

HW#5-I2C Slave Controller

Hua Xu	hx2136@columbia.edu
Teng Yang	ty2227@columbia.edu

Attachment:

- 1) A diagram of time sequence table for both Slave act as sender and slave receiver.
- 2) Detailed simulation diagrams.
- 3) VHDL code.
- 4) Moore FSM states diagram.

CSEE W4823x

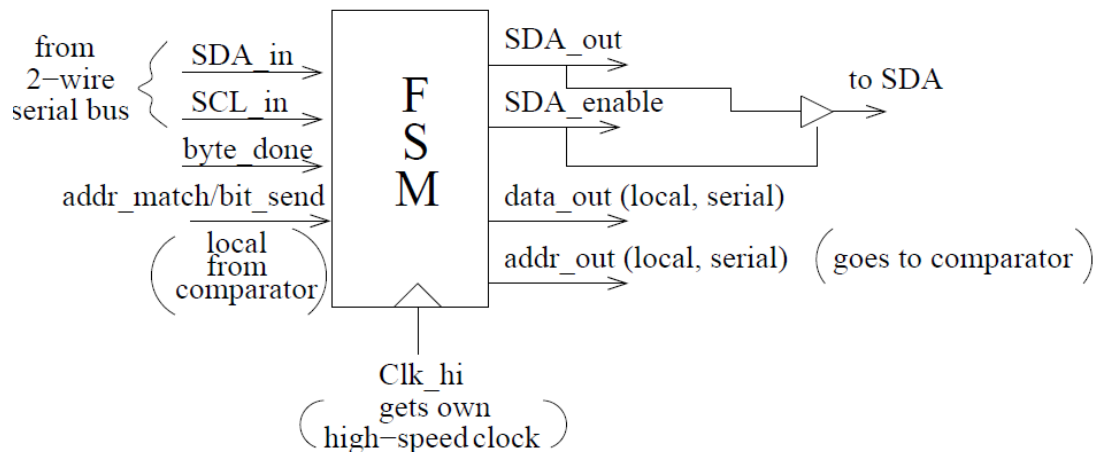
Prof. Steven Nowick

HW5 Designing a Slave Controller for the Philips I2C Bus Protocol

You can view and download our whole project in <https://github.com/xuhua/HW5>

or I can add your RSA public key to access our project at git@github.com:xuhua/HW5.git

We assign code as following,



MSB LSB
 { SDA_in, SCL_in, Byte_done, addr_match/bit_send }

MSB LSB
 { SDA_out, SDA_enable, data_out, addr_out }

(i) The symbolic state diagram of the Moore FSM is attached to this file.

(ii) The completed VHDL.

The way we implement this FSM as we firstly draw the state signal diagram out separately using master to slave and slave to master protocols.

It could be easily find out the state of none matched device. So we are not include it in diagram.

We attach these diagram alone with this file.

When slave receive bits from master, we send out the address and data in the low clock cycle to make sure it is stable to local comparator and data device. The reason why we do this, is because there will be a state that could give a new number to comparator just before stop signal.

And our design could handle unexpected exit when master send stop signal to slave when sending data.

There are some important points we have in our assignment.

1) Slave act as receiver

A diagram of time sequence table is attached with the documents.

(2) Slave act as sender

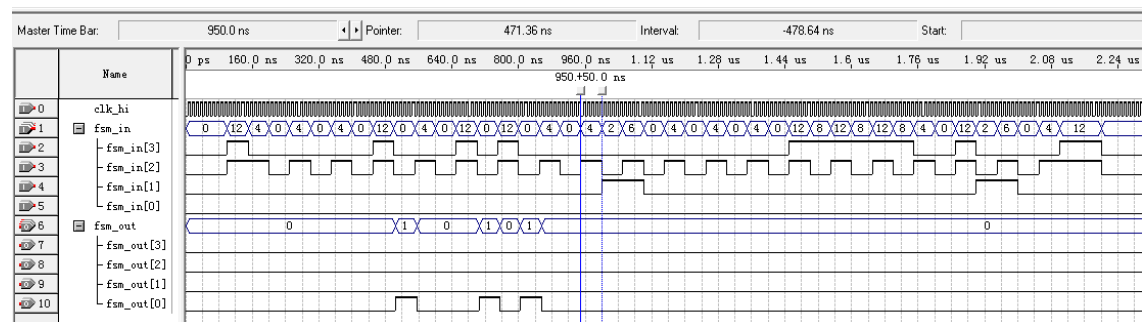
A diagram of time sequence table is attached with the documents.

Here are signal diagram of our simulations.

