I2C ZYNQ - SLOW CONTROL PROBLEM

Ricardo Sato

Pierre Auger Observatory - Malargue

Dec/2018

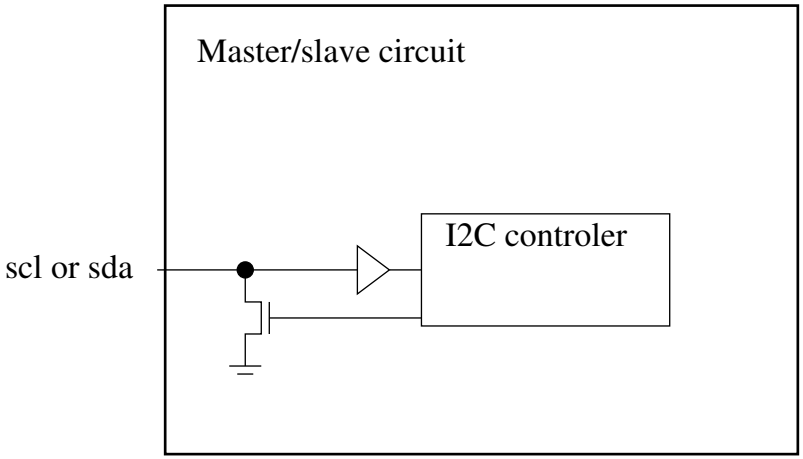
Overview

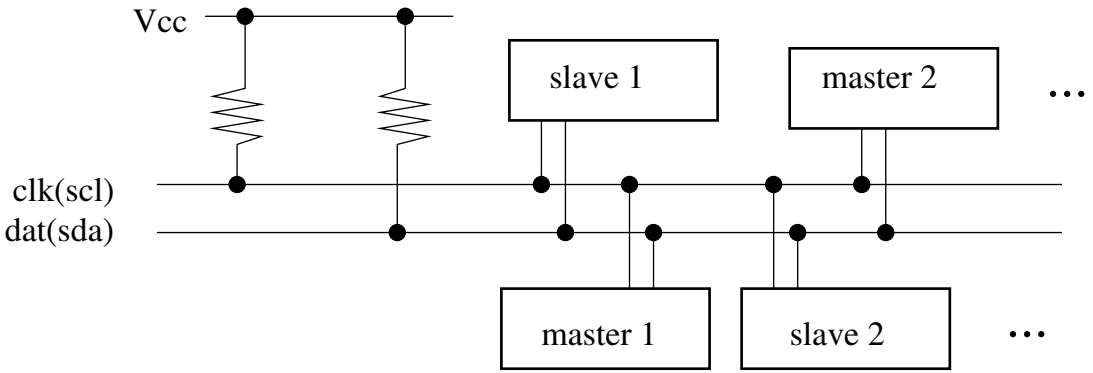
- I2C introduction.
- I2C signals when FPGA/CPU-Slow control stops.
- One trick which “avoid” the problem
- Summary - I2C communication problem.
- I2C electric issues.

I2C introduction

- use two lines: Clock (SCL) and Data(SDA)
- I2C may connect many devices (slaves and masters) in the same line.
- It is controled by one or more master (arbitration). We are using a connection with a single master.
- Each slave is identified by its address.
- All devices in the line would only Pull to the ground the signal lines (clock and/or data).
  - It avoid possible damage in the devices, since it would never generate short circuits.
  - It need a external Pull up resistor or something similar.
  - However, some devices use a pull up resistor to try to simplify the circuit (normally possible to configure).

