

University of Tehran

Electrical and Computer Engineering Department

ECE (8101) 342

Object Oriented Modeling of Electronic Circuits – Spring 1402-03

Computer Assignment 4

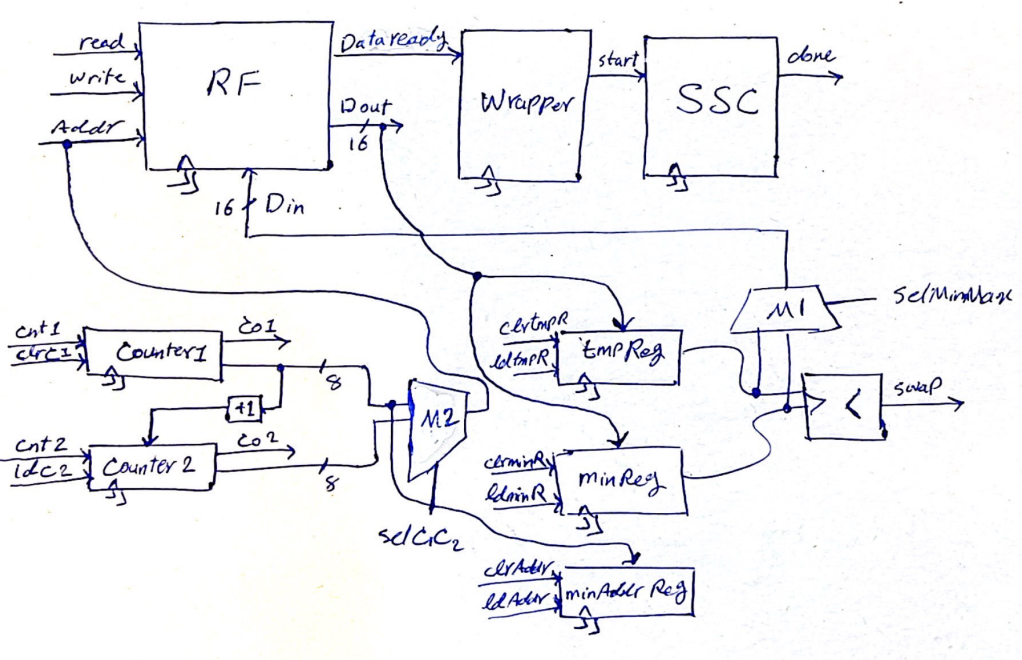
Zainalabedin Navabi

**Name:** Amirhesam Jafari rad

**SID:** 810100247

First of all lets have a look at the datapath ,in which three states, namely update0, update1, and update2, are assessed according to the CA deliveries.

**DataPath:**

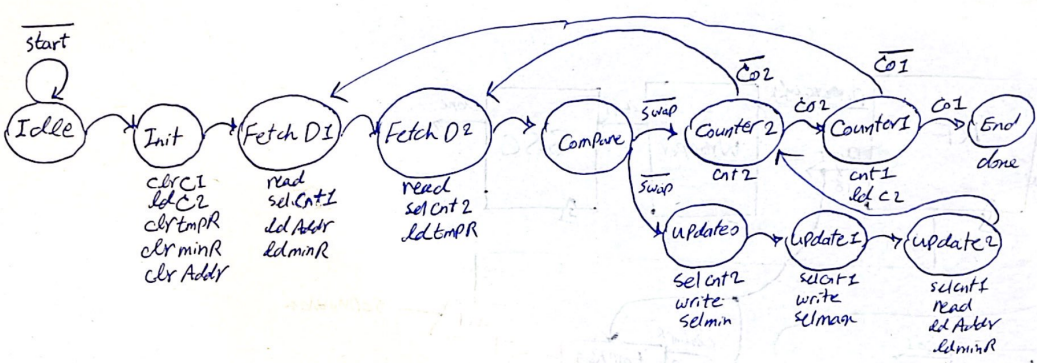


Note that I’ve only implemented the SSC Unit which contains register file, and not wrapper. As shown, there are three top modules named RF, Wrapper, and SSC corresponding to the register file, processing element and Selection Sorter Circuit respectively.

Then, all needed components for SSC is prepared, consisting of 2 counters, 2 muxes, 3 registers and a comparator. The +1 unit that connect Counter1 output to the Counter1 input isn’t implemented as an independent but is assumed to be done internally.

The controller of the SSC design is as below:

**Controller:**

****



As we can see, there are separated states for each operation. However, in reality, the Compare state was not that much necessary. In CPP implementation, the absence of this state leads to incorrect result due to incomplete inputs of the comparator unit.

