OWOKO PHILIP

H31/2207/2013

COMPUTING LAB 2

OBJECTIVE

To model and simulate a 2x4 binary decoder using SystemC.

THEORY/ INTRODUCTION

A decoder is a circuit that changes a code into a set of signals. It is called a decoder because it does the reverse of encoding.

A 2 to 4 decoder consists of two inputs and four outputs, truth table and symbols of which is shown in fig.1.1 below

The block diagram of 2 to 4 line decoder is shown in the fig.1.0. A and B are the two inputs where D through D are the four outputs. Truth table explains the operations of a decoder. It shows that each output is 1 for only a specific combination of inputs. Figure 1.2 shows the logic circuit diagram for the 2 to 4 decoder.

Block diagram

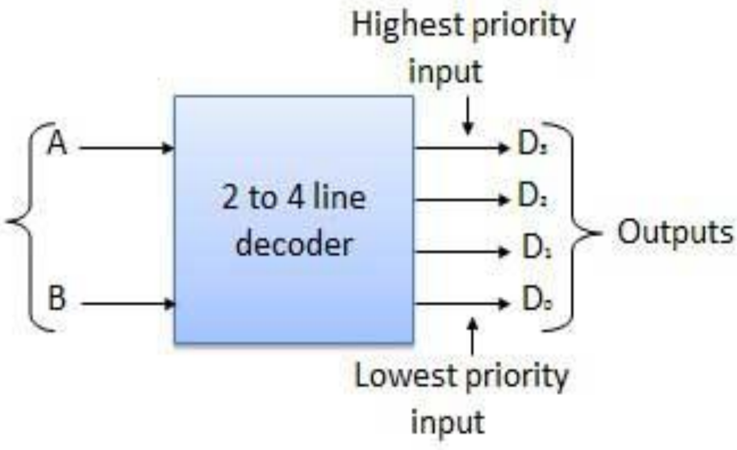
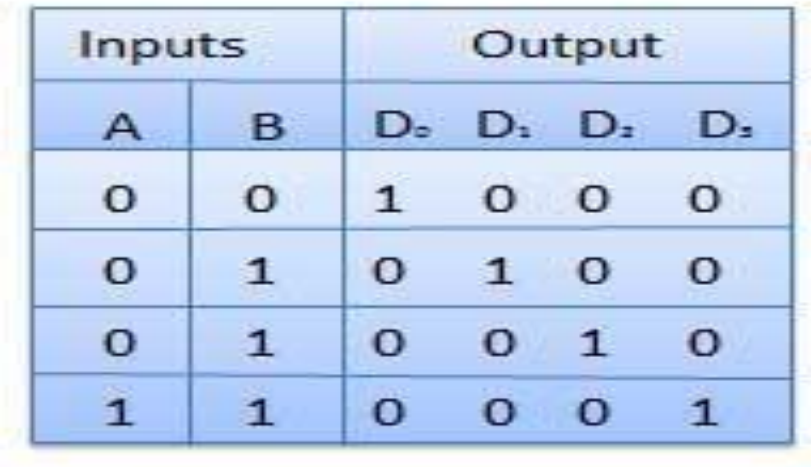


fig.1.0 Block diagram of a 2x4 Decoder

Truth Table



Inputs		Output			
A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
0	1	0	0	1	0
1	1	0	0	0	1

fig 1.1 Truth table for a 2x4 Decoder

Logic Circuit

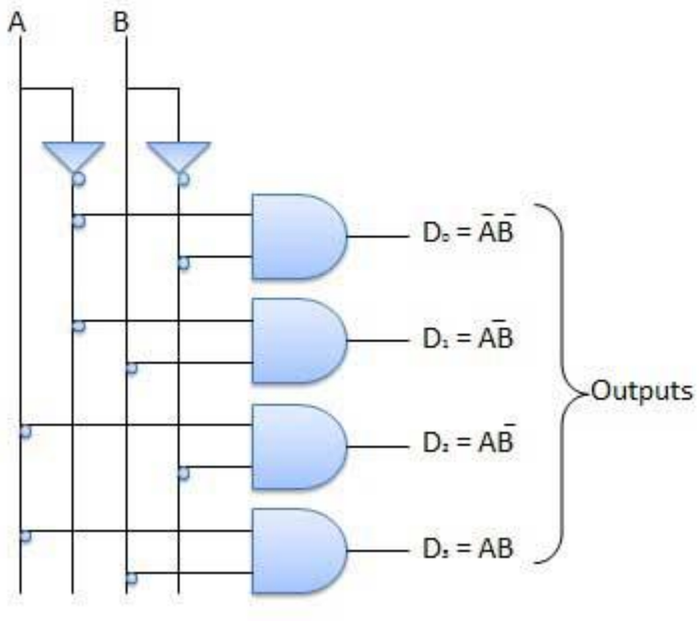


fig 1.2 Logic circuit for a 2x4 Decoder

PROCEDURE

A computer equipped with SystemC and Gtkwave was used to create three modules, decoder, monitor and driver, which were then linked to model the 2 to 4 binary decoder

The driver module was used to produce the inputs to the the decoder, after a defined time interval of 5 nanoseconds which looped. An exerp of the looping structure from the decoder is shown below.

```
while(1)
     {
                              //0 0 d_b being the most
          d_a=0; d_b=0;       significant
          wait(5, SC_NS);
          d_b=1;              //0 1
          wait(5, SC_NS);
          d_a=1; d_b=0;       //1 0
          wait(5, SC_NS);
          d_b=1;              //1 1
          wait(5, SC_NS);
     }
```

The monitor module just as its name, was tasked with the functionality of checking through the input side to verify the correctness of the inputs provided and also to check on the result if it’s the one intended.

Then there was the decoder module which played a very critical role in the whole simulation. Just as from fig1.0 to 1.1 above, the decoder had two inputs and then produced the respective four outputs.

It incorporates the services of both the two modules, driver and monitor.

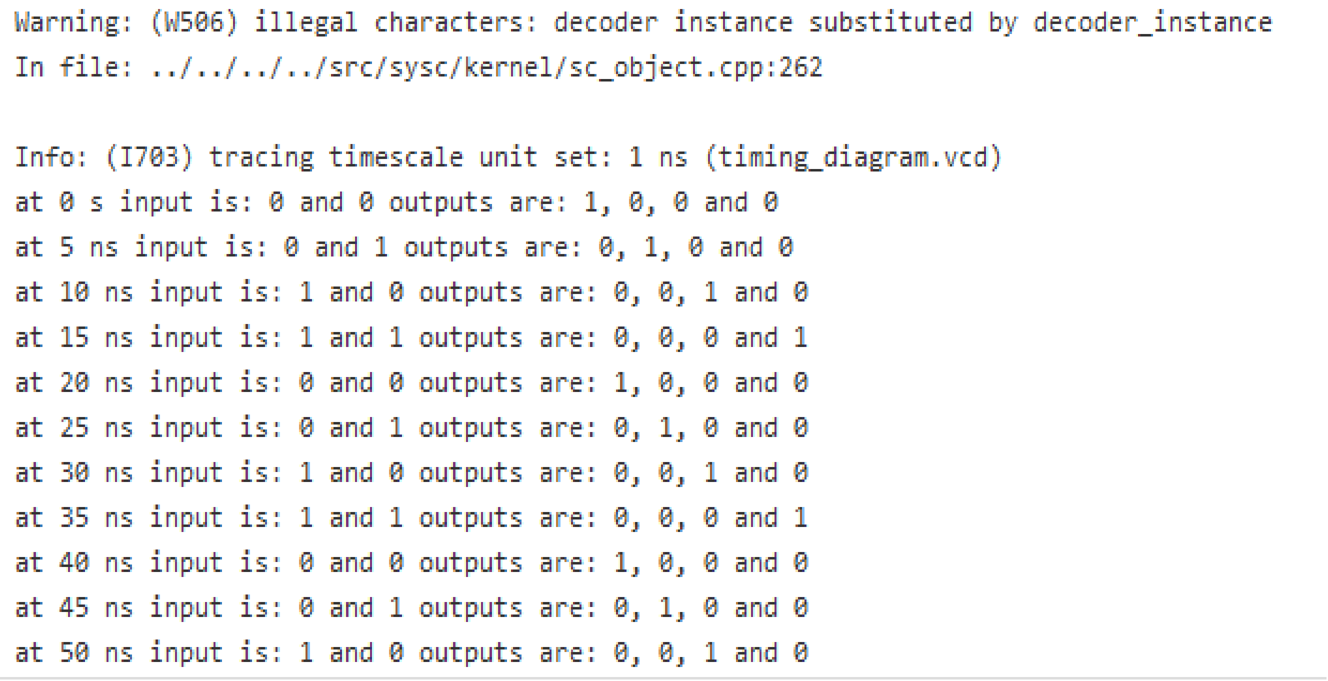
A trace file was also created that was opened in the installed Gtkwave to view the actual simulation of the decoder. When I tested the files in other machines, it produced the expected result as shown in fig 1.3 but on running the on mine, I got a rather weird one in fig 1.4

