MSE908QB8_2L87X
Rev. 0, 5/2005

Mask Set Errata for Mask 2L87X

Introduction

This mask set errata applies to the mask 2L87X for these products:

- MC68HC908QB8

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 2L87X. All standard devices are marked with a mask set number and a date code.

LIN Break Delimiter Recognition

SE91-LIN_Break

Description

Under specific timing conditions, the ESCI module will fail to properly detect the break delimiter character in a LIN message. When this occurs, the falling edge that marks the beginning of the actual START bit of the synch 0x55 character is not properly seen by the ESCI. Therefore, the next falling edge to appear on the receive pin after that time (beginning of bit 1 of 0x55 character in LIN) is incorrectly interpreted to be the beginning of a START bit of a new byte.

The timing conditions which cause this to occur are when both:

1. The break character is observed at the receive pin to end approximately on a half bit boundary (about 11.5 bits, 12.5 bits, 13.5 bits, etc.) according to the ESCI local receive clock.
2. The break delimiter received appears to be shorter than one bit time, as measured by the ESCI local receive clock.

The length of the break character as observed by the ESCI at the receive pin (See 1. above) is affected by all of the following:

- The length of the actual transmitted character
- The frequency drift in the local slave clock due to temperature and voltage variations
- Ground shift and edge slew effects in the physical layer

These factors can combine to cause a break character that was transmitted by a master to be nominally 13 bits (as measured by the master clock) to be observed by the slave device as ending on a half bit boundary (13.5 bits as measured by the slave clock, for example).

The break delimiter character only needs to be two or more clock cycles shorter than one bit time to cause the break delimiter to not be detected. (See 2. above)

This situation can occur without violating LIN timing specifications or microcontroller oscillator specifications.

Workarounds

Switching the Receiver Off/On — Polling Method of Delimiter Detection

One way to properly sample the BREAK delimiter is to switch off the receiver after the rising edge of the delimiter is detected by the arbiter, wait a clock cycle and switch back on the receiver. This will reset the internal sampling clock and will insure accurate recognition of the delimiter.

To implement this solution, in the break symbol interrupt (error interrupt when BKF is set):

1. Set up the arbiter to measure falling- to rising-edge pulse
2. Poll AFIN until set to indicate edge of break delimiter
   – To minimize this polling time, ensure that the LINR bit is set to move the break detection interrupt to the 11th-bit of the break character.
3. Clear RE to disable receiver, execute a NOP, then set RE to re-enable receiver
   – The purpose of the NOP instruction is to prevent a C compiler optimization from deleting this step.
4. Clear FE interrupt by reading data register then exit ISR.

This solution uses polling of the AFIN bit to detect the rising edge of the delimiter. As this is a polling method, the user must evaluate impact of this method to system timing performance.

Switching the Receiver Off/On — Using a Timer or Keyboard Interrupt Pin for Delimiter Detection

A non-polling solution can also be implemented using an additional input capture or keyboard interrupt input to detect the rising edge at the end of the break symbol.

The input used must be both:

- Physically connected to the ESCI receive pin
- Able to generate an interrupt upon detection of a rising edge on that pin

To implement this solution, in the break symbol interrupt (error interrupt when BKF is set):

1. Set up the appropriate module to detect rising edge on the receive pin and generate an interrupt
2. Clear the FE and BKF flags by reading the data register, and then exit the ISR.

In the rising edge detection interrupt:

1. Clear RE to disable the receiver, execute a NOP, then set RE to re-enable the receiver
   - The purpose of the NOP instruction is to prevent a C compiler optimization from deleting this step.
2. Clear appropriate flags if needed and exit the ISR.

Combination

A combination of both previous methods could also be implemented to reduce CPU polling impact. For instance, a periodic interrupt could execute some code during the polling and then return to polling, or the user could put some application code in the polling loop.

The time between when the rising edge actually appears on the receive pin and when the receiver is switched back on is finite and is governed by the timing of the falling edge of the delimiter. The receiver must be switched back on no later than 9/16th of a bit time before the end of the delimiter. See Figure 1.

