Texas A&M University

ECEN-719 Lab6 Report

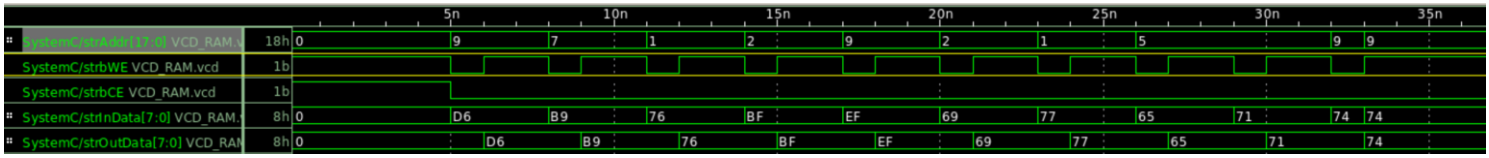
NAME: Hai-Ming, Hsu

UID: 433004106

Section: 602

Department of Electrical & Computer Engineering

1. Screenshots of the waveform with analysis



Part 5ns to 20ns is Verification 1, Addr & InData is randomized as required.
Addr & InData is reloaded per test cycle.
Addrs are constrained between 0 to 10. InData >80.

Part 20ns to 35ns is Verification 2, Addr & InData is randomized as required.
Indata are constrained between 100 to 120 with the weight of 5% / 95%.

