Mask Set Errata

MSE912DG128A_4K91D
Rev 0.0, 01/2003

Mask Set Errata for
68HC912DG128A
Mask 4K91D

MOTOROLA intelligence everywhere™ | digital dna™

Introduction

This mask set errata applies to the following MCU mask set:

- 4K91D

MCU Device Mask Set Identification

The mask set is identified by a 5-character code consisting of a version number, a letter, two numerical digits, and a letter, for example 0K51E. All standard devices are marked with a mask set number and a date code.

MCU Device Date Codes

Device markings indicate the week of manufacture and the mask set used. The date is coded as four numerical digits where the first two digits indicate the year and the last two digits indicate the work week. For instance, the date code “0201” indicates the first week of the year 2002.

MCU Device Part Number Prefixes

Some MCU samples and devices are marked with an SC, PC, or XC prefix. An SC prefix denotes special/custom device. A PC prefix indicates a prototype device which has undergone basic testing only. An XC prefix denotes that the device is tested but is not fully characterized or qualified over the full range of normal manufacturing process variations. After full characterization and qualification, devices will be marked with the MC or SC prefix.

MSE Published Date: 26 March 2003

Errata Summary

Errata number	Module affected	Description
AR_658	CGM	Race condition within Bus Clock Switcher Circuit
AR_659	ATD	Abort in last ATDCLK of sequence does not restart
AR_624	ATD	10-bit ATD converter accuracy is not within spec
AR_593	CGM	Operation with 16MHz quartz crystals is not recommended
AR_627	CGM	CPU does not start up correctly after switching to external crystal
AR_636	CGM	Limp home clock frequency is too high
AR_650	CGM	XIRQ during last cycle of STOP instruction causes run away
AR_638	WCR/ CGM	Exit from pseudo-stop in limp home
AR_644	ECT	PA Overflow flag not set when event is concurrent with write of $FFFF
AR_526	IIC	SCL divider has an extra clock at 8Mhz bus frequency
AR_573	IIC	IIC hold both SCL and SDA lines low when IBB bit is not busy
AR_548	IIC	Disabling IIC can glitch and corrupt IIC bus
AR_646	MSCAN	MSCAN extended ID rejected if stuff bit between ID16 and ID15

Race condition within Bus Clock Switcher Circuit — Errata Number: HC12_AR_658

Description

Under certain conditions it is possible for a race condition to occur within the Bus Clock Switcher Circuit, causing incorrect operation of the CPU. This race condition can occur when the SYSCLK is switched to the PLL clock before the PLL has achieved lock.

Workaround

Do not switch the system clock source to the PLL until the PLL has attained lock.

Abort in last ATDCLK of sequence does not restart — Errata Number: HC12_AR_659

Description

When writing ATDCTL4 and/or ATDCTL5 during an active conversion the write is considered an abort and restart. However, when writing during the last

ATDCLK of a sequence, the current conversion is aborted, but a new conversion is not started. This occurs whether the sequence is 1 or 4 or 8 conversions. Since writes to ATDCTL4 start a conversion then it is possible for successive byte writes to ATDCTL4/5 to result in this problem. This would occur if an IRQ service related to another interrupt source occurs, separating the two byte writes, and the RTI of this returns delaying the second write to occur in the last ATDCLK.

Workaround

The first aspect of the solution is to use word writes to ATDCTL4/5. This eliminates the possibility of other IRQ sources causing delay between writes to ATDCTL4/5. This would be the only solution required when starting the first conversion. It would also be the only solution needed when SCAN=0 if all further conversion sequences are initiated from an ATD interrupt routine. In addition, this is the only solution needed if code, in general, does not abort ongoing conversions.

The second aspect to the solution regards cases that abort conversions. The easiest solution is to toggle the S8C bit. This effectively cleans up the abort and the second write to the ATDCTL5 will perform a successful restart. Bracket this toggle sequence with SEI and CLI to prevent the second write from occurring during a last ATDCLK of a sequence.

Another method is possible using dual writes to start a conversion with a minimum of one ATDCLK period between the writes. This effectively allows the first write to abort and flush by the next write which would start (or restart) the conversion. The second write also needs to occur before another sequence complete time elapses. This method should also be prefixed by a SEI and followed by a CLI. This would prevent the case of other IRQ sources causing the same problem as well.

10-bit ATD converter accuracy is not within spec — Errata Number: HC12_AR_624

Description

When running the ATD module in 10-bit resolution mode, worst-case absolute error was found to be 5 counts at hot (125°C) temperature. Worst-case 10-bit absolute error was found to be 8 counts at room (25°C) and at cold (–40°C) temperatures. In 8- bit resolution mode, worst-case absolute error was found to be 2 counts at all temperatures.

Workaround

ATD1 was found to have slightly better overall accuracy than ATD0. Also, better accuracy was observed when running the ATD module with a higher prescaler (PRS4-0) value of 00010 (ATDCLK = 1.33MHz, bus speed = 8MHz) when compared to the accuracy with ATDCLK = 2MHz. These values can be modified by writing to register ATDCTL4.

Operation with 16MHz quartz crystals is not recommended

Errata Number: HC12_AR_593

Description

Interaction of the resonator and microcontroller characteristics can result in a small proportion of applications failing to start up and stabilize correctly even though typical product combinations work well under test conditions. Resonator operation should be restricted to maximum 10 MHz

Workaround

1. Use 10 MHz (or slower) resonators and generate higher bus frequencies using the PLL module. Note: When using 10 MHz or slower resonators proper and robust operation of the oscillator circuit requires close attention to board layout to ensure correct gain margin and negative resistance margin. There is a well documented analysis technique performed to measure Negative Resistance Margin which indicates the margin for stable oscillation of the combined microcontroller and resonator. However, an alternative approach is to include gain margin analysis. Since a negative resistance margin optimization cannot include all process, temperature, and voltage variance of the microcontroller, it is possible that the components chosen for the optimum negative resistance point may not yield acceptable component values for gain margin. In this case a compromise between the negative resistance margin and gain margin is desired. However option 2 (below) may be necessary should this remain unachievable.
2. The EXTAL pin input accepts frequencies greater than 10 MHz. In this case, use of an external quartz oscillator module or other source of externally generated clocks at the desired frequency, up to the 16 MHz specification, will allow the MCU to function correctly.

CPU does not start up correctly after switching to external crystal

Errata Number: HC12_AR_627

Description

The device can prematurely indicate that the oscillator has stabilized releasing the part from Limp Home clock mode to the oscillator clock mode with an unstable oscillator. This can cause unpredictable behavior of the MCU. This situation can arise with short external power-on reset periods and / or crystal oscillator circuits that exhibit slow startup characteristics. If the PLL is not being used (Vddpll connected to Vss) Limp Home mode is disabled and this issue does not apply.

All customers should review any applications based on the referenced devices. If the crystal clock is stabilized before the external RESET line is released and the customer is not using stop mode (pseudo-stop is not affected) then there is

no problem. If the clock is not stable when external RESET is released then they should contact Motorola for consultation.

Common practice for the start up mode of operation of HC12 microcontrollers is for the external RESET line to be held active until such time as the crystal has stabilized at its operating frequency. On release of the external RESET line and when the WCR (counter register) reaches a count of 4096 cycles, normal operating mode is entered with the CPU clocked from the crystal frequency (see fig. 1).

The HC12 mode of operation known as Limp-Home Mode (LHM) is enabled when the VDDPLL pin is at VDD and is entered if for any reason the external crystal ceases to oscillate. During this mode the CPU will be clocked from the free-running VCO clock of the PLL (at a nominal frequency of 1MHz). If LHM is enabled during the start-up phase (i.e. VDDPLL=5V, NOLHM bit=0) and the external RESET line is not held active until after the crystal frequency is stable then the device starts up in LHM since no crystal oscillations will be detected. This situation can arise with short reset periods and/or crystals that exhibit slow start-up characteristics.

For the first 4096 cycles i.e. during the internal reset period, Limp Home mode will be de-asserted if oscillator activity is detected by the clock monitor circuit - due to the asserted Reset signal there can be no CPU activity during the Reset phase. Following release of the external or internal POR RESET in LHM (which ever is later) the crystal oscillator is sampled by the clock monitor circuit after another 4096 VCO clock cycles and at intervals of 8192 clock cycles thereafter until the crystal is deemed to be operating. If the crystal oscillator is showing activity at the time it is checked then it will be deemed to be good, even though it may not have fully stabilized, and LHM will be de-asserted. This can cause the device to switch from LH mode to normal mode with the CPU clocked from an unstable signal from the crystal oscillator (see fig. 2) resulting in unpredictable function of the CPU.

