Multimedia SoC Design

Homework of Lab 2

March 25, 2016

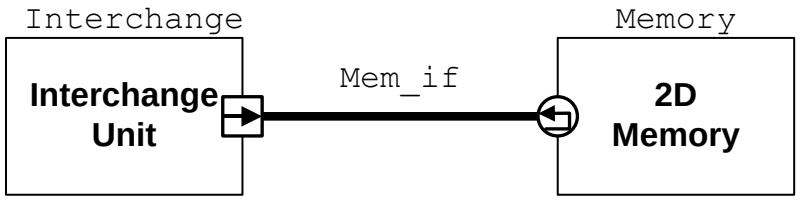
In this homework, you will be familiar with the relationship among interface, channel, and port. Our goal is to learn how to define a channel interface for SW and architecture design. Another goal is to learn how to refine the communication between building blocks of a system. A simple computation unit is adopted in this homework.

0.

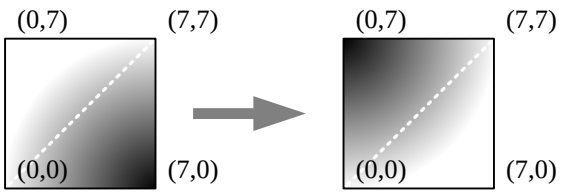
Practice customizing channels and data with the following examples:
Custom Primitive Channel: *Interrupt*
Custom Data Type: *Packet*
Custom Hierarchical Channel: *Heartbeat* and *Hier_Chan*

1.

Model the following design in untimed level:



The 2D memory module randomly generates a 8-by-8 block of data in its elaboration phase. Print the generated block of data to console. The rotation unit reads a 8-by-8 block {(0,0)(1,0)(2,0)…(7,7)} from transform memory to local memory, and then interchanges the points symmetric to line x=y in local memory as the following figure. After interchanging, the interchange unit writes the 8-by-8 block to 2D memory module. In the clean-up phase, print the block in 2D memory to console.



The interface is designed to supports two levels of communication model. One is for functional verification and the other is for architecture. These interface functions are defined as:

Functional model:

```
void direct_read(int** block);
void direct_write(int** block);
```

Architecture model:

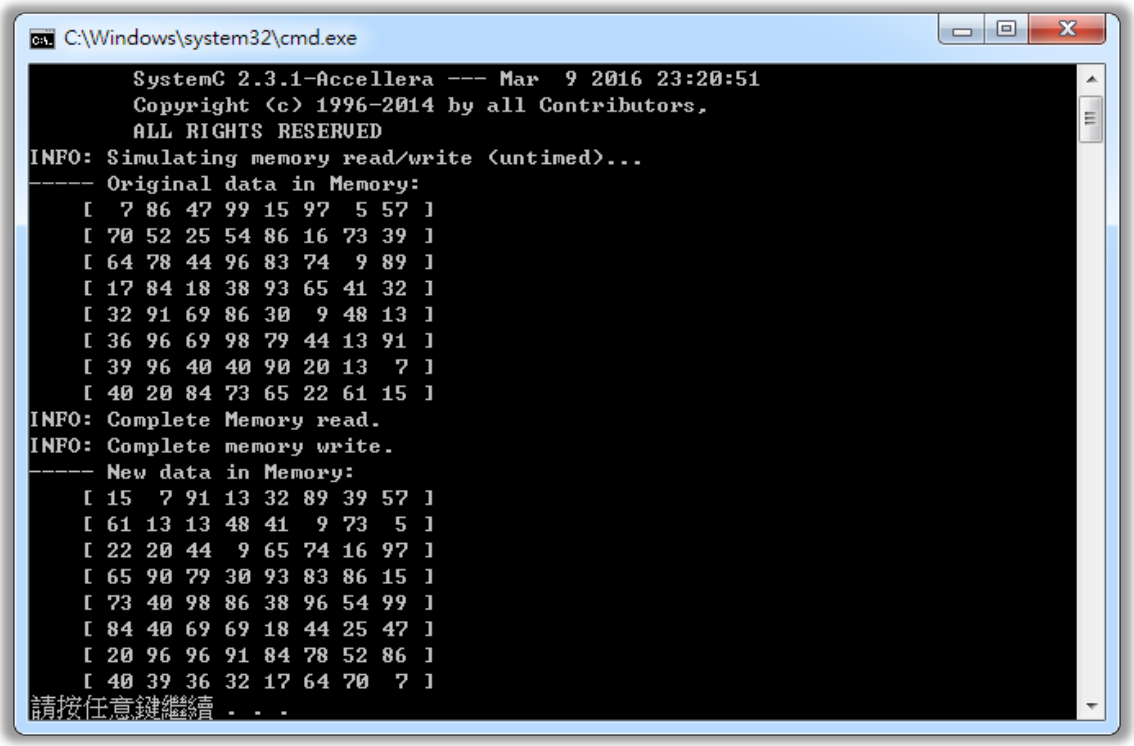
```
void word_read(unsigned x, unsigned y, int& d);
void word_write(unsigned x, unsigned y, int d);
```

1-1.

