Deepinder Singh & Matthew Beard

**Mini Project SPART Report**

The SPART module allows for communication between a keyboard and a computer monitor that run on asynchronous clock cycles by transmitting and data at a rate consistent with the sixteen times the BAUD rate. This module works with 4 BAUD rates, 4800 b/s, 9600b/s, 19200b/s and 38400b/s, which can be specified using DIP switches 7(LSB) and 8(MSB) on the Virtex-5 board. Our SPART design contains multiple modules which are used to make it function correctly.

The bus interface module (BI.v) assigns value to the status register that were necessary for our simple driver, as well as controlling logic for what should be loaded into the bidirectional data bus between the driver and the SPART.

The BAUD rate generator module (BAUD.v) generates an enable signal that goes high at times which are consistent with the BAUD rate. A counter is set with the number in the divisor buffer and then counts down to zero at every clock cycle, where it resets back to the divisor number. This is done as a method of using the clock to generate the slower BAUD rate signal.It will only start generating when after both pieces of the divisor buffer are set.

The receive module (Receive.v) handles the logic required to receive data from the keyboard. When the module detects a start bit coming in from rxd, it begins loading the receive buffer with a new bit every time it senses an enable signal coming from the BAUD generator. When the buffer s filled with the new data, it sends a signal to the driver telling it about the new data and sends it via the bus interface.

The transmit module (transmit.v) handle the logic associated with sending data from the drive out to the monitor. When the driver indicates that it is ready to transmit data, it sends the data that needs to be transmitted through the bus interface and into the transmit shift register. That data is then sent out one bit at a time through txd, with a new bit being transmitted every time it sees and enable from the BAUD generator.

In order to test our SPART design, we devised a test bench (top\_level\_tb.sv) which would first load the devisor buffer of the BAUD rate generator, and then supply a random bit vector into the receive module and then be echoed out of the processor. Our bench sends 10 random bit vectors into the receive buffer and checks if the same bit vector is returned through the transmit buffer.

There were two major dilemmas that we encountered on the regarding our transmit module and BAUD rate divisor. Our transmit module encountered a bug where the signal used to determine when the block was ready to transmit was being set incorrectly. This problem was due to a line of code where we set the transmit signal being out of place. When moved to its correct spot (moved from an if statement to its subsequent else statement) the transmit logic worked as expected. The error in with our BAUD rate divisor we not have as good of an understanding of. We were getting a bug where when we typed on the keyboard, vastly different characters would sometimes get an incorrect character. We found through trial and error that reducing the value we loaded into the divisor buffer would drastically increase the accuracy of the of the SPART communication.