The COP Reset doesn't exhibit this behavior as, although the same reset sequence is followed, the oscillator isn't stopped.

When exiting Stop mode (DLY=1) a similar 4096 cycle delay is executed and therefore this behavior could also show up at this time. In applications where this is likely to be an issue, using pseudo-stop is recommended as an alternative. Current draw will increase <100 μA at 4MHz in pseudo stop versus stop mode.

Following a loss of external clock in normal operation, Limp Home mode will be entered successfully but if the oscillator is reconnected for some reason a similar situation may arise.

Workaround

The Reset condition can be overcome by allowing the crystal oscillator circuit to stabilize before releasing the external RESET line (see fig. 3). Operation is similar to that shown in fig 2.

To determine if crystal is 'stable' at the release of reset can be difficult and the time can vary some from board to board. If the customer has special high

impedance probes, it is possible to monitor the amplitude of the voltage from XTAL to ground (<2 pF scope probes are recommended). Please note that any loading on the circuit can affect its operation. (Any resistance to ground or Vdd on the EXTAL pin can greatly attenuate the amplifier gain and cause erroneous operation.)

A second way to measure the oscillator startup time is to monitor the XFC pin. This method does not require a high impedance scope probe. The PLL will not lock until the oscillator clock feeding it is present and stable. Remove the external reset circuit and during power up watch the XFC pin. The voltage should start high (Vdd). After the part releases internal reset it will drop to some stable voltage between Vdd and Vss. If external reset (measured independent from this test) is held till this 'stable voltage' time the oscillator will be stable. Please note the filter components mounted on the XFC pin will affect this ramp (for evaluation purposes, alternative components can be selected to provide a fast lock time). More than one board should be measured because of pcb and crystal variability. It is also recommended that the test be run over the operating temperature of the device.

Lastly, an alternative and simpler approach is to just hold reset low for a substantial time (> 100 milliseconds) after Vdd has reached the operating voltage range.

In some applications it may be possible avoid this issue by delaying the connection of Vddpll to Vdd until the device has exited reset. This will sacrifice the limp home mode safety function upon startup, i.e. the part will no longer be able to start without a functioning crystal. A similar technique (disable PLL under software control) can be used to overcome the limitations of Stop mode.

