I^2C communication errors on a 17-slot ELMA crate

January 7, 2013

Theodor-Adrian Stana (CERN/BE-CO-HT)

Contents

List of Figures

List of Tables

1 Introduction

This short test report outlines issues we have had with the I2C bus on the SCL and SDA VME lines and outlines suggestions for improvements to be made on the SysMon side of the bus.

Working with a 17-slot VME crate from ELMA (type 041-452, firmware v2.27), we started noticing some issues in the communication. These issues came in the form of incorrect data received by the SysMon, "I2C timer expired" and NACK errors.

After re-checking our side of the communication, and increasing the size of the glitch filter on the SCL and SDA lines to be immune to glitches up to 400 ns wide, the errors kept appearing when loading the bus with more than three cards.

2 Test setup

The test setup contains the following:

1. 17-slot ELMA crate
2. CONV-TTL-BLO with register test firmware in VME slots

When scope measurements are made, the following items were added to the list above:

3. 12 pF 10:1 scope probes connected to SCL and SDA lines
4. Small I2C adapter RTM board

The I2C adapter RTM contains some test pins to connect the oscilloscope probes and some pads to add the resistors mentioned later in this report.

The CONV-TTL-BLO [1] is a VME form factor card used for pulse conversion and repetition between two amplitude standards local at CERN. On the communication side, it contains only the I2C lines connected to the P1 connector through a SN74VME522501 buffer from TI [2]. The VME lines are not connected to the board.

The communication tests are performed using Python scripts that send *readreg* and *writereg* commands to a RAM embedded within the FPGA we have on the CONV-TTL-BLO boards. The value **0x55443322** is written to the first location of the RAM and then the location is read back for checking.

3 Communication issues

We had noticed I2C timer expired and NACK errors appearing with even one slot plugged into the 17-slot ELMA crate. We increased the width of the glitch filter on the SCL and SDA lines to up to 400 ns to make sure no communication errors appear due to this. After this, we started testing the communication again. We gradually introduced one card at a time and ran the test using our Python script.

Communication errors on the bus only start appearing after plugging in the third CONV-TTL-BLO card. When the third card is plugged in, NACKs start being received by the SysMon. With the fourth card plugged in, I2C timer errors appear. These errors become more and more frequent as more cards are plugged in. When the seventh card is plugged in, data reception errors start appearing.

An example of such errors is shown below:

Table 1: Data reception errors

Sent	**Received**
0x55443322	0xffff3322
0x55443322	0x56443322
0x55443322	0xffff3322
0x55443322	0xffffff23
0x55443322	0xffff3922

This was taken out of a test run with seven loads (seven CONV-TTL-BLO cards plugged into the crate). The log of this test can be found in the appendix.

These errors appear at random times and, as can be seen in the test log in the appendix, they are mixed with I2C timeout and NACK errors received from the SysMon.

The fact that errors still appeared led us to some more debugging of our FPGA, whereby we checked to see if the data errors were coming from incorrectly stored values in our RAM due to incorrect reading. This debugging session confirmed that the data stored to the RAM was the correct value of **0x55443322**.

This, together with the fact that the communication works correctly until the third slot plugged in, led us to believe the microcontroller on the SysMon is not sampling the SDA line correctly due to too slow rising edge on the SCL line.

To check whether this was the case, we started by measuring the rise times on the I2C lines.