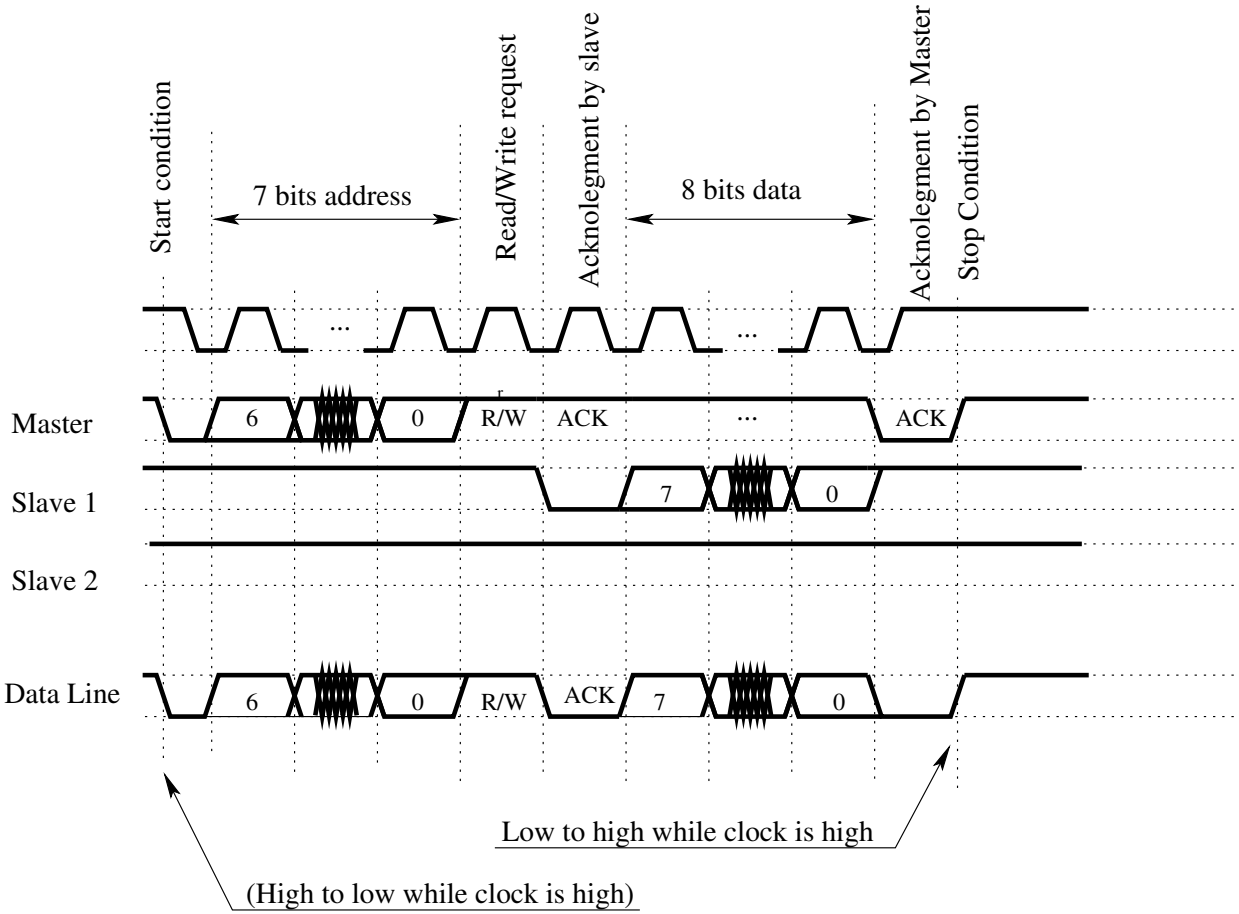
I2C introduction - general I2c Circuit





I2C introduction - read sequence

Only masters can start communication.