Implement the untimed functional model. The interchange unit uses direct_read() and direct_write() functions to pass the pointer to its local memory. The memory module can be a hierarchical channel which implements the interface. (a trivial channel implementation as the transactor example in lecture slide). Data in 2D memory are directly copied to local memory of interchange unit.

1-2.

Refine the model to untimed (or timed) architecture model. the interchange unit uses word_read() and word_write() functions to copy data from and to memory. You may use delay notification to model the timing in interchange unit or channel.



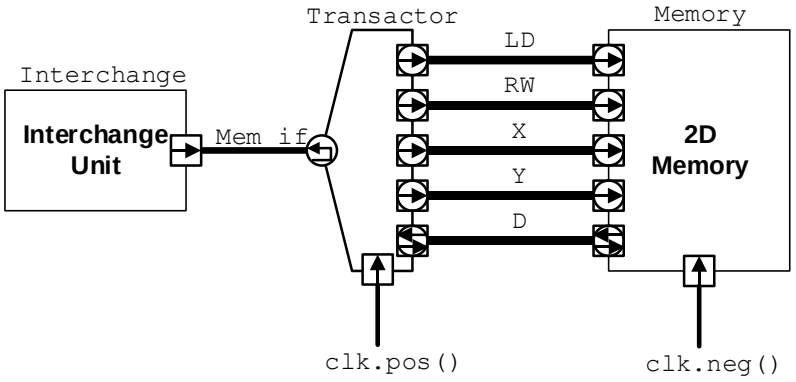
2.

In problem 2, you are asked to add timing into the model. The system is refine part-by-part. Signals are defined as:

Signal Name	**Channel Type**	**Port Type**	**Description**
LD	sc_signal<bool>	sc_in<bool> sc_out<bool>	true: activate operation false: idle
RW	sc_signal<bool>	sc_in<bool> sc_out<bool>	true: read memory false: write memory
X	sc_signal<unsigned>	sc_in<unsigned> sc_out<unsigned>	address along x direction
Y	sc_signal<unsigned>	sc_in<unsigned> sc_out<unsigned>	address along y direction
D	sc_signal_rv<32>	sc_inout_rv<32>	bi-directional data bus

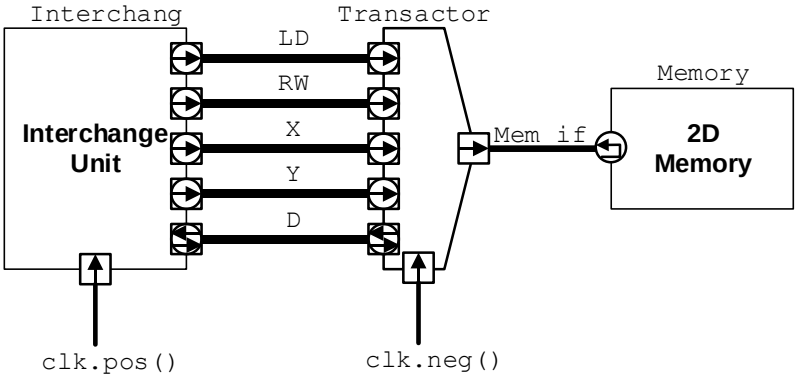
2-1.

Follow the transactor example, refine the memory model to pin-cycle accurate model.



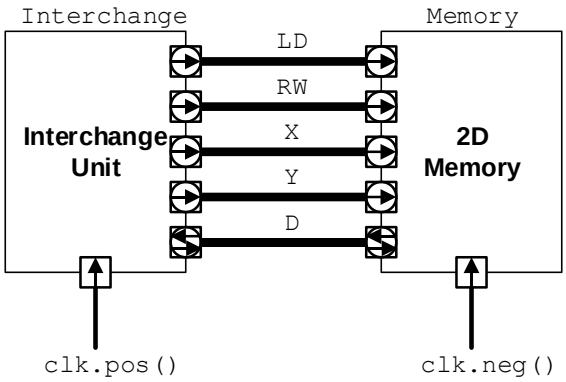
2-2.

