Development and Research of Different Architectures of I²C Bus Controller

Speaker: Vasiliev Eugeniy
Master Student in MIET-Synopsys Educational Center

I²C is a serial interface for communication of the integrated schemes, developed by Philips in the early eighties as the simple bus of internal communication for operating electronics. I²C uses two bidirectional lines with an open drain - a serial line of data (SDA, English and a serial line of clocking (SCL, *Serial CLock*), both pulled up with resistors.

As a rule, interface **I²C** is used for connection of low-speed peripheral components to a motherboard, built-in systems and mobile phones. There is a number of interfaces which also can be used for these purposes, however they have advantages and disadvantages in comparison with **I²C**. **I²C** is attractive due to its simplicity. Low speed of data transmission is compensated by economy of hardware resources. Magistral organization provides simple bus wiring. Serial digital interfaces, which can be used as alternatives to **I²C**, are:

- SPI
- UART
- CAN
- 1-Wire

All transactions on **I²C** bus begin with a START (S) and can be terminated by a STOP (P). A HIGH to LOW transition on the SDA line while SCL is HIGH defines a START condition. A LOW to HIGH transition on the SDA line while SCL is HIGH defines a STOP condition. START and STOP conditions are always generated by the master. For initialization of data transmission master device generates START condition or Repeated Start condition (if the previous data transmission has not been finished by STOP condition), and then sends first 8 bits (the address and a direction bit ($R/\overline{W}$ bit)). After each 8 bits transferred through the bus, receiver device should send

an acknowledgement (ACK), LOW level of SDA line during SCL pulse (SCL HIGH). If in the end of transmission master device which initialized the transmission wishes to stop data transfer process, it generates not acknowledgement (NACK)

Controller of **I²C** bus – is a device that performs interaction with **I²C** bus: transmits and receives data to and from a bus, analyzing conditions of **I²C** bus and transferring data to or from **I²C** bus in serial form received or transmitted from or to a connected device in parallel form. Features of **I²C** -bus protocol, such as START condition, STOP condition, Acknowledge, Synchronization, Arbitration etc. which are chosen for required configuration are usually built in **I²C** controller.

Design of **I²C** slave controller requires choosing a way of converting data from parallel to a serial form and vise versa, accounting frame format. Methods, which are usually used for this are Mealy state machine (counter plus register plus additional logic) and shift register (shift register itself plus additional logic), last way is more attractive. But this task can also be solved using combinational logic to access registers storing transmitted and received bytes. This logic can be built on multiplexers and demultiplexers. All these ways of operating with frames of **I²C** bus are implemented in following architectures:

- **State machine architecture**
- **Demultiplexer/Multiplexer chains architecture**
- **Two shift registers architecture**

Described architectures were used for designing 3 different slave controller devices, which have the same ports, and perform the same functions. All devices were described using Verilog HDL, verified and then synthesized using Synopsys Design Compiler in basis of 0.18 um CMOS technology. Powers and areas of devices obtained after logical synthesis are presented on following pictures: