I39/2208/2013 SPH 336 REPORT

INTRODUCTION

In this semester we learned system-level modeling using systemc. Systemc is a set of c++ classes and micros which provide simulation interface.

LANGUAGE FEATURES

systemc comprises of the following features:

Modules

Modules are the basic building blocks of a systemc design hierarchy. A systemc model usually consists of several modules which communicate via ports.

Ports

Ports allow communication from inside a module to the outside(usually to other modules) via channels.

Exports

Exports incorporate channels and allow communication from inside a module to the outside(usually to other modules).

Processes

Processes are the main computation elements. They are concurrent

Channels

Channels are the communication elements. They can be be either simple wires or complex communication mechanisms like FIFOs or bus channels.
The channels are:

- signal
- buffer
- FIFO
- mutex
- semaphore

Interfaces

Ports use interfaces to communicate with channels

Events

Events allow synchronization between processes and must be defined during initialization

I39/2208/2013 SPH 336 REPORT

SYSTEMC PROJECT 1

in this project I was supposed to modify the design of the 1 by 2 decoder to 2 by 4 decoder.

TRUTH TABLES

1 by 2 decoder

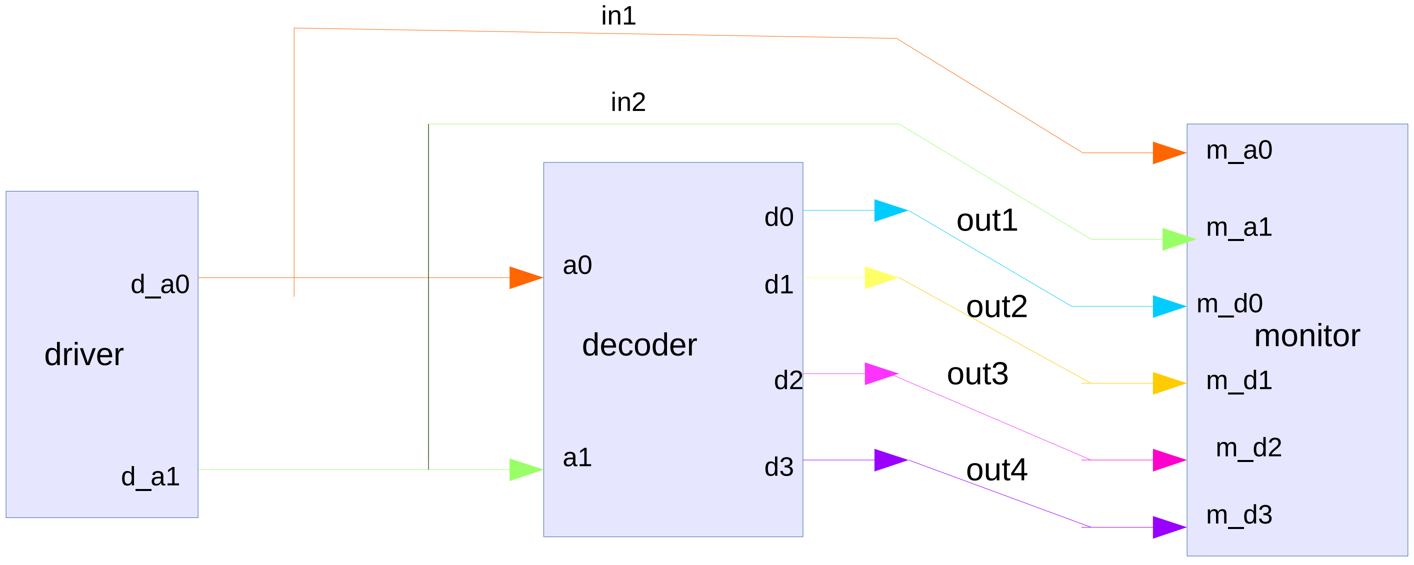
A	D_1	D_0
0	0	1
1	1	0

2 by 4 decoder

A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

MODULES

The module diagram is as shown below:



- driver: this module produced the input signals for the decoder.
- Decoder: implemented the working of the decoder

- monitor: displayed the variation of signals in the program.

FILES DESCRIPTION

driver2.h

This file defined the driver module
I declared two output ports of type bool named d_a0 and d_a1.
The function drive was to run in a while loop sending a combination of 1s and 0s to the decoder module.

decoder2.h

This file defined the decoder module
I declared two input ports of type bool as a0 and a1 and four output ports as d0,d1,d2,d3.
The implementation of the decode function is as follows:

- if input a0 was 0 and input a1 0,output d1,d2, and d3 were to be 0 and d0 to be 1.
- if input a0 was 1 and input a1 0, output d0,d2 and d3 were to be 0 and d1 to be 1.
- if input a0 was 0 and input a1 1, output d0,d1 and d3 were to be 0 and d2 to be 1.
- if input a0 was 1 and input a1 1, output d0,d1 and d2 were to be 0 and d3 to be 1.

monitor2.h

This file defined the monitor module
I declared six input ports of type bool as m_a0,m_a1,m_d0,m_d1,m_d2 and m_d3
the function monita was to output the values read from the ports.

decoder2.cc

this file implemented the connection between the three modules.
I defined six signals of type bool as in1,in2, out1, out2,out3 and out4.
I connected the corresponding ports as shown in the diagram above using the signals.
I also declared a trace file ft and a timing diagram file called timing_diagram to be used in the simulation.