Limp-Home mode start-up issue

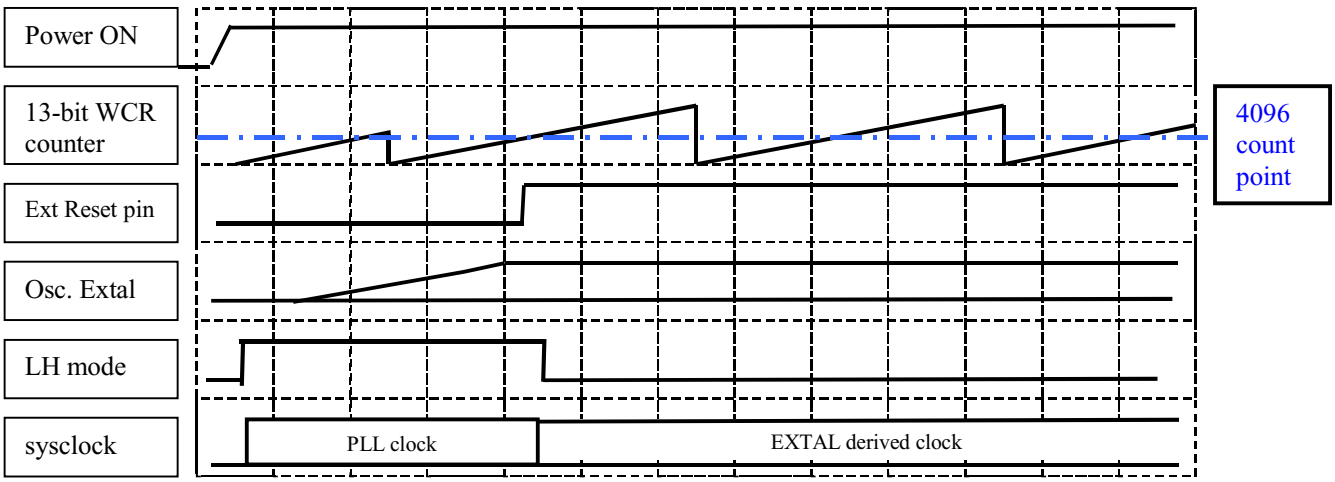


Figure 1. Representation of Normal start-up condition

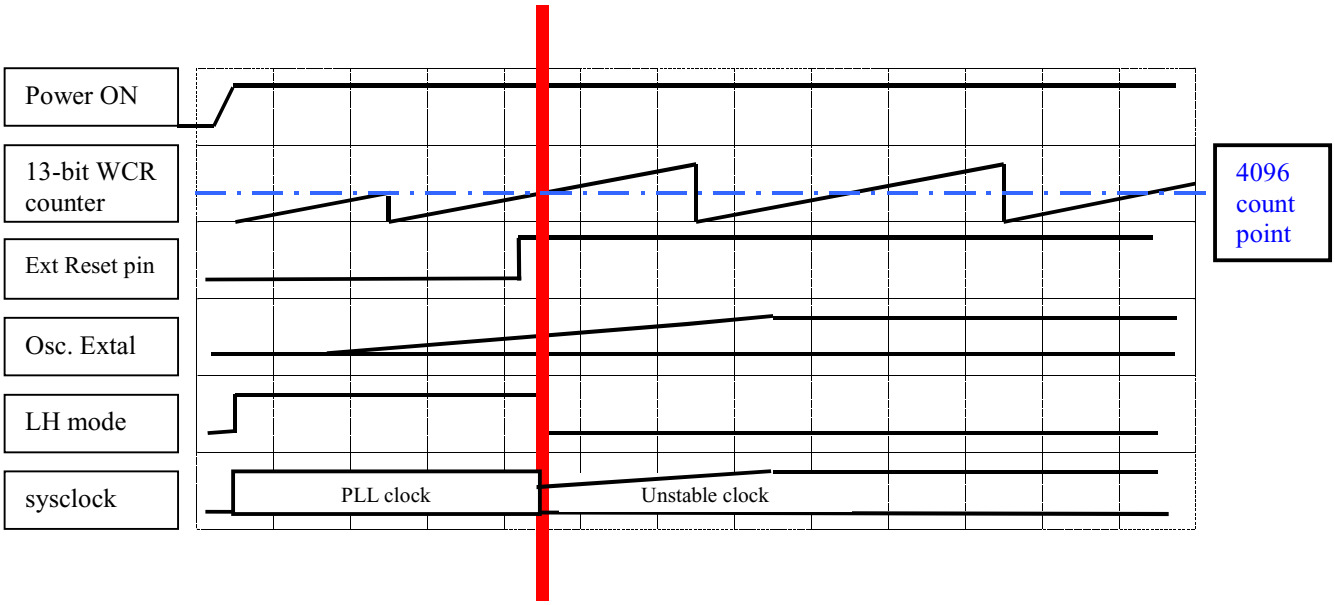


Figure 2. Representation of unreliable start-up mode with slow crystal

Figure 3. Representation of preferred means of overcoming issue with Figure 2

Limp home clock frequency is too high — Errata Number: HC12_AR_636

Description

The typical bus clock frequency in limp home mode is 3 MHz, which is above the maximum specification. The maximum frequency under all operating conditions will be 7.5 MHz until adjustments are made on future versions.

Workaround

Any program code which is frequency dependent during limp home mode should be modified.

XIRQ during last cycle of STOP instruction causes run away — Errata Number: HC12_AR_650

Description

If an XIRQ interrupt occurs during the execution of the STOP instruction with the control bit DLY=0 (located in the INTCR register), the CPU may not run the software code as designed.

Workaround

1. Set the delay control bit DLY=1 so that a delay will be imposed prior to coming out of STOP.
2. If using XIRQ with a stable external clock and DLY=0, contact Motorola Applications Department for a detailed workaround.