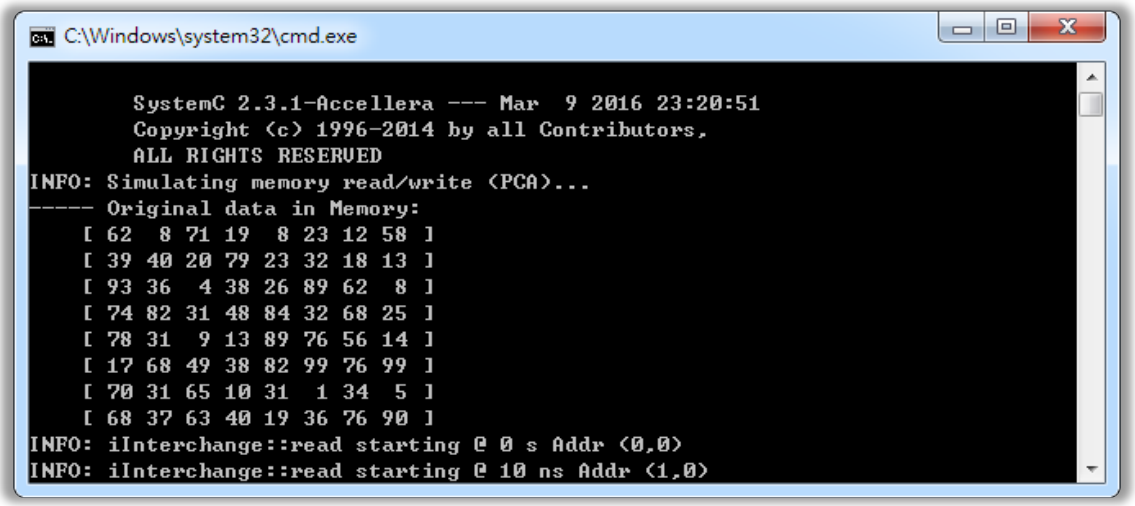
Follow the transactor example, refine the interchange model to PCA model.

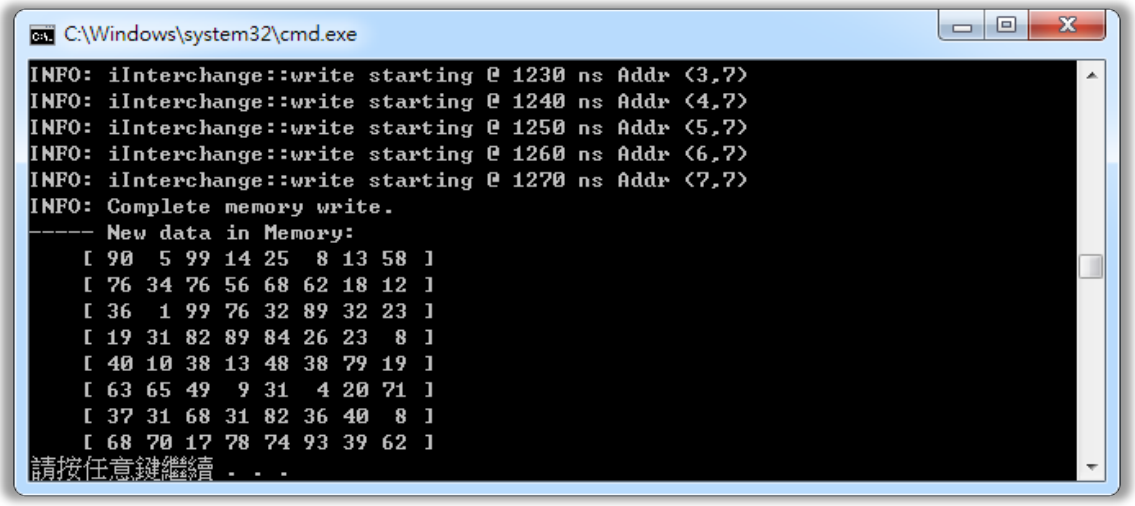


2-3.

Combine your refined interchange unit and 2D memory. Verify the result.







- All the files need to be compressed as a single ZIP or RAR file. Send this file to TA via email: ntumsoc@gmail.com

  Examples of email title:
  [MSOC] HW2_R04901001
  [MSOC] HW2_R04901001_Ver2

  Examples of filename:
  MSOC_HW2_R04901001.zip
  MSOC_HW2_R04901001_Ver2.zip

  Due date: 2016/04/08 Before Class (2 weeks)

繳交規定

source code (包括 lab 和作業)、report in PDF
請將這些檔案放入一個資料夾內如下排放

/HW2/	放report
/HW2/Lab2/	為Lab2的專案資料夾
/HW2/problem_1_1/	為1.1的專案資料夾
/HW2/problem_1_2/	為1.2的專案資料夾
/HW2/problem_2_1/	為2.1的專案資料夾
/HW2/problem_2_2/	為2.2的專案資料夾
/HW2/problem_2_3/	為2.3的專案資料夾

請注意，專案資料夾內必須包含 .vcxproj[.*] 檔和 .sln 檔，將SystemC必須預先設定以下五項專案參數先設定好

1. VC++目錄的 Include目錄路徑 & 程式庫目錄路徑
2. systemc.lib
3. _CRT_SECURE_NO_WARNINGS
4. Multi-threaded Debug (/Mtd)
5. /vmg

- 為避免整份檔案過大，請在專案完成後，刪除所有的Debug資料夾，以及ipch資料夾、.sdf檔、.suo檔…等等。