**CPP Implementation:**

The library I’ve used for my wiring was buses.h which was prepared earlier. Then I create all my modules with their functionality implicitly in a components.h (Fig. 1) file where the components are defined as a C++ class. Then I instatiated each module in DataPath\_.h file which consist of instantiated modules and aa evl function, in which the whole datapath is evaluated. The content of evl method is declared in DataPath\_.cpp file (Fig. 2,3). After implementing datapath, I created a Controller.h file in which the states of shown controller is defined. It also has a .cpp file in which same as datapath, evaluations are handled (Fig. 4,5).

Finally I’ve write a top module file namely SSC.h and SSC.cpp which only instantiate the datapath and controller and evaluate them sequentially. The final file to test the design named testBench.cpp with the proper test cases. We can see that a memory named “mem.txt” is sorted in descending order in an “out.txt” file (Fig. 6,7).

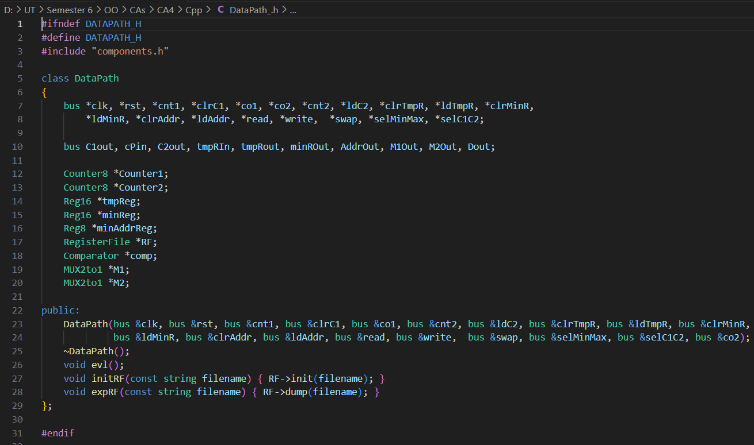
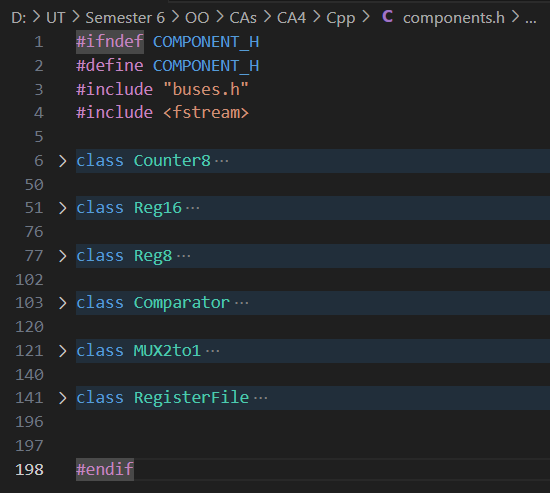