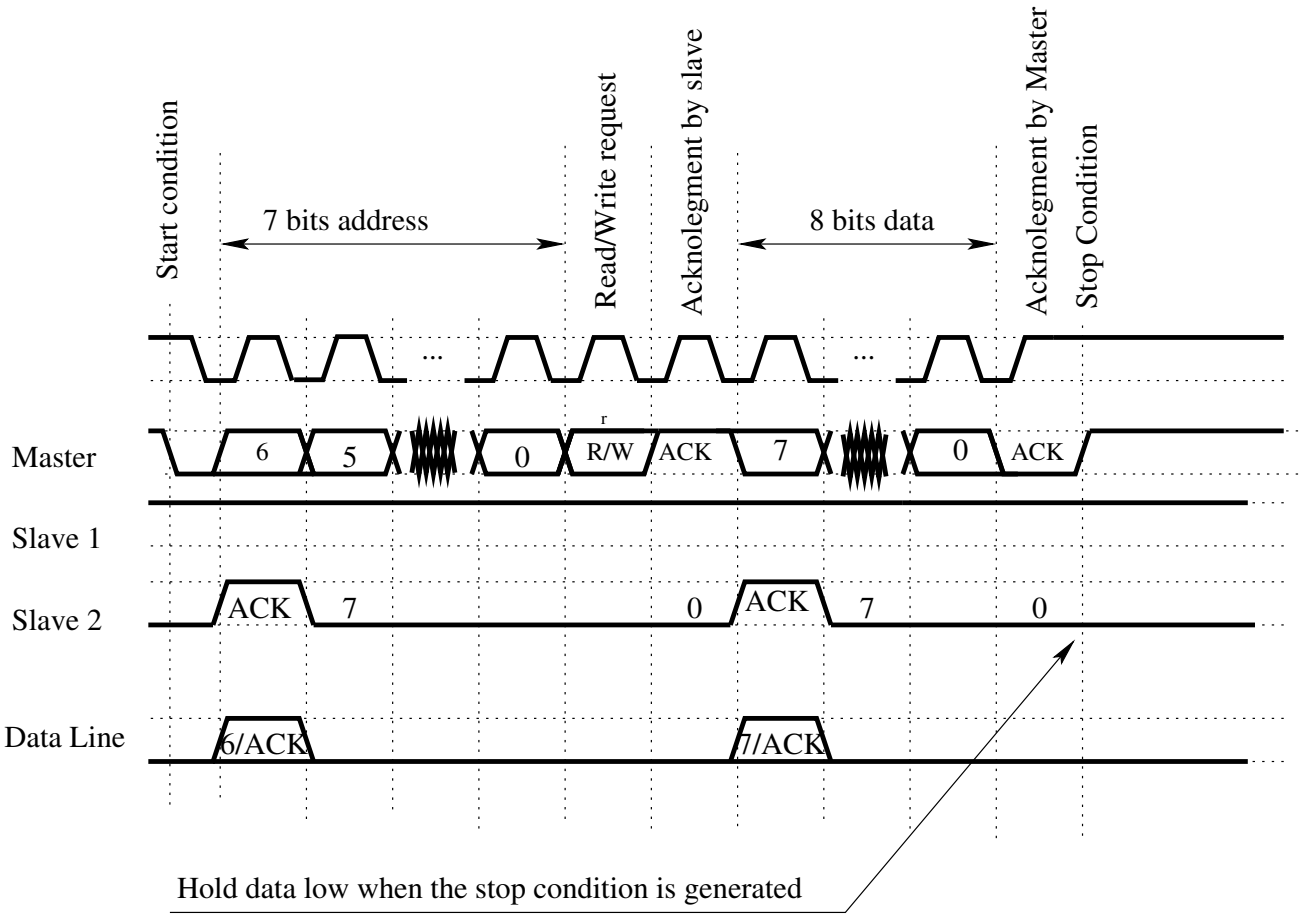


FPGA/CPU-Slow control stops

- It is still not clear why this problem appear.
- LAB tests fail to reproduce the problem.
- Data line is most of time in low (0) state
- Clock line is looks working fine.
- The problem looks to be due to the DAC of LED Flasher (second slave connected in the same line).
- The “start” and “stop” condition can never been visible.
- In One UUB version 2, the DAC of LED Flasher has been removed (the problem did not appear).
- We could figure a way to recover the communication through LINUX (David sugestion).

