Serial Communication

SPI & UART

Sai Ashish[1] Havish[2]

[1]EE16BTECH11043

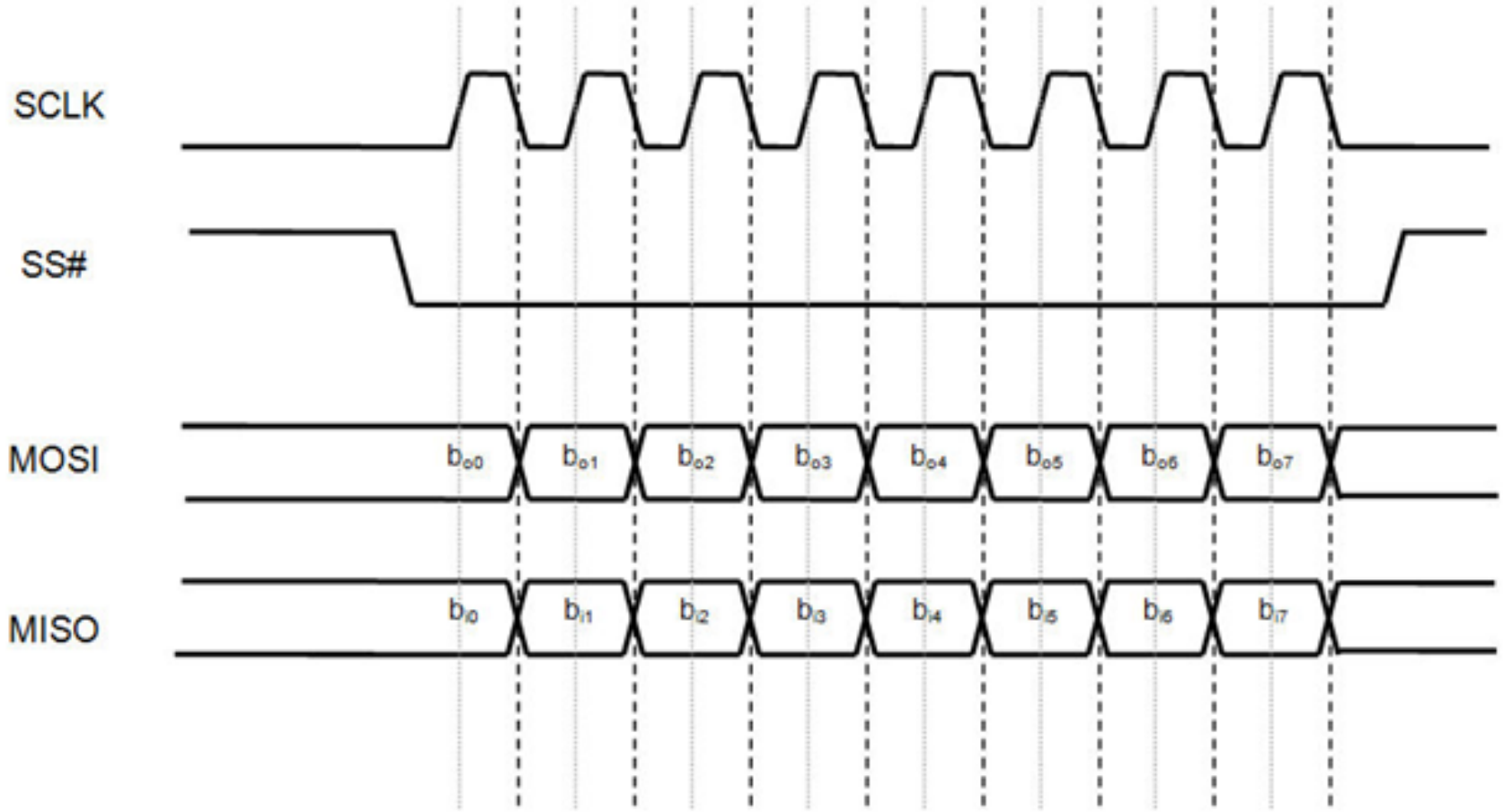
[2]EE16BTECH11023

April 25, 2019

SPI Protocol

- SPI stands for serial peripheral interface.
- Some times SPI protocol is also called a 4-wire protocol. It requires four wires MOSI, MISO, SS, and SCLK.SPI protocol used to communicate the master and slave devices.
- The master first configures the clock using a frequency. The master then selects the particular slave device for communication by pulling the chip select button. That particular device is selected and starts the communication between master and that particular slave. The master select only one slave at a time.
- It is full duplex communication protocol. Not limited to 8 bit words in the case of bit transferring.

