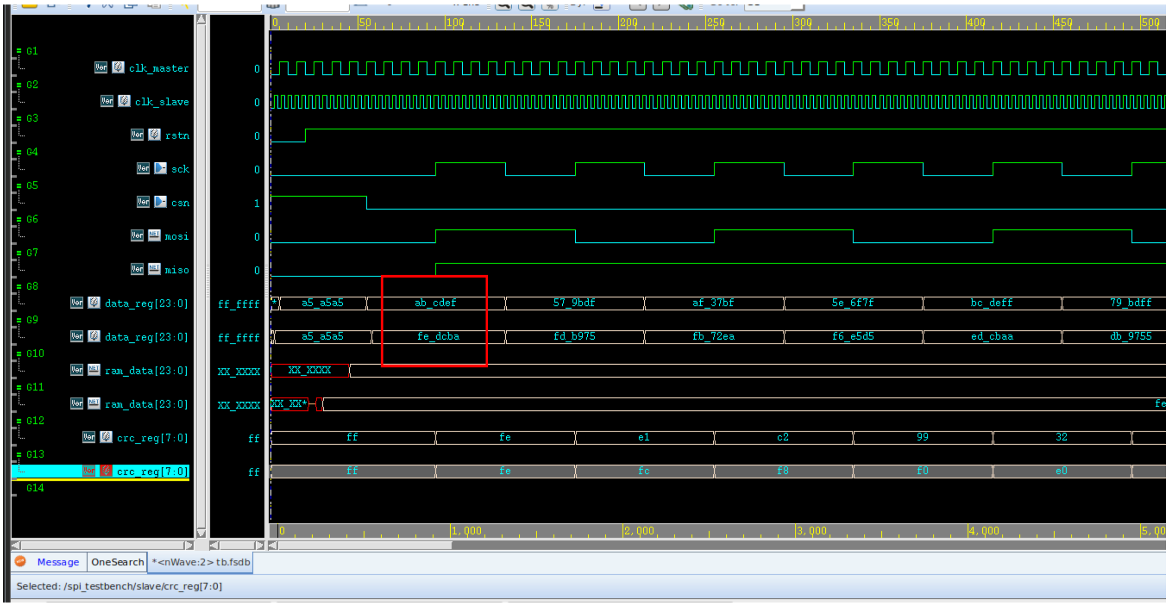
Completion items

1. SPI transmission
2. Dynamically generating CRC checksums
3. Reading data from memory and storing data in memory

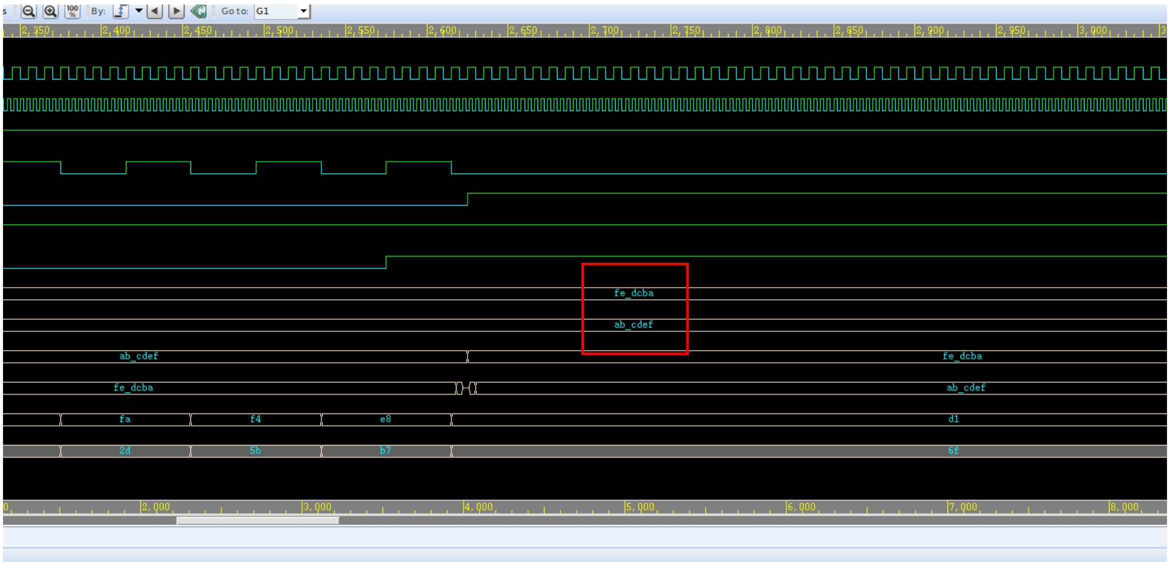
simulation waveforms

1， Beginning Stage



Selected: /spi_testbench/slave/crc_reg[7:0]

2， Ending Stage



It can be seen that the master and slave devices are still able to successfully transmit data to each other at different clock frequencies.