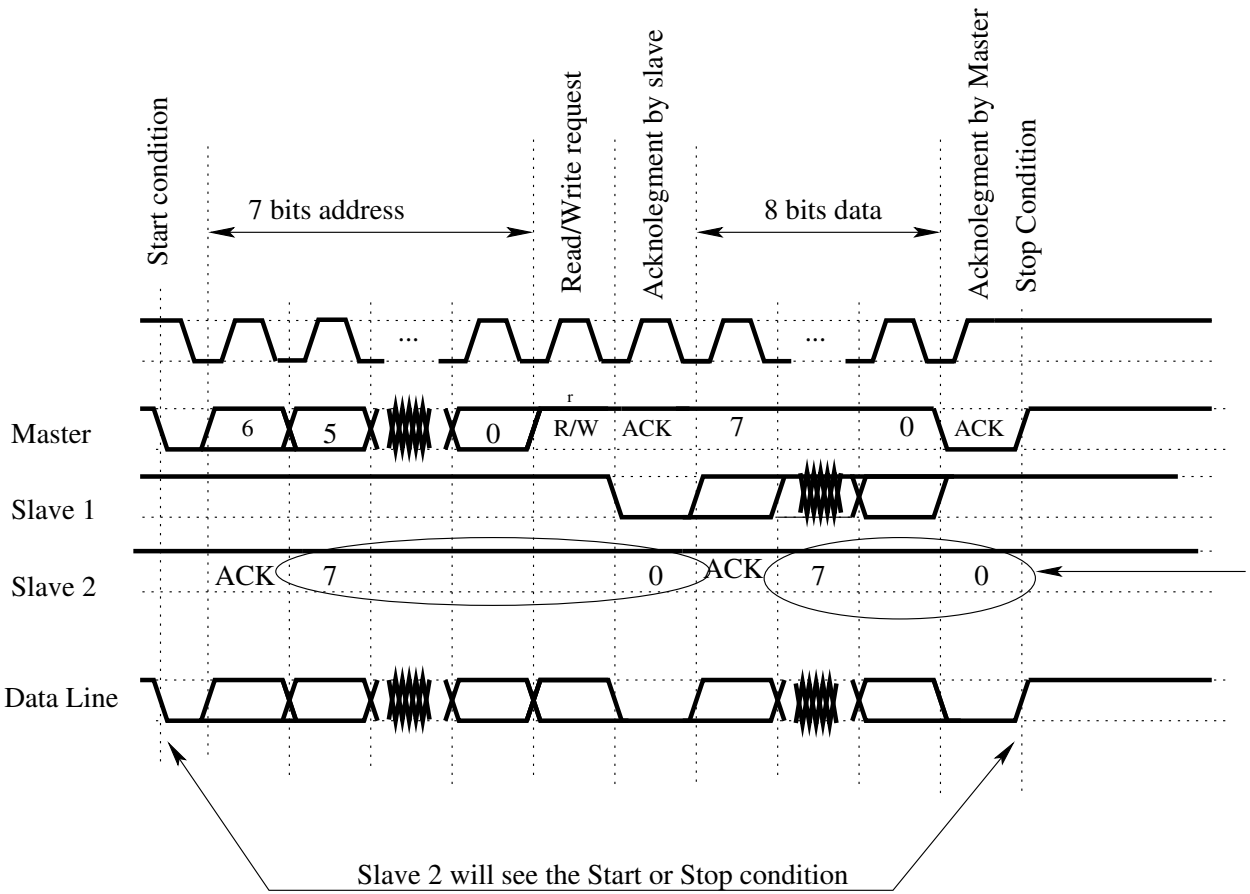
FPGA/CPU-Slow control stops - signal

What Looks to be happening: DAC of LED Flasher looks to enter in data transmission mode (reports the DAC values, which is normally "0").



One Trick which "avoid" the problem.

Set DAC values in the way all the digits is "1"



One Trick which "avoid" the problem.

- The DAC will "see" the "start" and/or "stop" condition.
- since setting DAC values to station Clais Jr(22), since Dec 6th, the I2C communication did not stop.
- The problem is probably appearing, but it looks to recover by itself.
- implemented also in Trak Jr(20) and Peteroa Jr(41) since Dec 7th.
- It does not means the problem is generated by DAC, but the DAC behavior makes the problem visible.

