TRANSPORTATION SYSTEMS GROUP
MASK SET ERRATA AND INFORMATION SHEET
Part: HC12BE32.00J38M.A Mask Set:
Report Generated: Aug 06, 99 02:00

HC12BE32.00J38M.A Modules
Current Module Revision
ATD8B8C.2.10
BDLC.1.5
BDM256.3.0
BKP.2.2
CORNER.2.6
CPU.3.6
ECT16B8C.1.1
EE768.2.2
INT.4.1
IOB.2.4
LIM_BE32.1.1
MEBI_B32.3.2
MMI_BE32.1.0
MSI1C1P.2.3
PADS_BE32.1.0
PWM8B4C.4.0
RAM1K.1.6
ROC.3.4
ROM32K.1.0
VDD.3.3
VDDA.3.3
VDDX.3.3
VPP.3.2
VREF.1.2
VSS.3.2
VSSA.3.2
VSSX.2.2
WCR.2.1
XTAL.2.6

HC12_AR_505 Customer Erratum HC12BE32.00J38M.A

DESCRIPTION:
Device is marginal after 100V machine model ESD zapping

WORKAROUND:
Extra care must be taken when handling this device.

H**C12_AR_374 Customer Erratum ATD8B8C.2.10**

DESCRIPTION:
ATD status bits and conversion counter are not reset properly when writing any ATD register while an active A/D conversion sequence is in the process of completion.

WORKAROUND:
When starting a new sequence, perform two writes to control registers 4/5 in quick secession. If the first write occurs when the status bit/conversion counter is not reset, the second write will correct ATD operation.

HC12_AR_311 Customer Information ATD8B8C.2.10

DESCRIPTION:
The (VRH-VRL)/2 internal reference conversion result may be $7F, $80 or $81.

WORKAROUND:
If the (VRH-VRL)/2 internal reference is used (perhaps for system diagnostics), expected pass result may be $7F, $80 or $81.

HC12_AR_493 Customer Erratum BDLC.1.5

DESCRIPTION:
To transmit a message using the BDLC, the user writes the first byte of the message to be transmitted into the BDLC data register (BDR). This will initiate the transmission process at the beginning of the next idle bus state.An invalid symbol being received by the BDLC clears any byte that had been previously written to the BDR. This will inhibit the transmission process until the user writes another byte to the BDR. The following scenario describes an event sequence that would prevent the user from knowing that an invalid symbol was received and that the BDR had been cleared.

WORKAROUND:
A two level strategy has been developed that positively signals the need to restart the transmission of a message. The first level looks for the special case of reception of an illegal symbol with a byte pending transmission in the BDLC data register, as described above. The second level uses a transmit watchdog timer to spot any case of a transmission not occurring within a maximum amount of time.1a) If an illegal symbol interrupt occurs with a byte pending transmission in the BDR, reload the BDR with the first byte of that message to restart transmission.

HC12_AR_519 Customer Erratum BDLC.1.5

DESCRIPTION:
CRC error in received message does not prevent IFR transmission.

WORKAROUND:
In response to the CRC Error interrupt ($18 in BSVR), immediately write an $FF byte into the BDR, and then clear the lower four bits in BCR2 (TSIFR, TMIFR0, TMIFR1, TEOD). This will cancel the IFR transmission.

HC12_AR_520 Customer Erratum BDLC.1.5

DESCRIPTION:
When programmed to transmit a single byte Type 2 IFR (byte loaded in the BDR and TSIFR bit set in BCR2), the BDLC will attempt to transmit a single IFR byte following the EOD of the message currently being received. If the byte to be transmitted loses arbitration, the BDLC will continue to retry transmission until it is successful or an error occurs or the TEOD bit is set.

WORKAROUND:
(Short form) When the first RXIFR interrupt occurs the requester message has been received without errors (invalid symbol or CRC). When the BDLC receives back correctly the IFR byte is was trying to transmit, then response by this node is complete and the requester message received can be passed to the application layer. Ignore any invalid symbol error that occurs after the first RXIFR interrupt, since transmission of IFRs are not retried.

HC12_AR_463 Customer Erratum BKP.2.2

DESCRIPTION:
Breakpoint Address and Data registers are properly reset only upon Power on Reset.

WORKAROUND:
To insure BRKAH, BRKAL, BRKDH and BRKDL registers have the correct default values, always clear each immediately after reset.

HC12_AR_288 Customer Erratum CPU.3.6

DESCRIPTION:
If the REV or REVW instructions are interrupted while processing a rules list, the results from the instructions may be incorrect. The Condition Code Register and Index Register Y (weight pointer for REVW) may be incorrect when the stacking occurs for the interrupt. The REV and REVW instructions produce the wrong result after returning from the interrupt because the V-bit in the CC register and IY registers (REVW) may not hold the same state as prior to the interrupt.

WORKAROUND:
Disable the interrupts prior to using the REV or REVW instruction (which could increase the interrupt latency). If a non-maskable interrupt has been enabled, there is no workaround.

HC12_AR_528 Customer Erratum INT.4.1

DESCRIPTION:
When using an edge sensitive IRQ signal to trigger an interrupt service routine and the IRQ is not released during servicing, the part will not enter stop mode if the following executed instruction is STOP.

WORKAROUND:
When IRQ is set to be edge sensitive, release pin before executing STOP instruction.

HC12_AR_441 Customer Erratum INT.4.1

DESCRIPTION:
There is a cycle at the beginning of executing a BDM instruction that is susceptible to being interrupted directly after the background has executed. Since the interrupt source is masked, the interrupt vector that is requested is $FFF6. The BDM firmware at $FFF6 points to the routine at $FF24. The interrupt was stacked when it was taken, but the BDM routines do not un-stack (no RTI). When you return from BDM the stack pointer is pointing to the wrong place. Avoid using TRACE during cycles that allow interrupts to become unmasked (i.e. TRACE of a CLI if interrupts are pending). Caution should a

WORKAROUND:
None.

HC12_AR_527 Customer Information INT.4.1

DESCRIPTION:
If the source of an interrupt is taken away by disabling the interrupt without setting the I mask bit in the CCR, an SWI interrupt may be fetched instead of the vector for the interrupt source that was disabled.

WORKAROUND:
Before disabling an interrupt using a local interrupt control bit, set the I mask bit in the CCR.

HC12_AR_510 Customer Erratum IOB.2.4

DESCRIPTION:
Expanded mode operation causes an increase in bus driver shoot through current pin leakage which can cause inaccurate A/D conversions.

WORKAROUND:
Enabling reduced drive on ports A, B, and E while the HC12 is in expanded mode reduces the amount of VDD/VSS noise caused by bus driver shoot through current. Therefore, the A/D accuracy meets specified limits.

HC12_AR_333 Customer Information PWM8B4C.4.0

DESCRIPTION:
HC11 code for the PWM module is not directly portable to the HC12.The PWM Concatenation (16-bit) mode implementation is not compatible with the HC11 PWM.When operating in Concatenation mode, the HC11 PWM channel output pin and clock source are both controlled by the “low-order” byte.i.e. if concatenate channels 1(3) & 2(4) the output pin is channel 2(4) and the clock source for the concatenated counter is the clock source for channel 2(4).When operating in Concatenation mode, the HC12 PWM channel output pin is the pin associated with the “high-order” byte and the clock source is controlled by the “low-order” byte.i.e. if concatenate 01(23), then pin 0(2) is the output and channel 1(2) controls the clock source.

WORKAROUND:
Modify system configuration based on HC12 specification information.

HC12_AR_148 Customer Information PWM8B4C.4.0

DESCRIPTION:
HC11 code for the PWM module is not directly portable to the HC12.

WORKAROUND:
Modify all HC11 duty register values to be the desired duty value - 1 on the HC12.

HC12_AR_327 Customer Information PWM8B4C.4.0

DESCRIPTION:
The PWM scaled clock (S0,S1) equations are incorrect in the HC12 documentation. The incorrect equations are: Clock S0 = A / 2 * PWSCAL0Clock S1 = B / 2 * PWSCAL1

WORKAROUND:
The correct PWM scale clock (S0,S1) equations are: Clock S0 = A / 2 * (PWSCAL0 + 1)Clock S1 = B / 2 * (PWSCAL1 + 1)Therefore, programming $FF is full scale divide of 256.