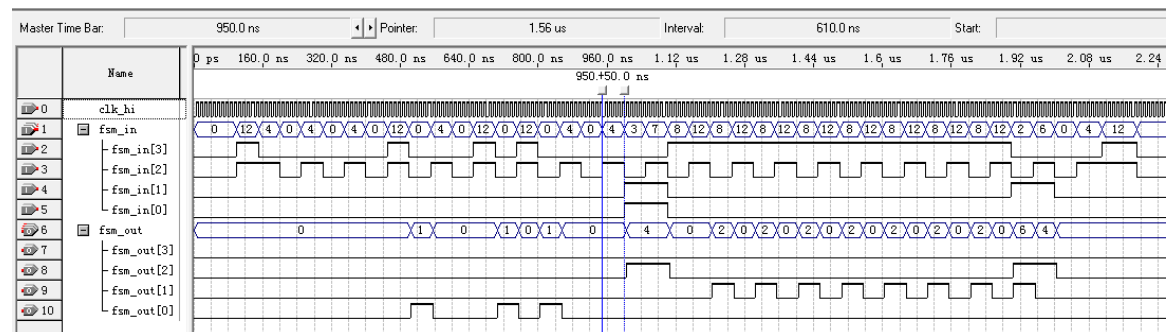
And texted diagram is attached to this document. There are more information on those papers.

slave receiver

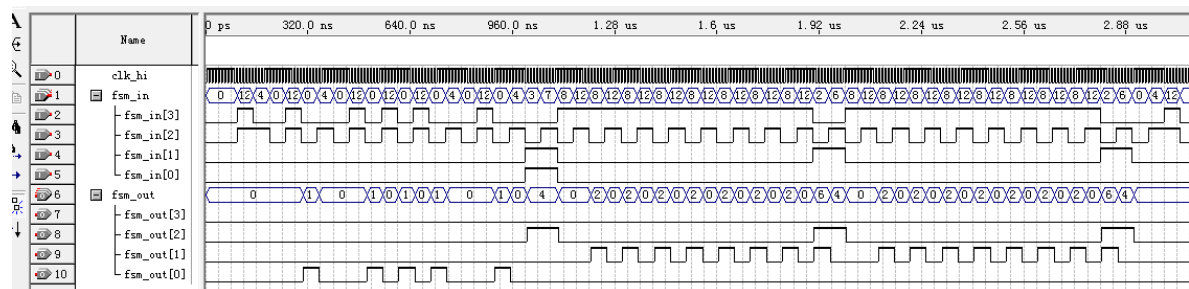
(i)



(ii)

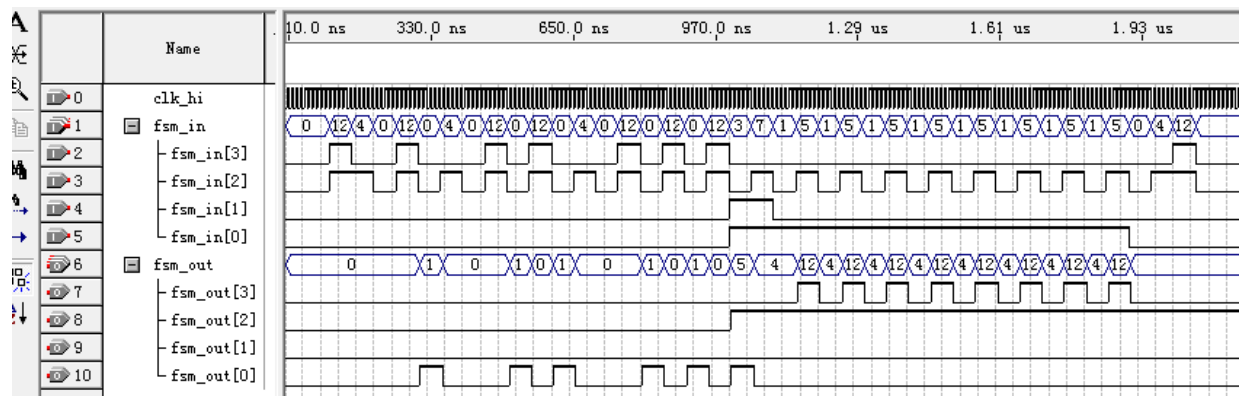


(iii)

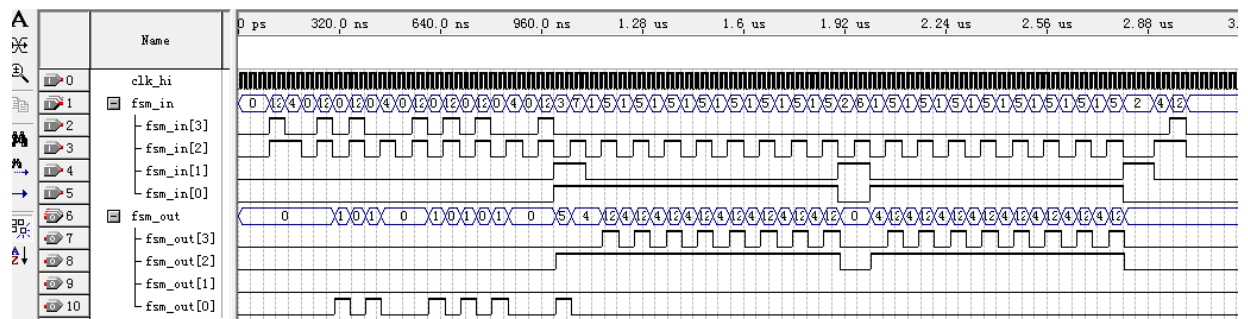


Slave as transmitter

(iv)



(v)



(vi) The signal diagram below shows our FSM can handle unexpected stop signal during data transmission. (Master to slave)