Exit from pseudo-stop in limp home — Errata Number: HC12_AR_638

Description

If the MCU is operating with the crystal oscillator driving the bus clock (BCSP=0) as it enters pseudo-stop, the chip goes through limp home mode after it wakes up. This is not necessary since the crystal oscillator will be running and available immediately. The limp home bus frequency will not be as accurate as the crystal-driven frequency.

Workaround

N/A

PA Overflow flag not set when event is concurrent with write of $FFFF Errata Number: HC12_AR_644

Description

When the value $FFFF is written to PACA or PACB and, at the same time, an external clocking pulse is applied to the PAC, the pulse accumulator may overflow from $FFFF to $0000, but the pulse accumulator overflow flag [PAFLG,PBFLG] is not set. Same situation may happen with 8-bit pulse accumulators PAC1 and PAC3.

Workaround

The input capture function for the subject channel be enabled prior to writing a value to the PACA or PACB. Write to the pulse accumulator register. Then do one NOP (to allow the input capture to update the interrupt flag) followed by a read of the input capture interrupt flag to see if it set. If yes, a check must be made for a missing pulse accumulator event. Steps for software workaround to see if event happens while writing to PAC:

1. Enable Input Capture on same pin as the pulse accumulator (and same type of event).
2. Clear the appropriate CxF in the timer interrupt flag register.
3. Read PAC and store as “Old PAC”.
4. Calculate desired PAC value and write it to the PAC.
5. Execute 1 NOP.
6. Read CxF in the timer interrupt flag register.

   If flag is not set, done (no events happened while writing to the PAC).

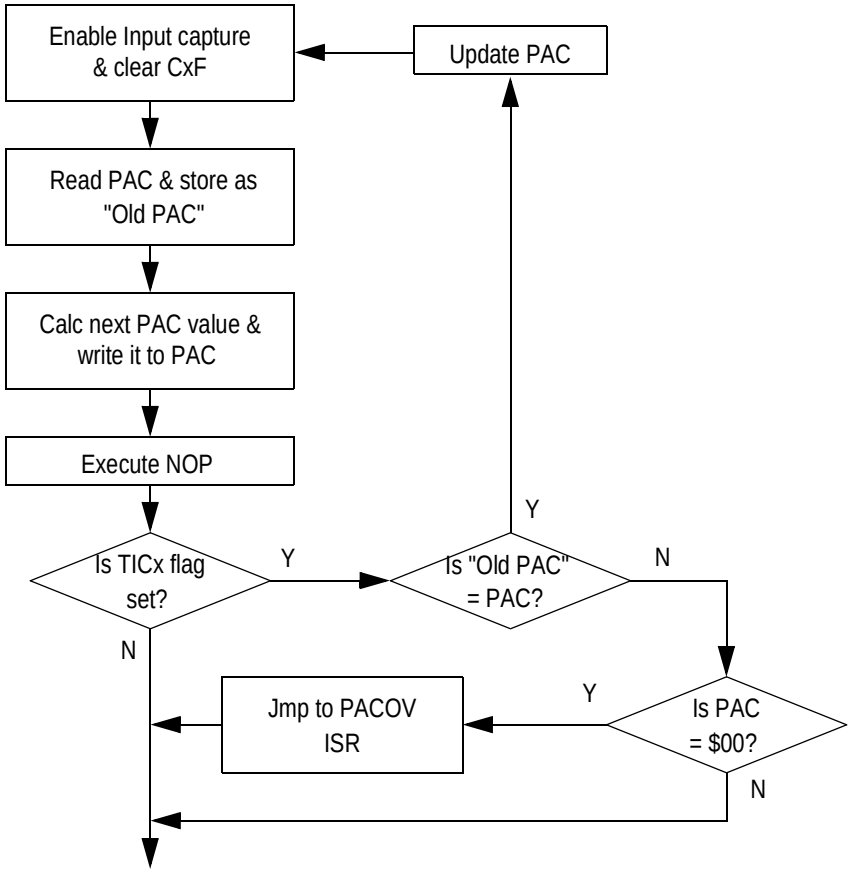
   If flag is set read PAC

   If “Old PAC” = PAC, then update PAC (event happened while writing to PAC and the PAC did not capture it). Note, if the updated PAC value is $00 jump to PACOV ISR.