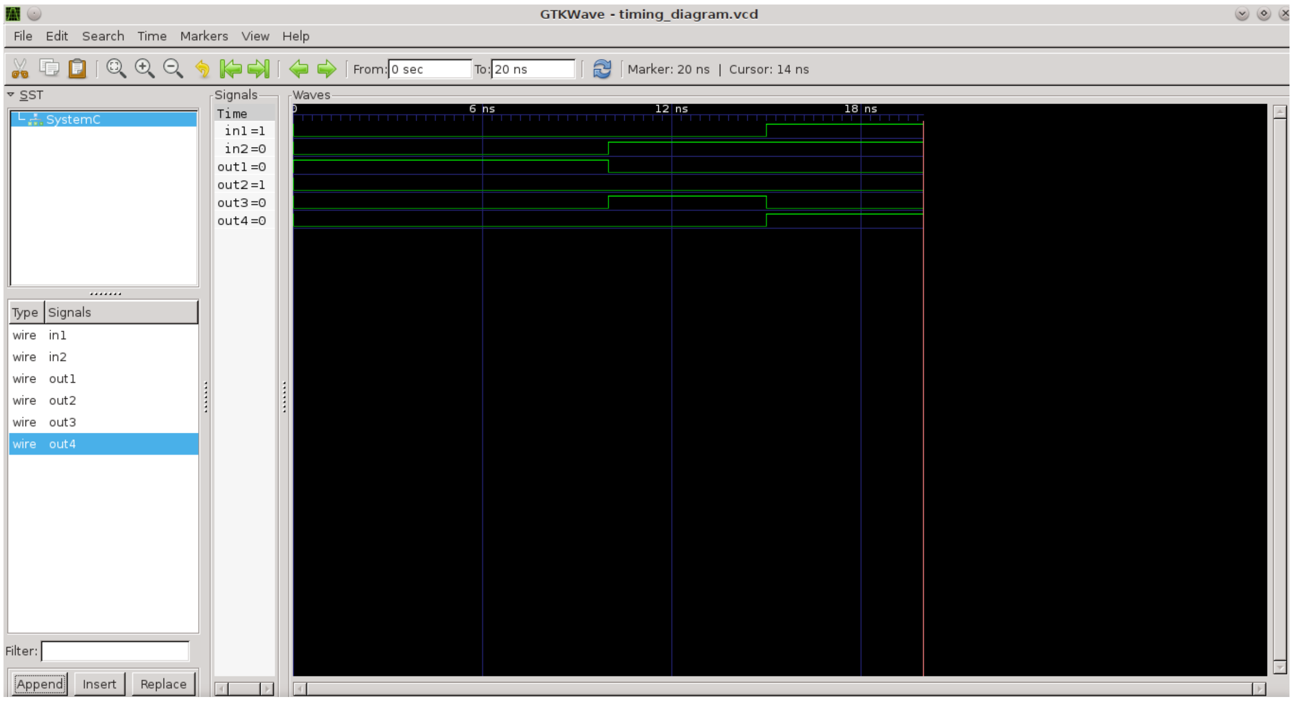
makefile

This file automated the process of compiling the different files and linking their object files into an executable called decoder.

I39/2208/2013 SPH 336 REPORT

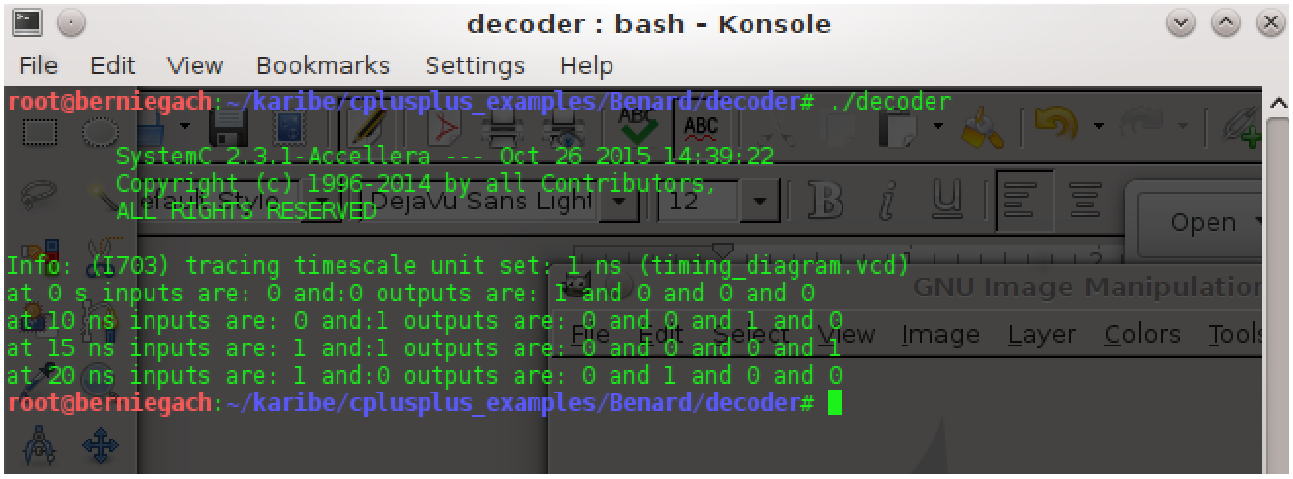
SIMULATION

For the simulation I used gtkwave and the output is as shown below.



RESULTS FROM THE SIMULATION

The results from the simulation are as shown below:



```
root@berniegach:~/karibe/cplusplus_examples/Benard/decoder# ./decoder

        SystemC 2.3.1-Accellera --- Oct 26 2015 14:39:22
        Copyright (c) 1996-2014 by all Contributors,
        ALL RIGHTS RESERVED

Info: (I703) tracing timescale unit set: 1 ns (timing_diagram.vcd)
at 0 s inputs are: 0 and:0 outputs are: 1 and 0 and 0 and 0
at 10 ns inputs are: 0 and:1 outputs are: 0 and 0 and 1 and 0
at 15 ns inputs are: 1 and:1 outputs are: 0 and 0 and 0 and 1
at 20 ns inputs are: 1 and:0 outputs are: 0 and 1 and 0 and 0
root@berniegach:~/karibe/cplusplus_examples/Benard/decoder#
```