2. Screenshots of the simulation output in Vista

```
200 Info: (I804) /IEEE_Std_1666/deprecated: sc_start(double) deprecated, use sc_start(sc_time) or sc_start()
201 Merging /home/grads/h/haiminghsu/ECEN719/lab6/src/Project/build/D_PRJDIR_/test_RAM.exe ...Done.
202 Saving types data file /home/grads/h/haiminghsu/ECEN719/lab6/src/Project/sim/test_RAM.db...Done.
203 WARNING: Default time step is used for VCD tracing.
204
205 Info: (I804) /IEEE_Std_1666/deprecated: sc_get_curr_process_handle deprecated use sc_get_current_process_handl
206 ------------------- Beginning of Verification I
207
208 *** SCV_WARNING: CONSTRAINT_WARNING_IGNORE_SOFT_CONSTRAINT at time 5 ns in process <main>
209     Soft constraints for over-constrained  object 'Pkt_Test' will be ignored.
210
211
212 *** SCV_ERROR: CONSTRAINT_ERROR_OVER_CONSTRAINED at time 5 ns in process <main>
213     Constraints for over-constrained object 'Pkt_Test' will be ignored.
214
215 *** SCV_ERROR: CONSTRAINT_ERROR_OVER_CONSTRAINED at time 5 ns in process <main>
216     Constraints for over-constrained object 'Pkt_Test' will be ignored.
217 Pkt_constraint Name: Pkt_Test
218   Hard constraints: (((1&&(default<262144))&&(default<256))&&1)
219   Soft constraints: (1&&1)
220   Number of elements: 3
221   Current value of elements:
222     :  9
223     :  214
224     :  214
225
226 *** SCV_ERROR: CONSTRAINT_ERROR_OVER_CONSTRAINED at time 8 ns in process <main>
227     Constraints for over-constrained object 'Pkt_Test' will be ignored.
228 Pkt_constraint Name: Pkt_Test
229   Hard constraints: (((1&&(default<262144))&&(default<256))&&1)
230   Soft constraints: (1&&1)
231   Number of elements: 3
232   Current value of elements:
233     :  7
234     :  185
235     :  185
236
237 *** SCV_ERROR: CONSTRAINT_ERROR_OVER_CONSTRAINED at time 11 ns in process <main>
238     Constraints for over-constrained object 'Pkt_Test' will be ignored.
```

```
*** SCV_ERROR: CONSTRAINT_ERROR_OVER_CONSTRAINED at time 11 ns in process <main>
    Constraints for over-constrained object 'Pkt_Test' will be ignored.
Pkt_constraint Name: Pkt_Test
  Hard constraints: (((1&&(default<262144))&&(default<256))&&1)
  Soft constraints: (1&&1)
  Number of elements: 3
  Current value of elements:
    :  1
    :  118
    :  118

*** SCV_ERROR: CONSTRAINT_ERROR_OVER_CONSTRAINED at time 14 ns in process <main>
    Constraints for over-constrained object 'Pkt_Test' will be ignored.
Pkt_constraint Name: Pkt_Test
  Hard constraints: (((1&&(default<262144))&&(default<256))&&1)
  Soft constraints: (1&&1)
  Number of elements: 3
  Current value of elements:
    :  2
    :  191
    :  191

*** SCV_ERROR: CONSTRAINT_ERROR_OVER_CONSTRAINED at time 17 ns in process <main>
    Constraints for over-constrained object 'Pkt_Test' will be ignored.
Pkt_constraint Name: Pkt_Test
  Hard constraints: (((1&&(default<262144))&&(default<256))&&1)
  Soft constraints: (1&&1)
  Number of elements: 3
  Current value of elements:
    :  9
    :  239
    :  239
------------------- End of Verification I

------------------- Beginning of Verification II

*** SCV_ERROR: CONSTRAINT_ERROR_OVER_CONSTRAINED at time 20 ns in process <main>
    Constraints for over-constrained object 'Pkt_Test' will be ignored.
Pkt_constraint Name: Pkt_Test
  Hard constraints: (((1&&(default<262144))&&(default<256))&&1)
  Soft constraints: (1&&1)
  Number of elements: 3
  Current value of elements:
    :  2
    :  105
    :  105

*** SCV_ERROR: CONSTRAINT_ERROR_OVER_CONSTRAINED at time 23 ns in process <main>
    Constraints for over-constrained object 'Pkt_Test' will be ignored.
Pkt_constraint Name: Pkt_Test
  Hard constraints: (((1&&(default<262144))&&(default<256))&&1)
  Soft constraints: (1&&1)
  Number of elements: 3
  Current value of elements:
    :  1
    :  119
    :  119

*** SCV_ERROR: CONSTRAINT_ERROR_OVER_CONSTRAINED at time 26 ns in process <main>
    Constraints for over-constrained object 'Pkt_Test' will be ignored.
Pkt_constraint Name: Pkt_Test
  Hard constraints: (((1&&(default<262144))&&(default<256))&&1)
  Soft constraints: (1&&1)
  Number of elements: 3
  Current value of elements:
    :  5
    :  101
    :  101

*** SCV_ERROR: CONSTRAINT_ERROR_OVER_CONSTRAINED at time 29 ns in process <main>
    Constraints for over-constrained object 'Pkt_Test' will be ignored.
Pkt_constraint Name: Pkt_Test
  Hard constraints: (((1&&(default<262144))&&(default<256))&&1)
  Soft constraints: (1&&1)
  Number of elements: 3
  Current value of elements:
    :  5
    :  113
    :  113
```

```
316
317*** SCV_ERROR: CONSTRAINT_ERROR_OVER_CONSTRAINED at time 32 ns in process <main>
318    Constraints for over-constrained object 'Pkt_Test' will be ignored.
319Pkt_constraint Name: Pkt_Test
320  Hard constraints: (((1&&(default<262144))&&(default<256))&&1)
321  Soft constraints: (1&&1)
322  Number of elements: 3
323  Current value of elements:
324    :  9
325    :  116
326    :  116
327------------------- End of Verification II
```