Fig. 1

Fig. 2

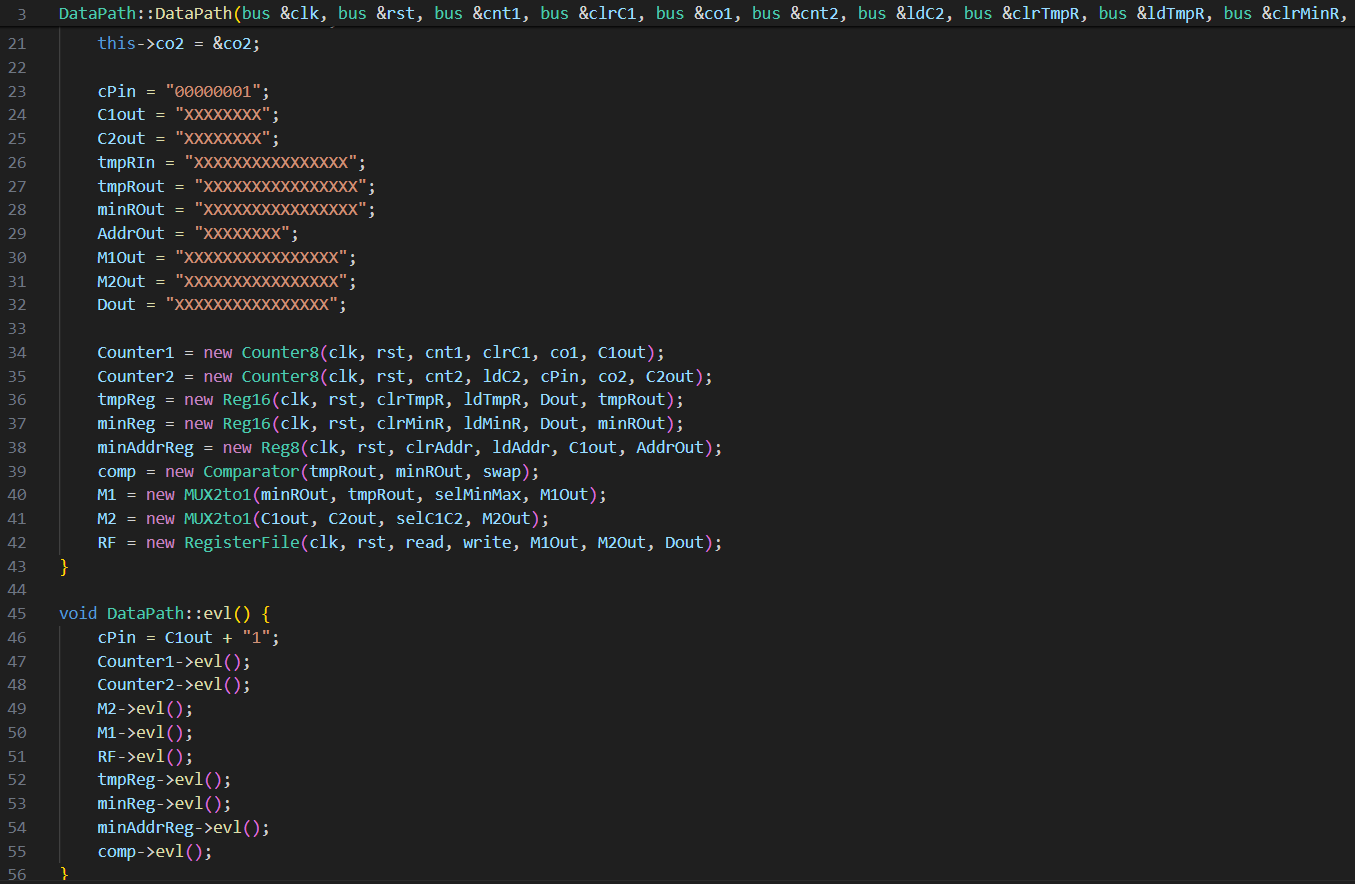
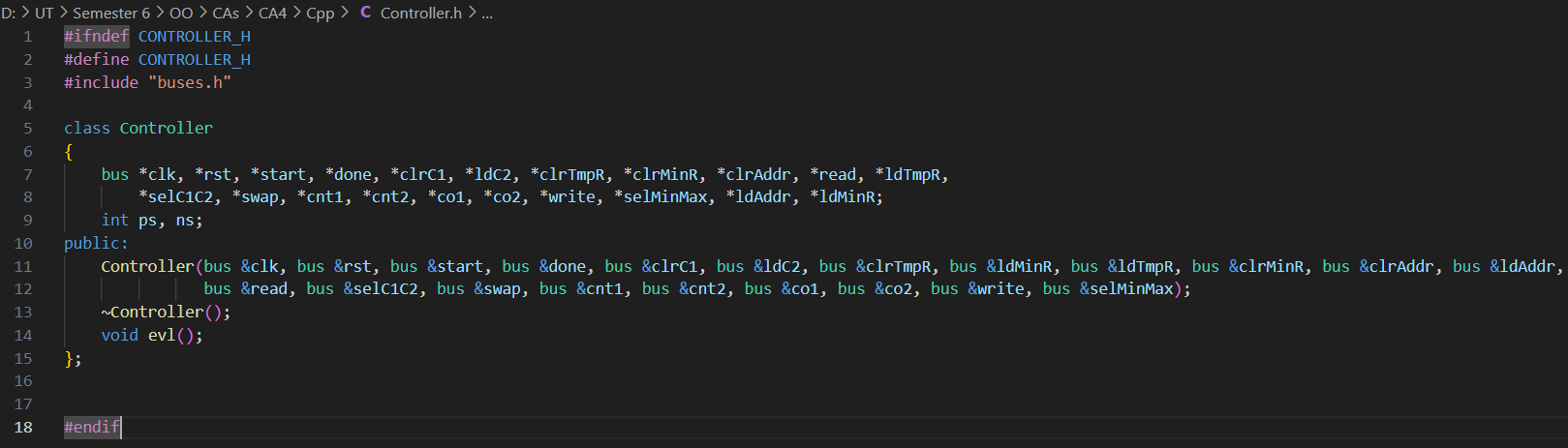
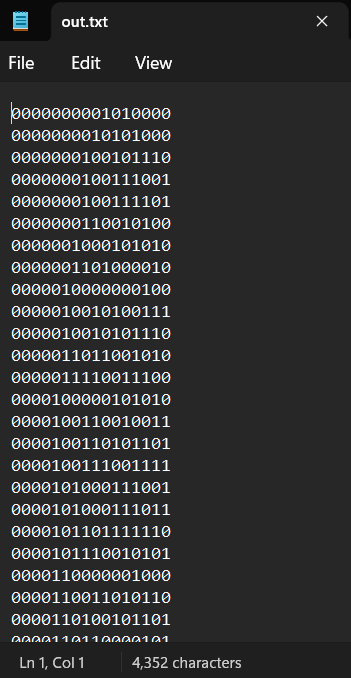


Fig. 4,5

Fig. 3





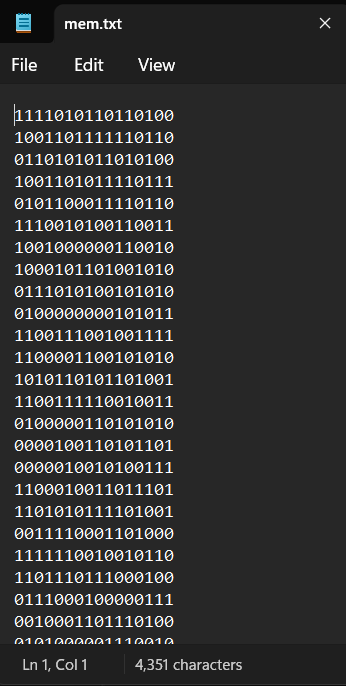
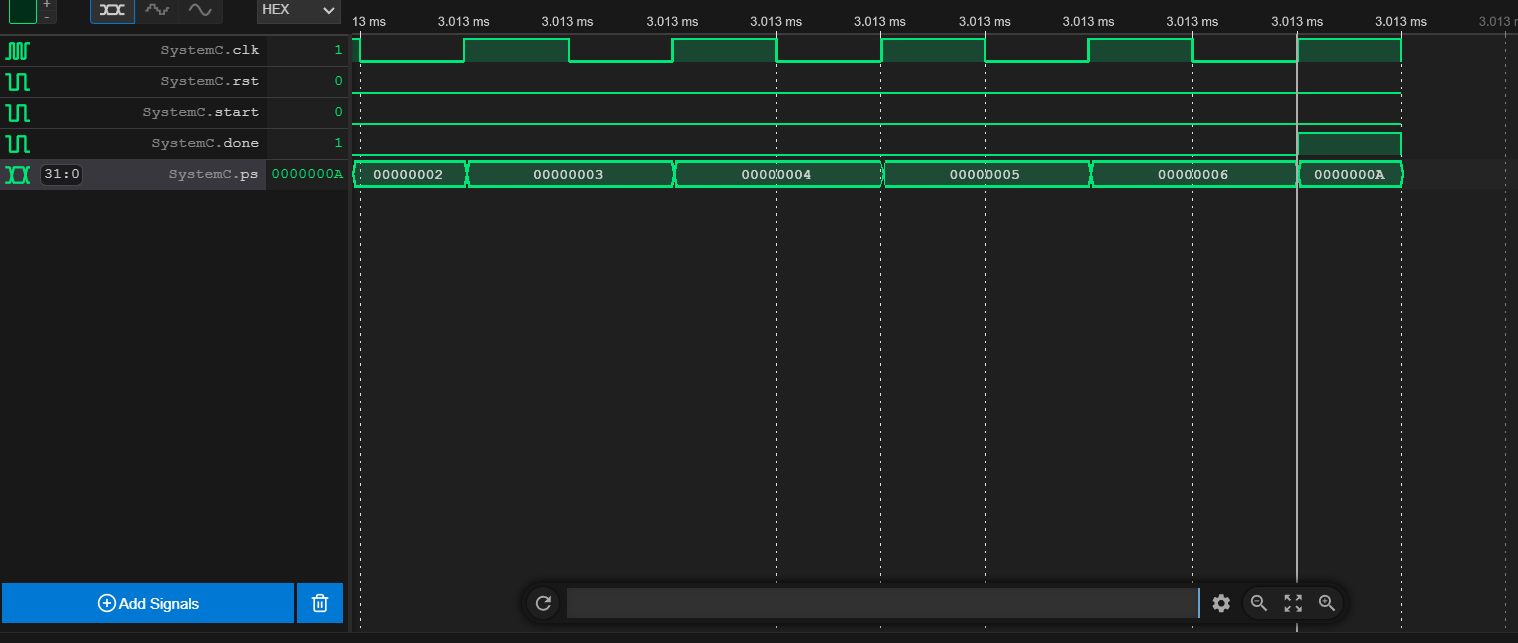


Fig. 6,7

**SystemC Impementation:**

The whole process is almost the same in systemC with some exception. In systemC there aren’t any evl function for modules so we don’t worry about the concurrency of our design components. I implemented each file with its corresponding file and structure in systemC even using the same file names. Just for components.h I also created a cpp file just for some beauty reason and make my code more readable.

The result of my systemC simulation using VCD files is shown below:



As is obvious the done signal is issued when the memory sorting is completely done.