4 SCL line rise-time and bus capacitance

Using the setup outlined above with only the adapter on the backplane plugged in, the oscilloscope capture of the two lines looks as below:

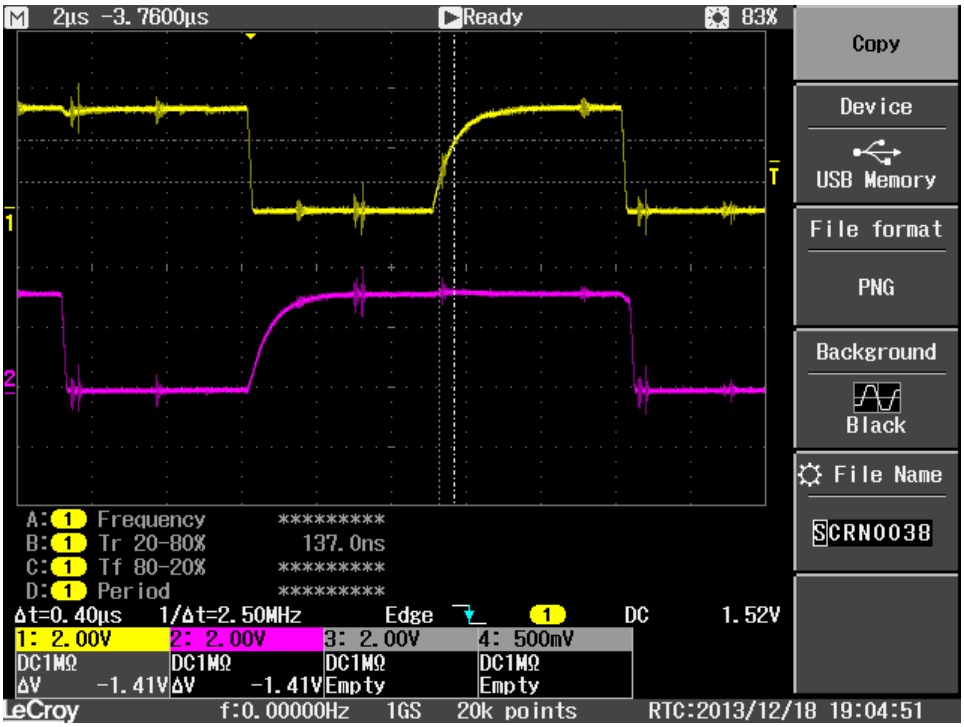

Figure 1: Scope capture with an empty bus

We notice a 30%-70% rise time of 400 ns. This yields (according to equation (1) in the I2C Specification [3] and considering the 2.2 kΩ pull-up resistors) a bus capacitance of 214.58 pF/in:

$$C_{bus} = \frac{400ns}{0.8473*2.2\ k\Omega} = 214.58pF$$

Some ripples can also be noticed in the SCL and SDA signals. These will be addressed in a later section.

Each time one CONV-TTL-BLO is added to a VME slot, the rise time increases by 40 ns. This yields a capacitance of **21.46 pF** for a CONV-TTL-BLO.

Finally, with all seventeen slots fully loaded, we have a 30%-70% rise time of **1160 ns**, which is above the I2C specification for maximum rise time of 1000 ns.

Considering that we cannot change the capacitance on the bus, we tried to decrease the pull-up resistance value to decrease the rise time, thereby potentially avoiding the communication errors.

5 Attempts to improve communication

In order to change the pull-up resistors on the I2C bus, we designed a simple RTM board to plug into the back of the ELMA crate, on the P1 connector of the first slot. This board contains some test pins and some pads to place resistors between the SCL and SDA lines, and VCC. These resistors in parallel to the already-present 2.2kΩ resistors on the SysMon, would decrease the pull-up resistance on the bus, thereby increasing the rise times.

According to the I2C Specification [3], the minimum I_{OL} allowable on the bus is 3 mA, which leads (using the V_{OL} specification from the H8S2462 SysMon microcontroller) to a minimum pull-up resistance of:

$$R_p = \frac{V_{DD}-V_{OL}}{I_{OL}} = \frac{3.3-0.33}{3} = 0.99k\Omega$$

We therefore placed a 1.8 kΩ resistor on the I2C adapter to bring down the pull-up resistance to 0.99 kΩ.

The oscilloscope capture with the bus empty, after changing the pull-up resistance is shown in Figure 2.