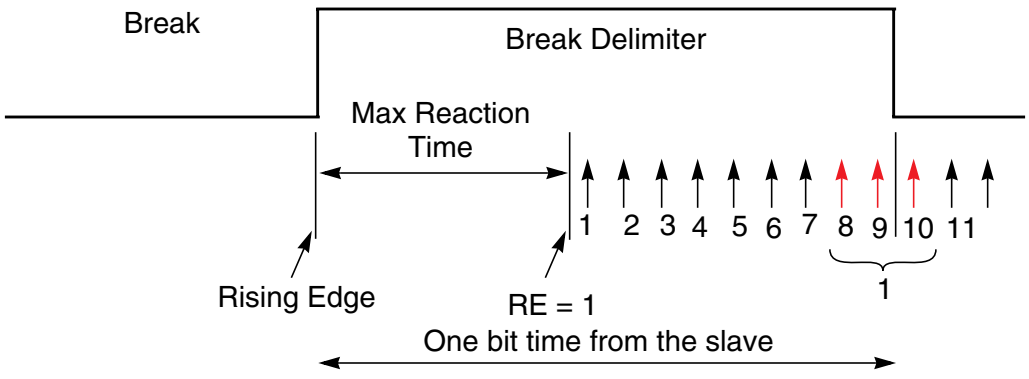

Figure 1. Maximum Time for Other Execution

From Figure 1, we also see that the “Maximum Reaction Time” is 7/16th slave bit-time.

Translating that into an example LIN system yields:

One bit time at 19200 bps is 52.1 µs.

The LIN 1.3 specification indicates the slave can run at 14%and the master at 0.5%of the nominal baudrate. The smallest bit-time the slave might observe is (accounting for accumulated timing errors of a slow slave and fast master):

(100%– 14.5%) x (52.1 µs) = (0.855) x (52.1 s) = 44.5µs

The Maximum Reaction Time for this example system is 7/16 x (44.5 µs) = 19.47 µs.

From the instant the rising edge appears on the receive pin (detected via the arbiter AFIN bit or a timer interrupt), the software has to switch the receiver OFF and back ON within 19.47 µs.

Because ground shift in the physical layer might also affect the timing of the falling edge of the START bit of the 0x55 synch byte (i.e., the end of the delimiter), best design practices suggest that the switching of the receiver should be done as quickly as possible after the rising edge is detected. The maximum reaction time calculated above should be treated as a guideline only.

Modifying the Baudrate to Reduce Occurrence of Missed Break Delimiters

Another workaround which may be used is to temporarily increase the baud rate of the slave device during the reception of the break delimiter. This helps to ensure that the break delimiter appears to the ESCI receiver as greater than one bit time, eliminating timing condition 2 under most operating conditions.

To implement this solution:

1. In the break symbol interrupt (error interrupt when BKF is set), increase the bit rate of the ESCI by a maximum of 9%
2. Remove the 9% speed increase when calculating the baud rate adjustment for that message frame. Only remove the portion of the 9% adjustment that is implemented in the enhanced prescaler (PD+PDFA), as this is the prescaler stage that affects the arbiter measurement of bit time.

The value of 9% allows for differences in clock speeds between master and slave (< 5.5%) and partially compensates for possible effects of a positive ground shift and slew rate effects in the physical layer. These physical layer effects can shorten the length of a recessive pulse (e.g.,the break delimiter) by as much as 20.8% (–10.4 µs at 20 kbps) of its nominal value and remain within the LIN specification. (Refer to the LIN Physical Layer Specification for more information).

This method is not 100% effective in removing timing condition 2, as the LIN specification does allow enough ground or battery shift to reduce the break delimiter by as much as 20.8%. This means that under absolute worst case conditions in a LIN system, this workaround will not guarantee that a delimiter can be seen.

Ground shift effects cannot be fully compensated for as you would need to increase baud rate by as much as 27%. However, 9% is the maximum safe value to increase baud rate, as it fully compensates for the 5.5% maximum clock difference between master and slave and partially compensates for smaller ground

shift effects. If these factors combine, an effective difference of 14.5% in baud rate may be seen with this workaround.

In a LIN system, since the 0x55 character is known to be the next character to be received, the typical guideline of 4.5% difference between two UART devices can be stretched to a little more than 15%. This is because the greatest distance between two falling edges in a 0x55 character is only two bit-times and the UART will resynch on those edges and guarantee reception of the character if both devices are within around 15% of each other's speeds.