IIC APPLICATION NOTES

DATE. MAR.05.2003

RELATED DEVICES

S3C44A0A, S3C44B0X, S3C2400X, S3C2410X

PROBLEM & WORKAROUND

1. SDA Data Hold Time Problem

Problem: The tSDAH(SDA data hold time) is min. 0ns by the specification. But, the real tSDAH is 3-clock duration based on MCLK or PCLK. This tSDAH may be insufficient for some IIC slave device. A few slave devices may not receive the valid address sometimes due to the lack of SDA hold time and will not acknowledge in spite of a valid addressing. If the SDA data hold time is insufficient, the error occurs very rarely and at random time.

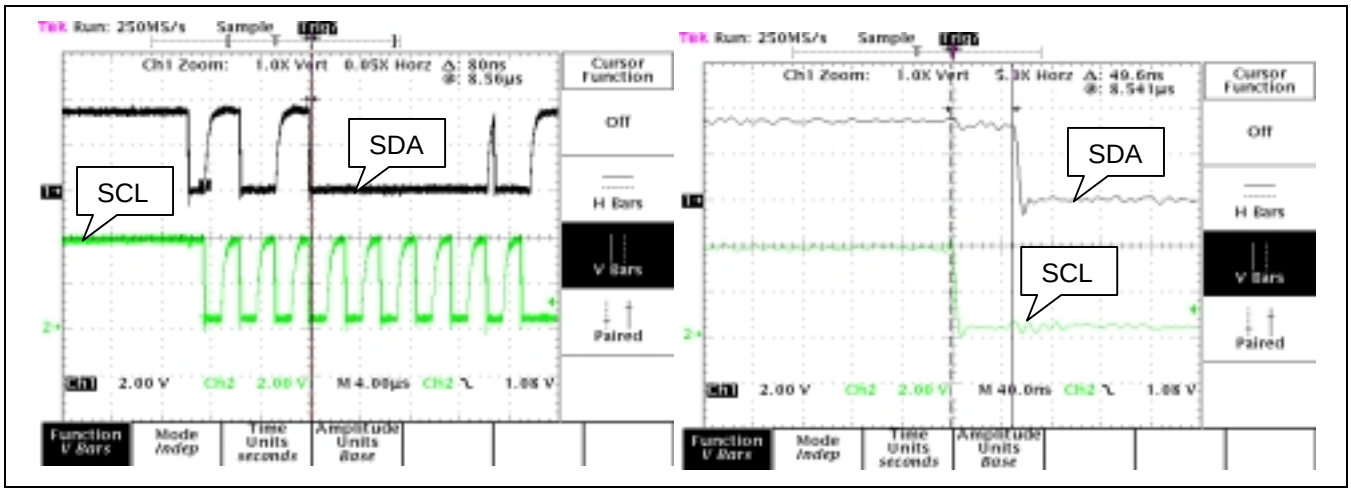
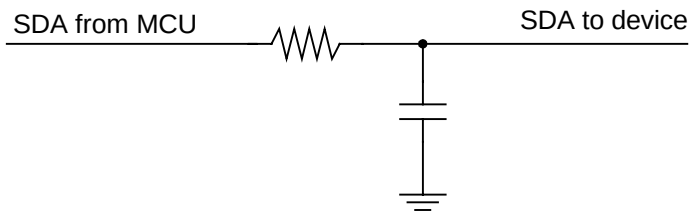


Figure 1. S3C44B0X IIC SDA/SCL Waveform

Workaround: If some device needs more SDA data hold time than 3 MCLK (or PCLK), RC delay circuit is needed on the SDA line as follows;



For example, R=100ohm and C=200pF can be one of available values. We recommend implementing the RC delay circuit on SDA line for case of need. Mostly it may not be needed.

2. SCL Glitch Problem When Using IIC Hub and SCL Clock Stretching.

Problem: If an IIC hub is used and a slave device are using the clock stretching, there may be a glitch on the SCL line of MCU.