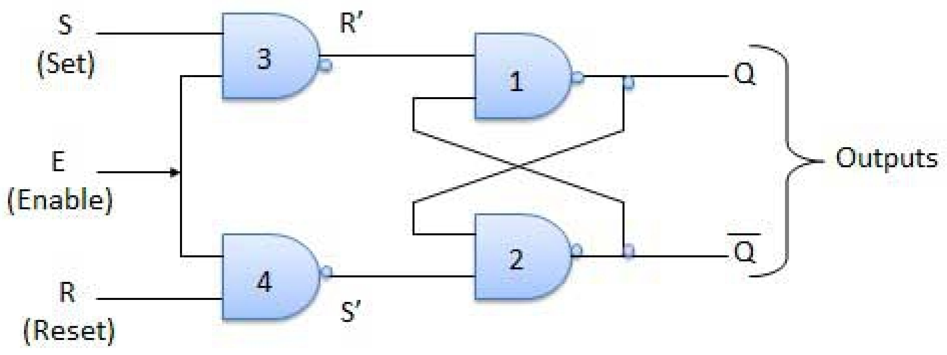
I39/2208/2013 SPH 336 REPORT

SYSTEMC PROJECT 2

In this project, I was supposed to design a sequential circuit of my choice using systemc and simulate it. I chose an S-R flip-flop as my circuit.

SEQUENTIAL CIRCUIT DIAGRAM

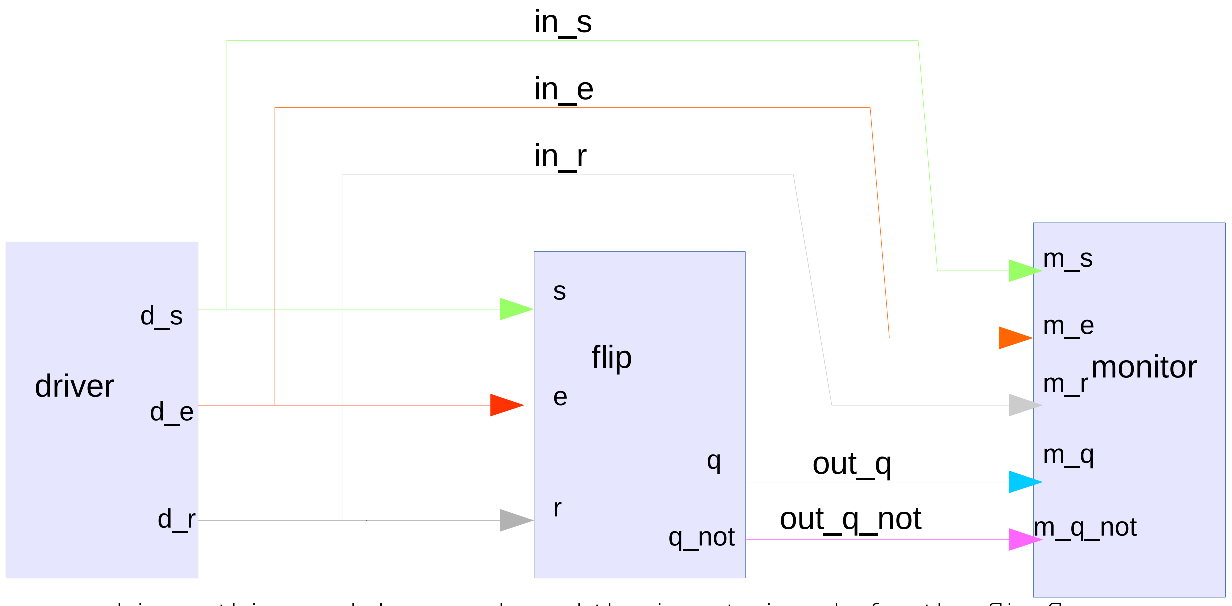
The circuit diagram for the sequential circuit is shown below:



MODULES

The module diagram is as shown below:

I used vi as my c++ editor and g++ as the compiler.
The module diagram is as shown below:



- driver: this modules produced the input signals for the flip flop
- flip: this module implemented the working of the flip flop
- monitor: this module displayed the variation of signals present in the program

FILES DECRIPTION

I39/2208/2013 SPH 336 REPORT

For this project, I used five different files to make the sequential circuit. These are

driver_seq.h

This file defined the driver module
I declared three output ports of type bool as d_s, d_r and d_e.
I named the only function as drive and it was supposed to run in a while loop sending a combination of 1 and 0 to the flip module with a wait of 5ns.

seque.h

This file defined the flip module.
I declared three output ports of type bool as s, e and r and two output ports of of type bool as q and q_not.
I declared the function as flip_do and its operation is as follows:

1. read the signals from s and e, carry out the boolean and on them, do the boolean not on the result and store in a temporary variable temp1.
2. read the signals from e and r, carry out the boolean and on them, do the boolean not on the result and store the result in a temporary variable temp2.
3. read the signal from q_not, do the boolean and with variable temp1, do the boolean not on the result and store the result in variable q.
4. read the signal from q, do the boolean and with variable temp2, do the boolean not on the result and store the result in variable q_not.

monitor_seq,h

This file defined the monitor module
I declared five input ports of type bool as m_e,m_s,m_r, m_q and m_q_not.
The function monita was to output the values read from this ports

seque.cc

This file implemented the connection between the three modules.
I declared five signals of type bool as in_s,in_e,in_r,out_q and out_q_not.
I connected the corresponding ports as shown in the diagrams using the signals.
I also declared a trace file ft and a timing diagram file called timing_diagram to be used in the simulation.