HC12_AR_328 Customer Information PWM8B4C.4.0

DESCRIPTION:
HC11 code for the PWM module is not directly portable to the HC12.The PWM scaled clock (S0,S1) equations are not compatible with the HC11 PWM.The HC11 PWM scaled clock equations are: Clock S0 = A / 2 * PWSCAL0Clock S1 = B / 2 * PWSCAL1The HC12 PWM scaled clock equations are: Clock S0 = A / 2 * (PWSCAL0 + 1)Clock S1 = B / 2 * (PWSCAL1 + 1)

WORKAROUND:
Modify all HC12 PWM scaled clock (S0,S1) values to use the following equations: Clock S0 = A / 2 * (PWSCAL0 + 1)Clock S1 = B / 2 * (PWSCAL1 + 1)Therefore, programming $FF is full scale divide of 256.

HC12_AR_329 Customer Information PWM8B4C.4.0

DESCRIPTION:
The PWM Center-Aligned Mode Duty Cycle equations are incorrect in the HC12 documentation. The incorrect equations are: Center-Aligned-Output Mode: (center=1)Duty cycle = ((PWPERx - PWDTYx) / (PWPERx + 1)) X 100% for Polarity = 1Duty cycle = ((PWDTYx + 1) / (PWPERx + 1)) X 100% for Polarity = 0

WORKAROUND:
The correct PWM duty cycle equations are: Duty cycle = [(PWPERx - PWDTYx) / (PWPERx)] X 100% for Polarity = 0Duty cycle = {(PWDTYx) / (PWPERx)] X 100% for Polarity = 1

HC12_AR_330 Customer Information PWM8B4C.4.0

DESCRIPTION:
The PWM Period equations are incorrect in the HC12 documentation. The incorrect equations are: Period = (Channel-Clock-Period / (PWPER + 1)) for center = 0Period = (Channel-Clock-Period / (2 * (PWPER + 1))) for center = 1

WORKAROUND:
The correct PWM Period equations are: Period = [Channel-Clock-Period * (PWPER + 1)] for center = 0Period - [Channel-Clock-Period * (2 * PWPER)] for center = 1

HC12_AR_472 Customer Erratum ROC.3.4

DESCRIPTION:
When the device exits WAIT mode, it does not return to the correct location within the routine if the stack is positioned in external memory and if the stretch bits have been enabled to lengthen the clock.

WORKAROUND:
To overcome this problem, locate the stack in internal RAM resources and/or clear the stretch bits to prevent clock stretching.

HC12_AR_504 Customer Erratum ROC.3.4

DESCRIPTION:
When a clock monitor reset occurs, the crystal may start-up with no delay period. As a result, the MCU attempts to fetch reset vectors before the crystal has fully stabilized.

WORKAROUND:
When clock monitor failure has occurred, hold the reset signal until the crystal has reached stable oscillation.

HC12_AR_400 Customer Information ROC.3.4

DESCRIPTION:
When coming out of STOP by using reset, the crystal start-up delay time-out does not occur. This means the MCU may be attempting to run before the crystal has become stable.

WORKAROUND:
When coming out of STOP with reset hold the reset signal until the crystal has reached stable oscillation.

HC12_AR_502 Customer Erratum WCR.2.1

DESCRIPTION:
When a clock monitor reset occurs, the crystal may start-up with no delay period. As a result, the MCU attempts to fetch reset vectors before the crystal has fully stabilized.

WORKAROUND:
When a clock monitor failure has occurred, hold the reset signal until the crystal has reached stable oscillation. Since it is not generally possible to detect the cause of reset with external circuitry, you could design your external reset so it always holds the reset pin low long enough to allow the oscillator to stabilize.

HC12_AR_504 Customer Erratum WCR.2.1

DESCRIPTION:
When a clock monitor reset occurs, the crystal may start-up with no delay period. As a result, the MCU attempts to fetch reset vectors before the crystal has fully stabilized.

WORKAROUND:
When clock monitor failure has occurred, hold the reset signal until the crystal has reached stable oscillation.

Last updated: Wednesday, 18-Aug-1999 16:02:07 CDT