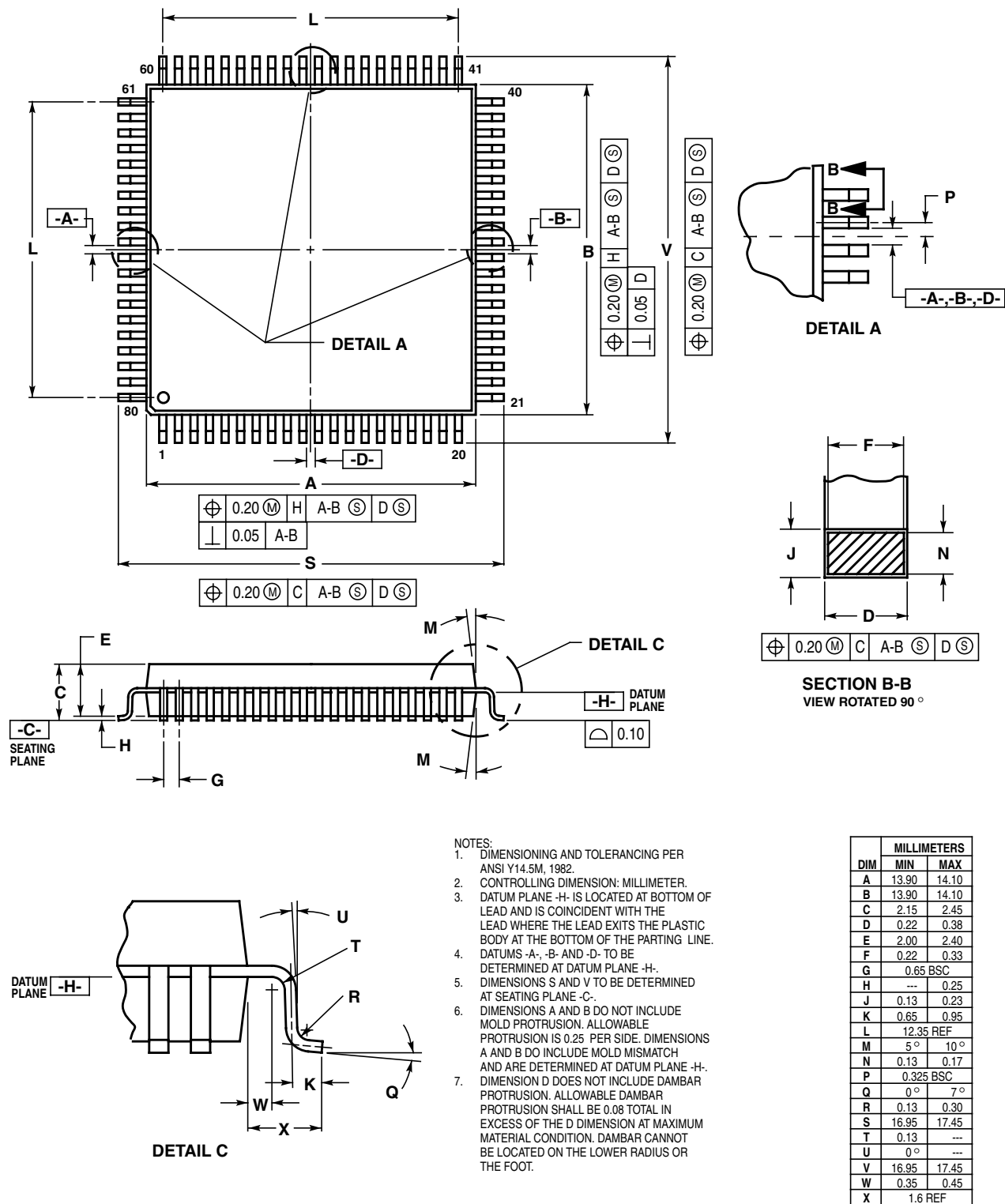


Figure C-1. 80-Pin QFP Mechanical Dimensions (Case no. 841B)