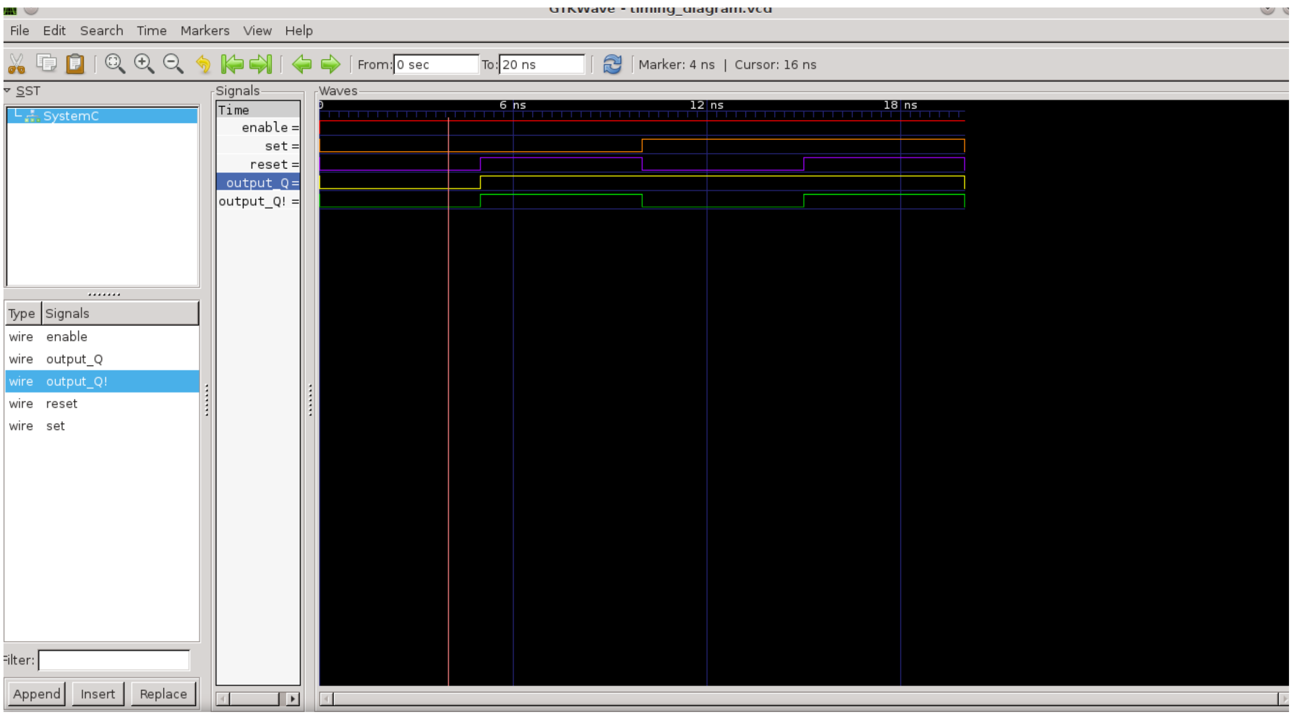
makefile

This file automated the process of compiling the different files and linking their object files into an executable called seque.

I39/2208/2013 SPH 336 REPORT

SIMULATION

For the simulation I used gtkwave and the output is as shown below.



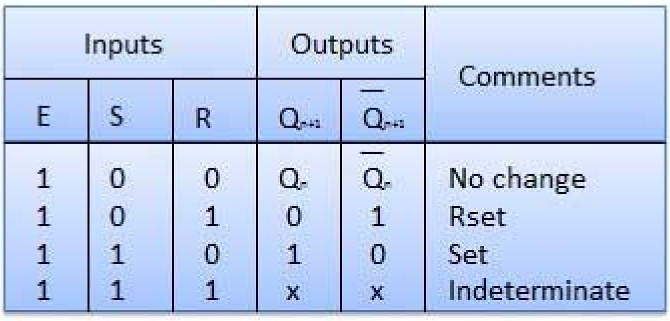
From: 0 sec | To: 20 ns | Marker: 4 ns | Cursor: 16 ns

Type	Signals
wire	enable
wire	output_Q
wire	output_Q!
wire	reset
wire	set

Signals: enable =, set =, reset =, output_Q =, output_Q! =

EXPECTED RESULTS

The theoretical expected results from the simulation is as shown in the table below.