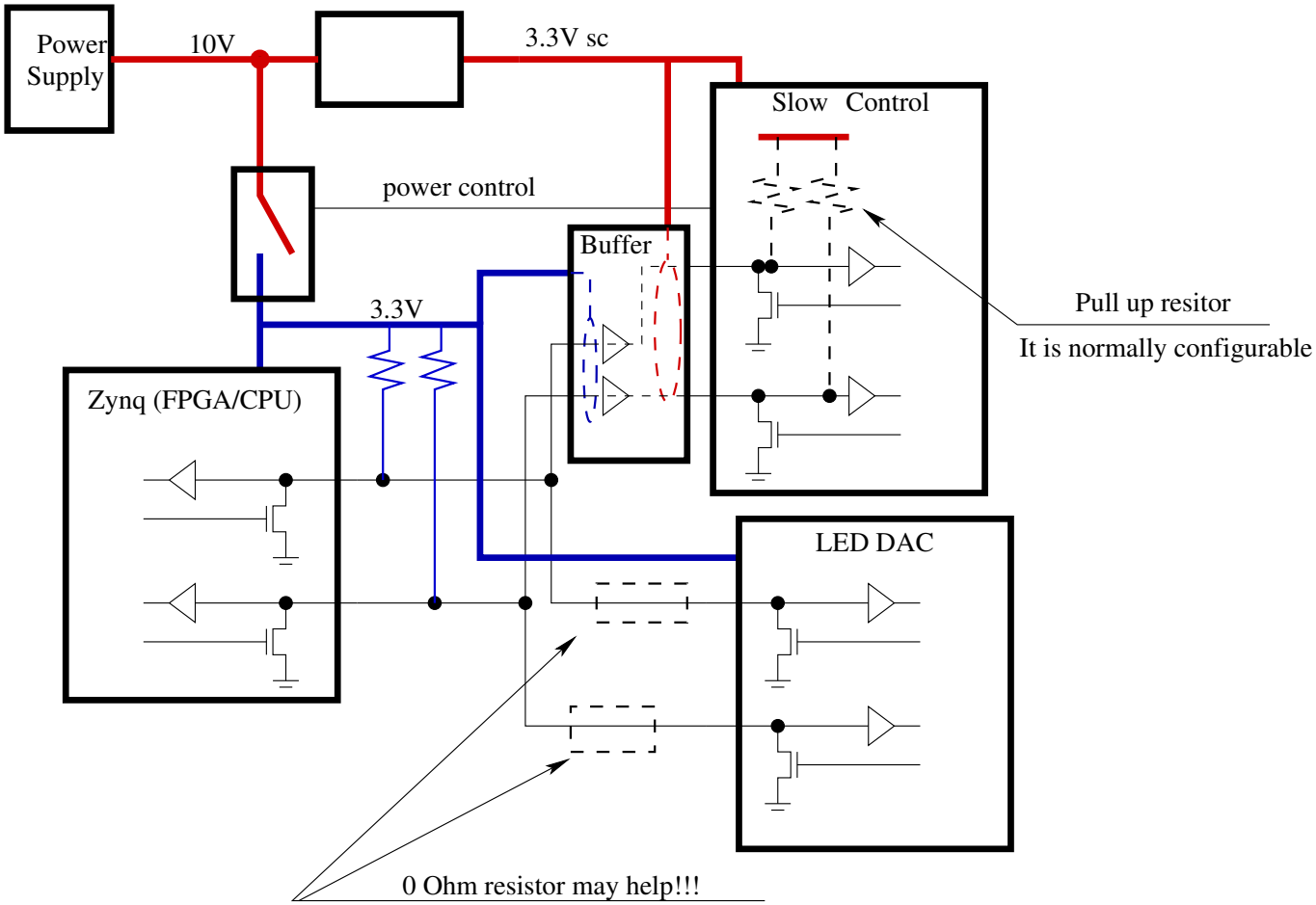
Summary- I2C stop communication issue

- It is related with LED DAC behavior.
  - However the real origin of the problem has not been identified.
  - There are some comments in the web which looks to have similar problem for zynq 7000 family.
- It can be recovered by the LINUX. A program need to be implemented.
- It does not look to be related with DAQ process and watchdog which access the I2C line simultaneously (Matthias consideration - AERA)
- May have some tricks which make the system self recoverable.
- Probably a different DAC (without read feature) could be better (HV PMT configuration is one which has only posibility to write).

Summary- I2C stop communication issue - Cont

- May be interesting to implement some simple error detection algorithms between FPGA/CPU - slow control.
- Others?

I2C electric issues - circuit.



I2C electric issues.

- According Eric, the I2C pins may have some voltage while FPGA/CPU may still be off.
  - Introduce a circuit to avoid such problem.
- From Roberto and my (Ricardo) point of view, this circuit is not needed, if the slow control have proper configuration of to use the I2C pins.