SPI Protocol



IMPLEMENTATION

- Used FPGA (ICOBOARD) as master.
- Used two arduinos as slaves.
- In our implementation we sent data from master to slaves (MOSI active; MISO inactive). In the similar way we can modify the code to send data from slave to Master.
- The clock frequency has been set to 2MHz
- We sent an 8 bit integer from the master to the slaves by selecting one slave at a time.

UART

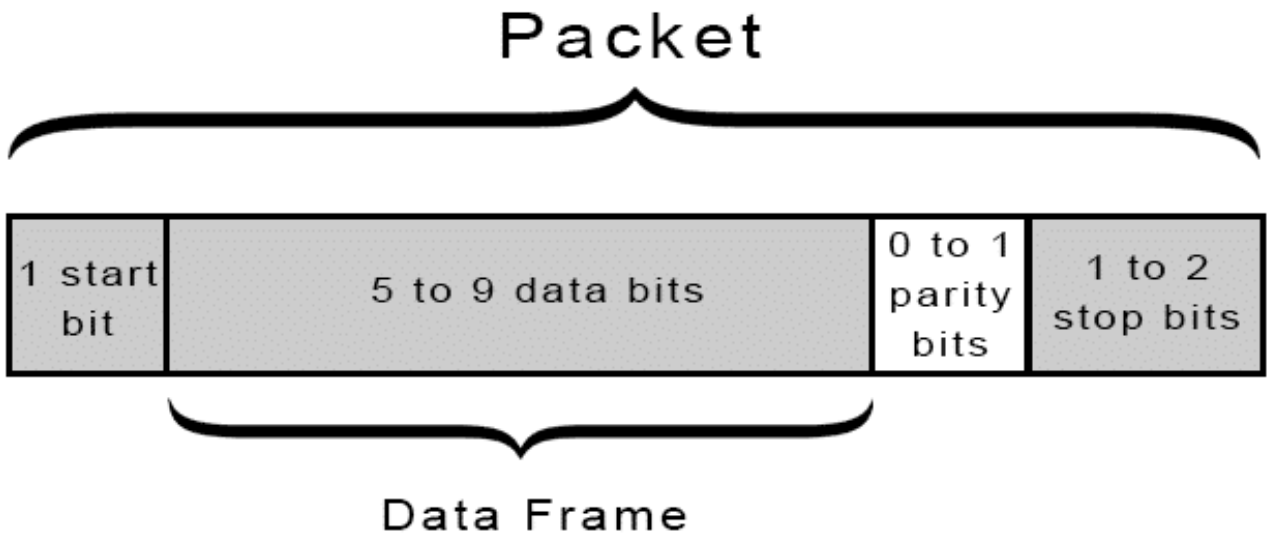
- UART or Universal Asynchronous Receiver Transmitter is a serial communication device that performs parallel to serial data conversion at the transmitter side and serial to parallel data conversion at the receiver side. It is universal because the parameters like transfer speed, data speed, etc. are configurable.
- There is no clock used to synchronize the data transmission between sender and the receiver, rather there is something known as baud rate. Baud Rate is measured in bits per second.
- This is the serial communication technique used in serial data transmission in Arduino and various other microcontrollers to various sensors and vice-a-versa.

The Communication protocol

- **Start Bit**: Start bit is a synchronisation bit that is added before the actual data. Start bit marks the beginning of the data packet. In order to start the data transfer, the transmitting UART pulls the data line from high voltage level to low voltage level (from 1 to 0).
- **Stop Bit**: The Stop Bit, as the name suggests, marks the end of the data packet. In order to end the transmission, the UART maintains the data line at high voltage (1).

The Communication protocol

- **Parity Bit**: Parity check bit, it is a channel coding technique to correct the transmitted (serial)data bits in case of any errors while it's transmission over the channel(wire).
- **Data Bits**: Data bits are the actual data being transmitted from sender to receiver.



IMPLEMENTATION

- The FPGA(ICOBOARD) acts as transmitter and the arduino acts as a receiver.
- We implemented this as a simple 4 state FSM which transitions betwen the states IDLE, START, DATA and STOP state. The states are self explanatory.
- The first part of the code sets the required baud rate for communication.
- We sent a ascii byte which represents an alphabet from FPGA to arduino.