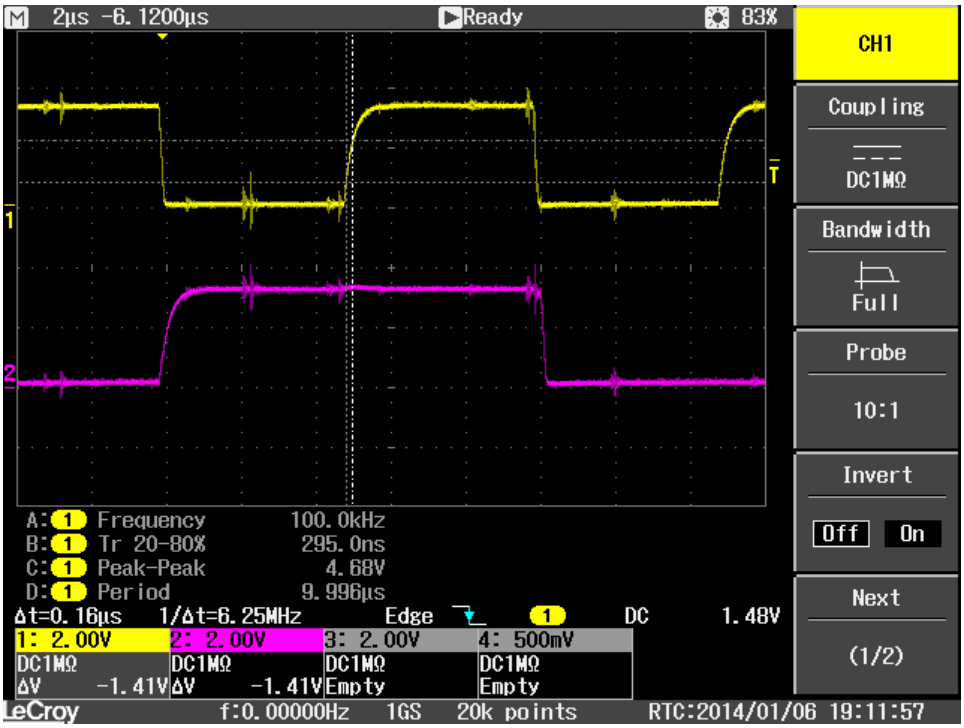

Figure 2: Scope capture with an empty bus and 1.1 kΩ pull-up resistor

As expected, the rise time is decreased, due to decreasing the pull-up resistance. With all seventeen cards plugged in, the rise time on the SCL line is **440 ns**, well within specification (1000 ns).

Also with all seventeen cards plugged in, on this run of the test, the communication presents no more failures, therefore decreasing the pull-up resistance helps with the communication.

6 Ripples on the I2C lines

As seen in both Figure 1 and Figure 2, some ripples appear on the I2C lines. These ripples were measured to check if they might not interfere with the communication, adding unwanted pulses on the SCL line, which may well lead to communication errors such as those seen in Table 1.

Figure 3 shows the oscilloscope capture on the SCL line with the oscilloscope channel set in AC mode and no communication on the line. As can be seen from the figure, these ripples can easily hit 1.5 Vpp, which means they might cause the SCL line on the bus to sample an incorrect SDA if a ripple occurs on a falling edge of SCL. It is unclear whether these ripples are due to the power supply or due to crosstalk.

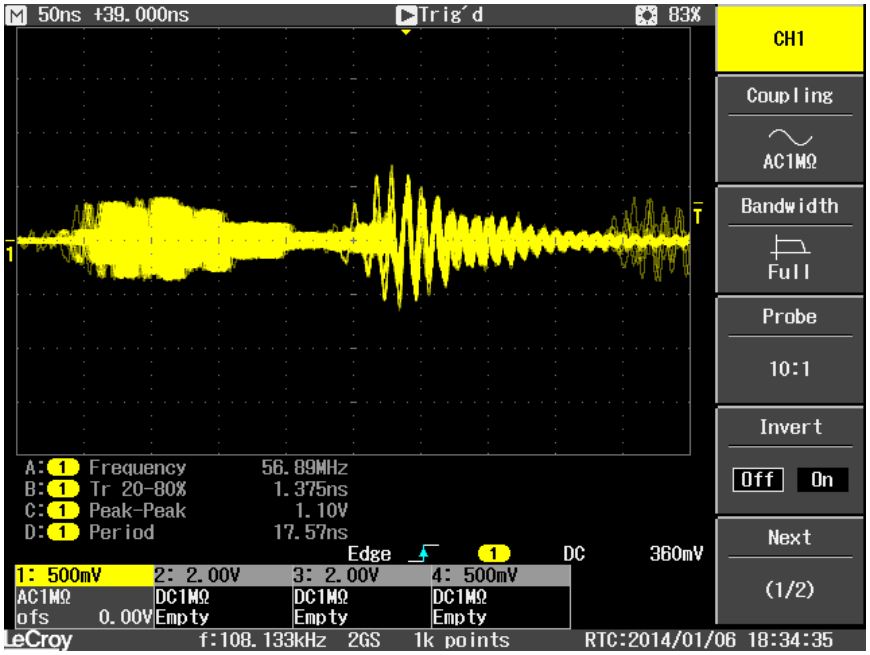

Figure 3: Ripples on SCL line

In previous experiments, we have noticed before errors occurring on the bus even with a lower pull-up resistance, so it is apparent these considerable ripples indeed affect the communication.

Such errors also appeared on one of our two-slot crates where the I2C communication had been working properly. We then noticed the power supply was heating up. This led us to try to place **22 pF** capacitors on

the SysMon, on the pins intended for the C119 and C122 capacitors in the SysMon schematics. Placing these capacitors removed the errors in communication. Unfortunately, the capacitors were removed from the SysMon to allow for testing the I2C pull-up adapter solution, and traces of this test run do not exist, therefore they cannot be included in this report.

7 Conclusions and suggestions for improvement

Errors in communication appear due to incorrect sampling of the SDA line due to:

- the high rise time on the SCL line of 1160 ns, or
- ripples introduced on the I2C lines by either the power supply, or crosstalk on the I2C lines

Some immediate improvements we recommend are:

- replace the pull-up resistors on the SCL and SDA lines with 1 kΩ resistors
- decouple the SCL and SDA lines by placing 22 pF capacitors to ground

In the meantime, since we have many of these crates already deployed, we will place adapter RTM cards such as the one described above in all crates where the I2C communication will be used. The pull-up adapter RTM will contain 1.8 kΩ resistors to V_{cc} and 22 pF capacitors to ground from both SCL and SDA lines.

The following should further be studied:

- where do the ripples on the SCL and SDA lines come from, are they due to the power supply, or due to crosstalk?
- how can the potential noise affecting the SCL and SDA lines be better filtered?
- do more errors occur on a fully-loaded 21-slot backplane?
- will VME accesses going on at the same time as I2C accesses generate crosstalk on the I2C bus?
- what happens when cards with higher capacitance on the I2C bus are plugged into the crate?

Appendices

A Test log

The test log below corresponds to a test run for one minute, with eight CONV-TTL-BLO cards plugged into an ELMA 041-452, firmware version 2.27.

The error outputs are split into columns as follows: first, the type of error that occured is listed together with the slot of the card which was written when the communication error occured. Possible errors can be:

- *wex* – write exception (error response from ELMA crate)
- *rex* – read exception (error response from ELMA crate)
- *mis* – mismatch in read value versus written value