Code Summary

spi_master_top.v

1. Module Composition

SPI Clock Generator Module (spi_clk_gen)

Generates the SPI clock sck with a frequency derived from the input clock clk and a division factor.

Stops the clock when the chip select signal csn is high.

Single-Port Synchronous RAM (single_port_sync_ram_master)

Stores SPI data and supports read/write operations.

Address width: 5 bits, Data width: 24 bits, Depth: 32.

Initialization: Address 0 is set to 0xABCDEF, all other addresses are set to 0.

SPI Master Top Module (spi_master_top)

Controls SPI communication, interacts with RAM, handles data transmission, and performs CRC checks.

Includes a state machine, data registers, CRC logic, and RAM control signals.

2. State Machine Logic

IDLE: Idle state, SPI stops working, waiting for a start condition.

WAIT: Reads data from RAM, preparing for communication.

START: Activates SPI communication by pulling the chip select signal csn low.

TRANSFER: Sends data and CRC values bit by bit while receiving data from the slave.

Completion: Verifies CRC, writes data back to RAM.

3. Data Transmission

Transmit (MOSI): Sends data on the rising edge of the clock and dynamically calculates CRC.

Receive (MISO): Receives data on the falling edge of the clock and updates CRC.

4. CRC Check

Dynamically generates CRC during transmission and recalculates it upon reception for verification.

5. Overall Workflow

a) Initialize registers and RAM.

b) Read data from RAM and start SPI communication.

c) Transmit and receive data bit by bit, then verify CRC.

d) Write the processed data back to RAM.

spi_slave_top.v

1. Module Composition

Single-Port Synchronous RAM (single_port_sync_ram_slave)

Stores SPI data and supports read/write operations.

Address width: 5 bits, Data width: 24 bits, Depth: 32.

Initialization: Address 0 is set to 0xFEDCBA, all other addresses are set to 0.

SPI Slave Top Module (spi_slave_top)

Handles SPI communication, interacts with RAM, performs data transmission, and CRC checks.

Includes a state machine, data registers, CRC logic, and RAM control signals.

2. State Machine Logic

IDLE: Idle state, resets RAM control signals and waits for communication to start.

WAIT: Reads data from RAM into data_reg, then waits one clock cycle for RAM data to stabilize.

START: Waits for the chip select signal csn to go low, then prepares for data transfer.

TRANSFER:

Sends data and CRC values bit by bit via so (MISO).

Receives data and CRC values bit by bit via si (MOSI).

After transfer, verifies CRC and writes data back to RAM.

3. Data Transmission

Transmit (MISO):

On the rising edge of sck, sends data from data_reg and dynamically updates the CRC register.

After sending 24 bits of data, sends the CRC value.

Receive (MOSI):

On the falling edge of sck, receives data into data_reg and updates the CRC register.

After receiving 24 bits of data, receives the CRC value.

4. CRC Check

CRC Polynomial: SAE-J1850 (8'h1D).

Transmit Side: Dynamically calculates CRC during data transmission.

Receive Side: Recalculates CRC after receiving data and compares it with the received CRC.

If mismatched, outputs a CRC error message.

If matched, outputs a success message.

5. Overall Workflow

a) Initialize registers and RAM.
b) Wait for the chip select signal csn to go low.
c) Read data from RAM and prepare for SPI communication.

d) Transmit and receive data bit by bit, including CRC.

e) Verify CRC and write the processed data back to RAM.

Testbench Summary

1. **Clock Generation:**

   Generates clocks for the master (clk_master) and slave (clk_slave).

2. **Module Instantiation:**

   Instantiates the SPI master and slave modules, connecting their SPI signals (sck, csn, mosi, miso).

3. **Initialization:**

   Resets the system (rstn) for 20 ns.

   Optionally initializes RAM data for both master and slave (commented out).

4. **Simulation:**

   Runs the simulation for a sufficient time to allow communication.

   Outputs a message and generates waveform files (tb.fsdb) for verification.