3. *Screenshots of your code in this design with reasonable comments*

```
#include "systemc.h"
#include "RAM.cpp"
#include <scv.h>

class Pkt_constraint : virtual public scv_constraint_base
{
    public:
    scv_smart_ptr<sc_uint<ADDR_WIDTH> > sAddr;
    scv_smart_ptr<sc_uint<DATA_WIDTH> > sInData;
    scv_smart_ptr<sc_uint<DATA_WIDTH> > sOutData;

    SCV_CONSTRAINT_CTOR(Pkt_constraint) {
        // define constraints
        SCV_CONSTRAINT( sAddr() < (1<<ADDR_WIDTH));

        SCV_CONSTRAINT( sInData() < (1<<DATA_WIDTH));
        }
};
```

```
int sc_main (int argc, char* argv[]) {

    // Declare Input/Output Signals
    sc_signal < sc_uint<ADDR_WIDTH> >   tAddr ;
    sc_signal < bool >          tbWE;
    sc_signal < bool >          tbCE;
    sc_signal < sc_uint<DATA_WIDTH> >   tInData;
    sc_signal < sc_uint<DATA_WIDTH> >   tOutData;

    int i = 0;

    // Connect the DUT(Design Under Test)
    SRAM RAM_01("SIMULATION_RAM");
        RAM_01.InData(tInData);
            RAM_01.Addr(tAddr);
        RAM_01.bCE(tbCE);
        RAM_01.bWE(tbWE);
        RAM_01.OutData(tOutData);

    // Open VCD(Value Change Dump) file
    sc_trace_file *wf = sc_create_vcd_trace_file("VCD_RAM");

    // Dump the desired signals
    sc_trace(wf, tInData, "strInData");
        sc_trace(wf, tAddr, "strAddr");
    sc_trace(wf, tbCE, "strbCE");
    sc_trace(wf, tbWE, "strbWE");
    sc_trace(wf, tOutData, "strOutData");

    // Initialize all variables
    tbCE.write(1);
    tbWE.write(1);
    sc_start(5);

    Pkt_constraint cPkt("Pkt_Test");

    typedef pair < sc_uint<ADDR_WIDTH>,sc_uint<ADDR_WIDTH> > addr_range;
    scv_bag<addr_range> addr_list;
    addr_list.add(addr_range(1,9),100);

    typedef pair < sc_uint<DATA_WIDTH>,sc_uint<DATA_WIDTH> > data_range;
    scv_bag<data_range> data_list1;
    data_list1.add(data_range(80,(1<<DATA_WIDTH)-1),100);
```

```
        // Verification I
        cout << "------------------- Beginning of Verification I" << endl;
        for(i=0; i<5; i++){
          // Set writing mode for RAM
            tbCE.write(0);
            tbWE.write(0);
          // Generate values for tInData & tAddr using "cPkt"
            cPkt.sAddr->set_mode(addr_list);
            cPkt.sInData->set_mode(data_list1);
            cPkt.next();
            tAddr.write(cPkt.sAddr->read());
            tInData.write(cPkt.sInData->read());
            sc_start(1);
          // Set reading mode for RAM
            tbCE.write(0);
            tbWE.write(1);
            sc_start(1);
          // Data read from tOutData, save to cPkt.sOutData
            *cPkt.sOutData = tOutData.read();
          // Print statistics
            cPkt.print();
            sc_start(1);
        }
        cout << "------------------- End of Verification I" << endl << endl;

        // Verification II
        cout << "------------------- Beginning of Verification II" << endl;
        for(i=0; i<5; i++){
          // Set writing mode for RAM
            tbCE.write(0);
            tbWE.write(0);
          // Set range distribution
        scv_bag<data_range> data_list2;
            data_list2.add(data_range(80,99),5);
            data_list2.add(data_range(100,120),95);
      // Generate values of tInData  using "cPkt"
            cPkt.sAddr->set_mode(addr_list);
            cPkt.sInData->set_mode(data_list2);
            cPkt.next();
            tAddr.write(cPkt.sAddr->read());
            tInData.write(cPkt.sInData->read());
            sc_start(1);
          // Set reading mode for RAM
            tbCE.write(0);
            tbWE.write(1);
            sc_start(1);
          // Data read from tOutData, save to cPkt.sOutData
            *cPkt.sOutData = tOutData.read();
          // Print statistics
            cPkt.print();
            sc_start(1);
        }
        cout << "------------------- End of Verification II" << endl;

    // Close trace file
    sc_close_vcd_trace_file(wf);

    return 0;   // Terminate simulation
}
```