```
        /*
         * I2C Slave RX States
         */

        case slaveAddressReceived: {
            uart0_puts("Entered state 0x60");
            mpI2CRegs->I2CONSET = 0x04;
            clearSIFlag();
            break;
        }

        case slaveDataReceived: {
            uart0_puts("Entered state 0x80");
            if (isFirst80) { //Register number is received
                isFirst80 = false;
                mTransaction.firstReg = mpI2CRegs->I2DAT;
            }
            else {
//                  if ((mTransaction.firstReg - *mTransaction.pMasterData) + write_counter + 1 <= mTransaction.trxSize){
//                      *(mTransaction.pMasterData + mTransaction.firstReg + write_counter++) = mpI2CRegs->I2DAT;
//                  }
//                  else {
//                      uart0_puts("buffsploit prevented");
//                  }
                *(mTransaction.pMasterData + mTransaction.firstReg + write_counter++) = mpI2CRegs->I2DAT;
            }
            clearSIFlag();
            break;
        }

        case slaveStoporRptStartRecv: {
            uart0_puts("Entered state 0xA0");
            isFirst80 = true;
            write_counter = 0;
            mpI2CRegs->I2CONSET = 0x04;
            clearSIFlag();
            break;
        }

        /*
         * I2C Slave TX States
         */
        case slaveDataSend: {
            uart0_puts("Entered State 0xA8");
            if (read_counter + 1 <= mTransaction.trxSize){
                mpI2CRegs->I2DAT = *(mTransaction.pMasterData + mTransaction.firstReg + read_counter++);
            }
            else {
                uart0_puts("Read too far");
            }
            mpI2CRegs->I2CONSET = 0x04;
            clearSIFlag();
            break;
        }

        case dataAckedByMaster: {
            uart0_puts("Entered State 0xB8");
            if (read_counter + 1 <= mTransaction.trxSize){
                mpI2CRegs->I2DAT = *(mTransaction.pMasterData + mTransaction.firstReg + read_counter++);
            }
            mpI2CRegs->I2CONSET = 0x04;
            clearSIFlag();
            break;
        }

        case masterNackRecv: {
            uart0_puts("Entered State 0xC0");
            read_counter = 0;
            mpI2CRegs->I2CONSET = 0x04;
            clearSIFlag();
            break;
        }
```