After the type of error, the time at which the error occured is listed, followed by the description of the error. In case of exceptions, the message received from the crate is printed. In case of mismatch errors, first the slot is printed, followed by the value written versus the value read.

```
------------------------------------------------------------
2013-12-16-17h31m09s: starting test
------------------------------------------------------------
wex 1 / 2013-12-16-17h31m15s / I2C Timer Expired exception.
wex 2 / 2013-12-16-17h31m16s / I2C Timer Expired exception.
wex 5 / 2013-12-16-17h31m17s / I2C Timer Expired exception.
mis 1 / 2013-12-16-17h31m18s / 000 : 55443322 != ffff3322
wex 3 / 2013-12-16-17h31m19s / I2C Timer Expired exception.
mis 3 / 2013-12-16-17h31m19s / 000 : 55443322 != 56443322
rex 8 / 2013-12-16-17h31m20s / I2C Timer Expired exception.
mis 3 / 2013-12-16-17h31m21s / 000 : 55443322 != ffff3322
rex 2 / 2013-12-16-17h31m21s / I2C Timer Expired exception.
wex 3 / 2013-12-16-17h31m25s / I2C Timer Expired exception.
wex 4 / 2013-12-16-17h31m26s / I2C Timer Expired exception.
rex 5 / 2013-12-16-17h31m28s / NACK received.
rex 8 / 2013-12-16-17h31m28s / I2C Timer Expired exception.
wex 1 / 2013-12-16-17h31m28s / I2C Timer Expired exception.
rex 6 / 2013-12-16-17h31m29s / I2C Timer Expired exception.
mis 6 / 2013-12-16-17h31m29s / 000 : 55443322 != ffff3322
wex 1 / 2013-12-16-17h31m35s / I2C Timer Expired exception.
rex 1 / 2013-12-16-17h31m36s / I2C Timer Expired exception.
wex 7 / 2013-12-16-17h31m37s / NACK received.
rex 5 / 2013-12-16-17h31m37s / NACK received.
```

```
wex 6 / 2013-12-16-17h31m39s / I2C Timer Expired exception.
mis 5 / 2013-12-16-17h31m44s / 000 : 55443322 != ffff3322
rex 2 / 2013-12-16-17h31m46s / I2C Timer Expired exception.
wex 6 / 2013-12-16-17h31m48s / I2C Timer Expired exception.
wex 4 / 2013-12-16-17h31m50s / I2C Timer Expired exception.
wex 5 / 2013-12-16-17h31m50s / NACK received.
mis 5 / 2013-12-16-17h31m51s / 000 : 55443322 != ffffff23
mis 8 / 2013-12-16-17h31m52s / 000 : 55443322 != ffffff23
rex 1 / 2013-12-16-17h31m52s / NACK received.
rex 2 / 2013-12-16-17h31m52s / NACK received.
wex 1 / 2013-12-16-17h31m53s / I2C Timer Expired exception.
wex 7 / 2013-12-16-17h31m56s / I2C Timer Expired exception.
wex 4 / 2013-12-16-17h31m57s / I2C Timer Expired exception.
rex 2 / 2013-12-16-17h32m00s / I2C Timer Expired exception.
rex 4 / 2013-12-16-17h32m02s / I2C Timer Expired exception.
mis 8 / 2013-12-16-17h32m04s / 000 : 55443322 != ffff3922
wex 4 / 2013-12-16-17h32m05s / NACK received.
wex 4 / 2013-12-16-17h32m06s / I2C Timer Expired exception.
rex 7 / 2013-12-16-17h32m07s / I2C Timer Expired exception.
------------------------------------------------------------
2013-12-16-17h32m09s: ending test
Ran for 60 seconds
------------------------------------------------------------
Write transactions: 13280
Write exceptions: 18

Read transactions: 13280
Read exceptions:  13

R/W mismatches:   8
------------------------------------------------------------
```

References

[1] "CONV-TTL-BLO on Open Hardware Repository." http://www.ohwr.org/projects/conv-ttl-blo/wiki.

[2] T. Instruments, "SN74VMEH22501." http://www.ti.com/product/sn74vmeh22501.

[3] NXP, "UM10204 – I^2C-bus specification and user manual," 10 2012. http://www.nxp.com/documents/user_manual/UM10204.pdf.