```
//=========================================
// Function : Asynchronous SRAM
//=========================================
#include "systemc.h"

#define DATA_WIDTH        8
#define ADDR_WIDTH        18
#define RAM_DEPTH         1 << ADDR_WIDTH

SC_MODULE (RAM) {
  // ----- Declare Input/Output ports -----
    sc_in < sc_uint<ADDR_WIDTH> >       Addr ;
    sc_in < bool >                      bWE;
    sc_in < bool >                      bCE;
    sc_in < sc_uint<DATA_WIDTH> >       InData;
    sc_out < sc_uint<DATA_WIDTH> >      OutData;

  // ----- Internal variables -----
  // ...
  sc_bv<DATA_WIDTH> internal_memory[RAM_DEPTH]; // 2D bit vector of our data
  int location_row; // we'll use this to store where we are read/write from

  // ----- Code Starts Here -----

  void function_decode(){
    sc_uint<ADDR_WIDTH> read_address = Addr.read();
    location_row =  read_address;
  }

  // Memory Write Block
  // Write Operation : When we_b = 0, ce_b = 0
  void function_write(){
      if(!(bWE.read()) && !(bCE.read())){
          function_decode();
          cout << "Writing " << InData.read() << " to location " << location_row << endl;
          internal_memory[location_row] = InData.read();
      }
  }

  // Memory Read Block
  // Read Operation : When we_b = 1, ce_b = 0
  void function_read(){
      if((bWE.read()) && !(bCE.read())){
          function_decode();
          cout << "Reading " << internal_memory[location_row] << " from " << location_row
 << endl;
          OutData.write(internal_memory[location_row]);
      }
  }


  // ----- Constructor for the SC_MODULE -----
  // sensitivity list
  SC_CTOR(RAM) {
    SC_METHOD(function_write);
        // recompute if bWE/bCE/Addr/InData change
        sensitive << bWE << bCE << Addr << InData;
    SC_METHOD(function_read);
        // recompute if bWE/bCE/Addr change
        sensitive << bWE << bCE << Addr;
  }
};
```

```
cd /home/ugrads/s/seth.barberee/ECEN468/Lab1/SRC/
gdb.shell.exe
spawn /opt/coe/mentorgraphics/vista312/linux64/tools.64bit/bin/gdb -quiet -interp=opengdb
 -runtcl="/opt/coe/mentorgraphics/vista312/generic/tcl/v2/gdb/gdb.tcl" --nx

(vista) BFD: /usr/lib64/libm.so.6: invalid relocation type 37
BFD: BFD 2.17.50 20061115 assertion fail elf64-x86-64.c:259
BFD: /usr/lib64/libm.so.6: invalid relocation type 37
BFD: BFD 2.17.50 20061115 assertion fail elf64-x86-64.c:259
BFD: /usr/lib64/libm.so.6: invalid relocation type 37
BFD: BFD 2.17.50 20061115 assertion fail elf64-x86-64.c:259
BFD: /usr/lib64/libm.so.6: invalid relocation type 37
BFD: BFD 2.17.50 20061115 assertion fail elf64-x86-64.c:259
BFD: /usr/lib64/libm.so.6: invalid relocation type 37
BFD: BFD 2.17.50 20061115 assertion fail elf64-x86-64.c:259
BFD: /usr/lib64/libm.so.6: invalid relocation type 37
BFD: BFD 2.17.50 20061115 assertion fail elf64-x86-64.c:259
BFD: /usr/lib64/libm.so.6: invalid relocation type 37
BFD: BFD 2.17.50 20061115 assertion fail elf64-x86-64.c:259
BFD: /usr/lib64/libm.so.6: invalid relocation type 37
BFD: BFD 2.17.50 20061115 assertion fail elf64-x86-64.c:259
BFD: /usr/lib64/libm.so.6: invalid relocation type 37
BFD: BFD 2.17.50 20061115 assertion fail elf64-x86-64.c:259
BFD: /usr/lib64/libm.so.6: invalid relocation type 37
BFD: BFD 2.17.50 20061115 assertion fail elf64-x86-64.c:259
BFD: /usr/lib64/libm.so.6: invalid relocation type 37
BFD: BFD 2.17.50 20061115 assertion fail elf64-x86-64.c:259
BFD: /usr/lib64/libm.so.6: invalid relocation type 37
BFD: BFD 2.17.50 20061115 assertion fail elf64-x86-64.c:259
BFD: /usr/lib64/libc.so.6: invalid relocation type 37
BFD: BFD 2.17.50 20061115 assertion fail elf64-x86-64.c:259
BFD: /usr/lib64/libc.so.6: invalid relocation type 37
BFD: BFD 2.17.50 20061115 assertion fail elf64-x86-64.c:259
BFD: /usr/lib64/libc.so.6: invalid relocation type 37
BFD: BFD 2.17.50 20061115 assertion fail elf64-x86-64.c:259
BFD: /usr/lib64/libc.so.6: invalid relocation type 37
BFD: BFD 2.17.50 20061115 assertion fail elf64-x86-64.c:259
BFD: /usr/lib64/libc.so.6: invalid relocation type 37
BFD: BFD 2.17.50 20061115 assertion fail elf64-x86-64.c:259
BFD: /usr/lib64/libc.so.6: invalid relocation type 37
BFD: BFD 2.17.50 20061115 assertion fail elf64-x86-64.c:259
BFD: /usr/lib64/libc.so.6: invalid relocation type 37
BFD: BFD 2.17.50 20061115 assertion fail elf64-x86-64.c:259
Vista SystemC 2.2 Runtime Kernel.
Built September 15, 2011. License version 2011.9.
Copyright (c) 2005-2011, Mentor Graphics Corporation.

             SystemC 2.2.0 --- Sep 15 2011 00:24:25
        Copyright (c) 1996-2006 by all Contributors
                    ALL RIGHTS RESERVED

Info: (I804) /IEEE_Std_1666/deprecated: sc_start(double) deprecated, use sc_start(sc_time
) or sc_start()
WARNING: Default time step is used for VCD tracing.
@5 ns:: Write mode
Writing 61 to location 61
@10 ns:: Write mode
Writing 62 to location 62
@15 ns:: Write mode
Writing 63 to location 63
@20 ns:: Set to Idle mode
@25 ns:: Read mode
Reading 00111101 from 61
@30 ns:: Read mode
Reading 00111110 from 62
@35 ns:: Read mode
Reading 00111111 from 63

Info: (I804) /IEEE_Std_1666/deprecated: You can turn off warnings about
             IEEE 1666 deprecated features by placing this method call as the
```

first statement in your sc_main() function:

```
  sc_report_handler::set_actions("/IEEE_Std_1666/deprecated", SC_DO_NOTHING);
```

```
Program is about to exit.
#0  0x00000000004e40c0 in summit_sc::Runtime::atExitCallback ()
(vista)

(vista)
```

1) What are the differences between asynchronous SRAM and synchronous SRAM?

Synchronous relies on a clock source as it will read/write when the clock changes

Asynchronous doesn’t rely on the clock.