If MCU drives the SCL line as 'L' during the clock stretching of a slave device, the MCU SCL line will be held as 'L' by MCU itself. Even though the MCU releases the SCL line during the clock stretching, the MCU SCL line should be also held as 'L' by an IIC hub because of clock stretch duration. Unfortunately, there may be some delay for IIC hub to drive MCU SCL line 'L' again. So, there may be short glitch on MCU SCL line.

MCU may recognize this glitch as a normal clock and start to operate in spite of the clock stretch duration.

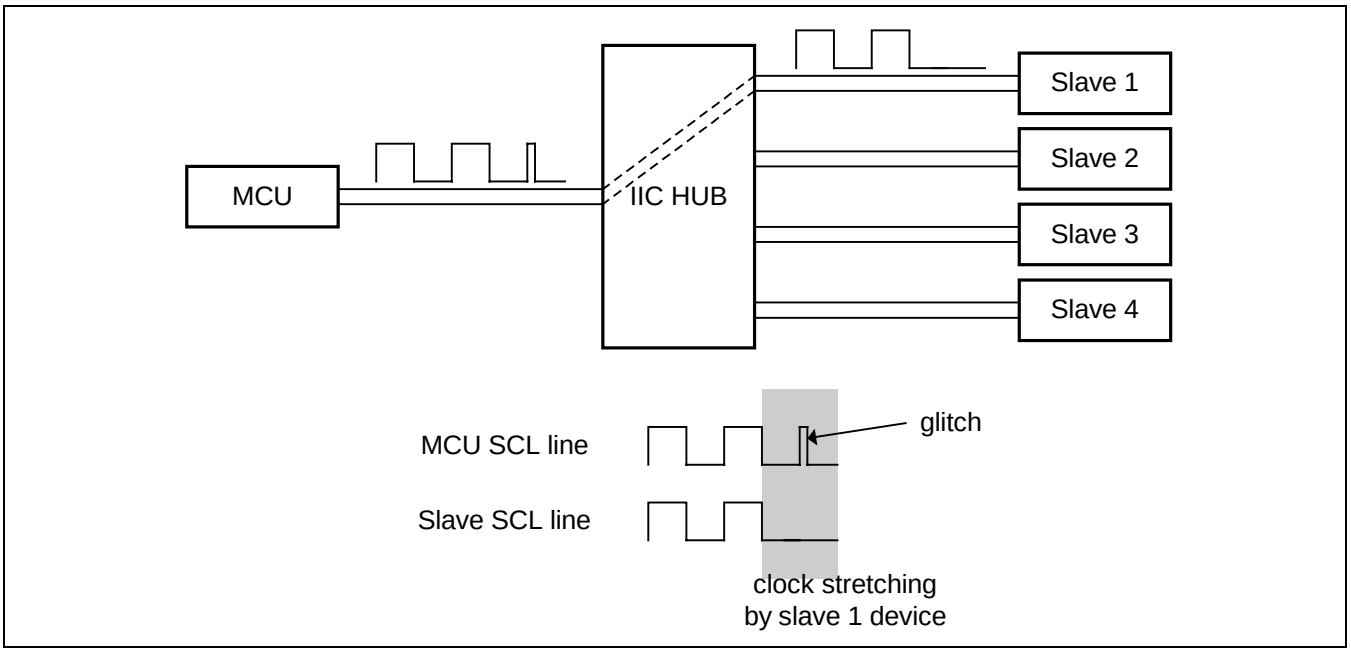
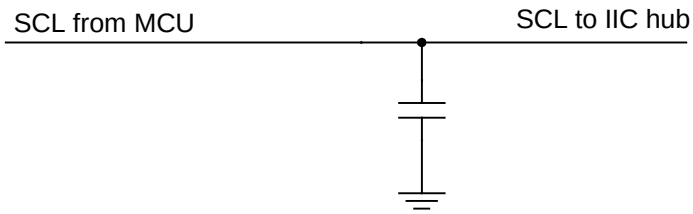


Figure 2. Glitch On MCU SCL Line Using IIC Hub and Clock Stretching

Workaround:

1) If the slave device using clock stretch is interfaced directly without IIC HUB, there is no problem
2) If you have to use an IIC hub and the clock stretching slave device, you may have to suppress the glitch by a capacitor on MCU SCL line as follows. The capacitor value will be changed by the pull-up resistor value and glitch pulse width.



3. SDA Setup Time

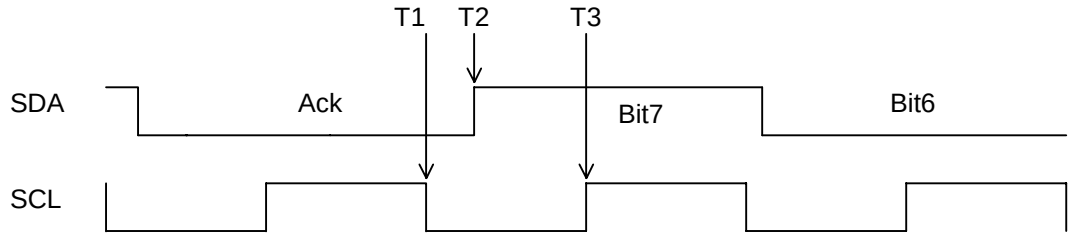
Problem: The SDA data setup time (tSDAS) may be insufficient. Most users may not encounter this problem if the IIC interrupt handler is executed fast enough. But, This problem will be encountered if the IIC interrupt handler is not executed immediately in multitasking environment or by other interrupt service routine.

Most interrupt handler S/W will write a new data and release the IIC pending condition as following codes;

```
rIICDS=_iicData[_iicPt++];
rIICCON=0xaf;        //resumes IIC operation.
```

If the above code is used for writing a new IIC data, there are two cases for IIC SDA setup time as follows;

Case 1. If the IIC interrupt handler is executed immediately, the SDA set-up time will be (T3 - T2).

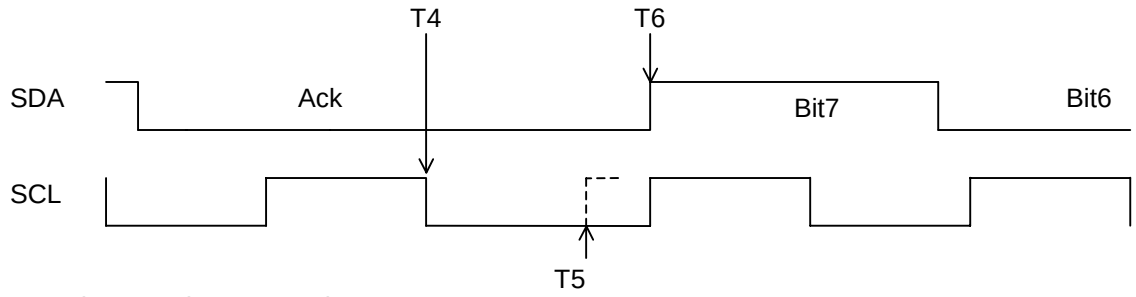


@T1. IIC interrupt is requested

@T2. rIICDS and rIICCON is written. But, SCL will not toggle at T2.

@T3. Just after 1/2-bit time is passed, the SCL line toggles.

Case 2. If the IIC interrupt handler is not executed immediately, the SDA set-up time will be insufficient.



@T4. IIC interrupt is requested.

@T5. The IIC 1/2-bit time is passed. So, after this time, the SCL line will toggle just after the IIC pending condition is released.

@T6. The IIC interrupt handler is executed lately by the excessive interrupt latency. rIICDS and rIICCON is written. In this case, SCL will toggle immediately just after rIICCON is written. So, the SDA setup time will be the interval the two instructions writing rIICDS and rIICCON.

Workaround: The SDA data setup time (tSDAS) should be guaranteed by S/W as follows. tSDAS will be over 250ns by the 10-time for-loop-statement.

```
rIICDS=_iicData[_iicPt++];
for(i=0;i<10;i++);  //for setup time until rising edge of IICSCL
rIICCON=0xaf;       //resumes IIC operation.
```