   If “Old PAC” does not equal PAC, does PAC = $00 ?

   If yes, jump to PACOV ISR.

   If no, done (event happened while writing to the PAC and PAC captured it). Read CxF in the timer interrupt.



SCL divider has an extra clock at 8Mhz bus frequency **Errata Number: HC12_AR_526**

Description At maximum system frequency, the IIC bus rate slows down as much as 5%.

Workaround Communication rate will be adjusted automatically to slower rate.

IIC hold both SCL and SDA lines low when IBB bit is not busy

Errata Number: HC12_AR_573

Description If SCL line is pulled low when generating a start signal the device will lock up.

Workaround After trying to generate a START signal and neither the IBB nor IBAL bits are set after several cycles, the IIC should be disabled and reenabled with the IBEN bit.

Disabling IIC can glitch and corrupt IIC bus

Errata Number: HC12_AR_548

Description If the IIC module is disabled by clearing the IBEN bit in IBCR register, the SDA and SCL lines in the IIC bus will glitch to zero if PORTIB bits 6 and 7 are zero.

Workaround Set PORTIB bits 6 and 7 to one prior to clearing IBEN bit in IBCR register.

MSCAN extended ID rejected if stuff bit between ID16 and ID15

Errata Number: HC12_AR_646

Description For 32-bit and 16-bit identifier acceptance modes, an extended ID CAN frame with a stuff bit between ID16 and ID15 can be erroneously rejected, depending on IDAR0, IDAR1, and IDMR1.

Extended IDs (ID28-ID0) which generate a stuff bit between ID16 and ID15:

IDAR0	IDAR1	IDAR2	IDAR3
********	***1111x	xxxxxxxx	xxxxxxxx

where x = 0 or 1 (don't care)
* = pattern for ID28 to ID18 (see following).

Affected extended IDs (ID28 - ID18) patterns:

a. xxxxxxxxx01 exceptions:00000000001
01111100001
xxxx1000001 except 11111000001

b. xxxxx100000 exception: 01111100000

c. xxxx0111111 exception: 00000111111

d. x0111110000

e. 10000000000

f. 11111111111
g. 10000011111

When an affected ID is received, an incorrect value is compared to the 2nd byte of the filter (IDAR1 and IDAR5, plus IDAR3 and IDAR7 in 16-bit mode). This incorrect value is the shift register contents before ID15 is shifted in (i.e. right shifted by 1).

Workaround

If the problematic IDs cannot be avoided, the workaround is to mask certain bits with IDMR1 (and IDMR5, plus IDMR3 and IDMR7 in 16-bit mode).

Example 1: to receive the message IDs
xxxx xxxx x011 111x xxxx xxxx xxxx xxxx
IDMR1 etc. must be 111x xxx1, i.e. ID20,19,18,15 must be masked.

Example 2: to receive the message IDs
xxxx 0111 1111 111x xxxx xxxx xxxx xxxx
IDMR1 etc. must be 1xxx xxx1, i.e. ID20 and ID15 must be masked.

In general, using IDMR1 etc. 1111 xxx1, i.e. masking ID20,19,18,SRR,15, hides the problem.

HOW TO REACH US:

USA/EUROPE/LOCATIONS NOT LISTED:

Motorola Literature Distribution;
P.O. Box 5405, Denver, Colorado 80217
1-303-675-2140 or 1-800-441-2447

JAPAN:

Motorola Japan Ltd.; SPS, Technical Information Center,
3-20-1, Minami-Azabu Minato-ku, Tokyo 106-8573 Japan
81-3-3440-3569

ASIA/PACIFIC:

Motorola Semiconductors H.K. Ltd.;
Silicon Harbour Centre, 2 Dai King Street,
Tai Po Industrial Estate, Tai Po, N.T., Hong Kong
852-26668334

TECHNICAL INFORMATION CENTER:

1-800-521-6274

HOME PAGE:

http://motorola.com/semiconductors

MOTOROLA

Additional mask set erratas can be found on the World Wide Web at http://motorola.com/semiconductors.