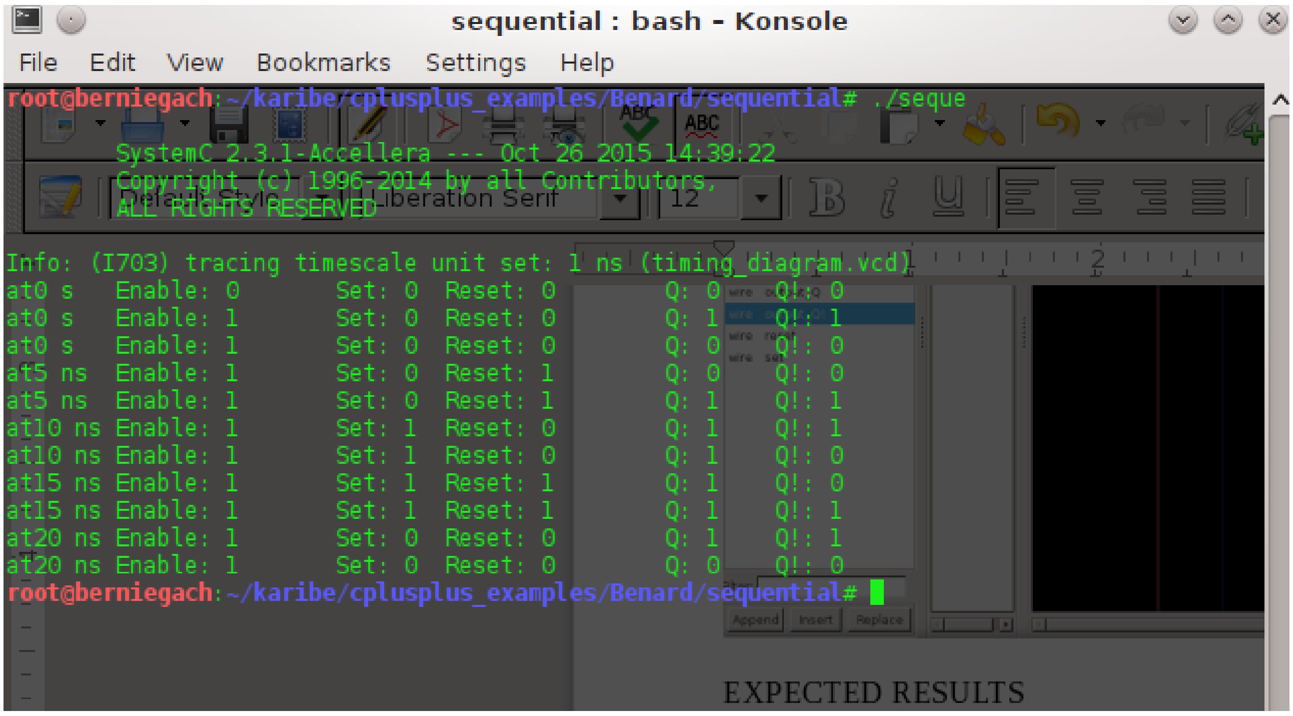


Inputs			Outputs		Comments
E	S	R	Q_{n+1}	$\overline{Q}_{n+1}$	
1	0	0	Q_n	$\overline{Q}_n$	No change
1	0	1	0	1	Rset
1	1	0	1	0	Set
1	1	1	x	x	Indeterminate

I39/2208/2013 SPH 336 REPORT

RESULTS FROM THE SIMULATION

The results from the simulation are as shown below


sequential : bash - Konsole
File Edit View Bookmarks Settings Help
EXPECTED RESULTS

```
root@berniegach:~/karibe/cplusplus_examples/Benard/sequential# ./seque

        SystemC 2.3.1-Accellera --- Oct 26 2015 14:39:22
        Copyright (c) 1996-2014 by all Contributors,
        ALL RIGHTS RESERVED

Info: (I703) tracing timescale unit set: 1 ns (timing_diagram.vcd)
at0 s   Enable: 0       Set: 0  Reset: 0        Q: 0    Q!: 0
at0 s   Enable: 1       Set: 0  Reset: 0        Q: 1    Q!: 1
at0 s   Enable: 1       Set: 0  Reset: 0        Q: 0    Q!: 0
at5 ns  Enable: 1       Set: 0  Reset: 1        Q: 0    Q!: 0
at5 ns  Enable: 1       Set: 0  Reset: 1        Q: 1    Q!: 1
at10 ns Enable: 1       Set: 1  Reset: 0        Q: 1    Q!: 1
at10 ns Enable: 1       Set: 1  Reset: 0        Q: 1    Q!: 0
at15 ns Enable: 1       Set: 1  Reset: 1        Q: 1    Q!: 0
at15 ns Enable: 1       Set: 1  Reset: 1        Q: 1    Q!: 1
at20 ns Enable: 1       Set: 0  Reset: 0        Q: 1    Q!: 1
at20 ns Enable: 1       Set: 0  Reset: 0        Q: 0    Q!: 0
root@berniegach:~/karibe/cplusplus_examples/Benard/sequential#
```

OBSERVATION

In the first project, the results from the simulation were the same as those in the truth table.
The project was successfully done.
The results from the simulation in project 2 were not fully the same as the expected results. There was some variation. This source of this error is the implementation of the flip_do method in seque.h file.

CONCLUSION

The objective of the unit was to teach me the skills of system modeling using systemc. Although in the last project the obtained results varied from the expected results I was still successful in learning how to use the systemc libraries to model various systems and do interpretation from the results obtained after simulation with gtkwave accordingly.