The decoder module at 2by4decoder.cc;

```
#include"2by4decoder.h"
#include"2by4driver.h"
#include"2by4monitor.h"
#include<systemc>

int sc_main(int argc, char *argv[])
{
    //signals for interconnections
    sc_signal<bool> in1, in2, out1, out2, out3, out4;
    //module instances
    decoder2to4 dec("decoder instance");
    driver2 dr("driver");
    monitor2to4 h("monitor");
    //interconnections between modules
    dr.d_a(in1);dr.d_b(in2);
    dec.a(in1);dec.b(in2);
    h.monitor_a(in1);h.monitor_b(in2);

    dec.c(out1);
    h.monitor_c(out1);
    dec.d(out2);
    h.monitor_d(out2);
    dec.e(out3);
    h.monitor_e(out3);
    dec.f(out4);
    h.monitor_f(out4);

    //create a trace file with nanosecond resolution
    sc_trace_file *tf;
    tf = sc_create_vcd_trace_file("timing_diagram");
    tf ->set_time_unit(1, SC_NS);
    //trace the signals interconnecting modules
    sc_trace(tf, in1, "first_binary_input");
    sc_trace(tf, in2, "second_binary_input");
    sc_trace(tf, out1, "input_is_zero");
    sc_trace(tf, out2, "input_is_one");
    sc_trace(tf, out3, "input_is_two");
    sc_trace(tf, out4, "input_is_three");

    //run a simulation for 50 systemc nanoseconds
    if (!sc_pending_activity())
        sc_start(55, SC_NS);
    //close the trace file
    sc_close_vcd_trace_file(tf);
    return 0;
}
```

RESULTS

The overall modelling produced the following result and was then stored in results.txt



```
Warning: (W506) illegal characters: decoder instance substituted by decoder_instance
In file: ../../../../src/sysc/kernel/sc_object.cpp:262

Info: (I703) tracing timescale unit set: 1 ns (timing_diagram.vcd)
at 0 s input is: 0 and 0 outputs are: 1, 0, 0 and 0
at 5 ns input is: 0 and 1 outputs are: 0, 1, 0 and 0
at 10 ns input is: 1 and 0 outputs are: 0, 0, 1 and 0
at 15 ns input is: 1 and 1 outputs are: 0, 0, 0 and 1
at 20 ns input is: 0 and 0 outputs are: 1, 0, 0 and 0
at 25 ns input is: 0 and 1 outputs are: 0, 1, 0 and 0
at 30 ns input is: 1 and 0 outputs are: 0, 0, 1 and 0
at 35 ns input is: 1 and 1 outputs are: 0, 0, 0 and 1
at 40 ns input is: 0 and 0 outputs are: 1, 0, 0 and 0
at 45 ns input is: 0 and 1 outputs are: 0, 1, 0 and 0
at 50 ns input is: 1 and 0 outputs are: 0, 0, 1 and 0
```

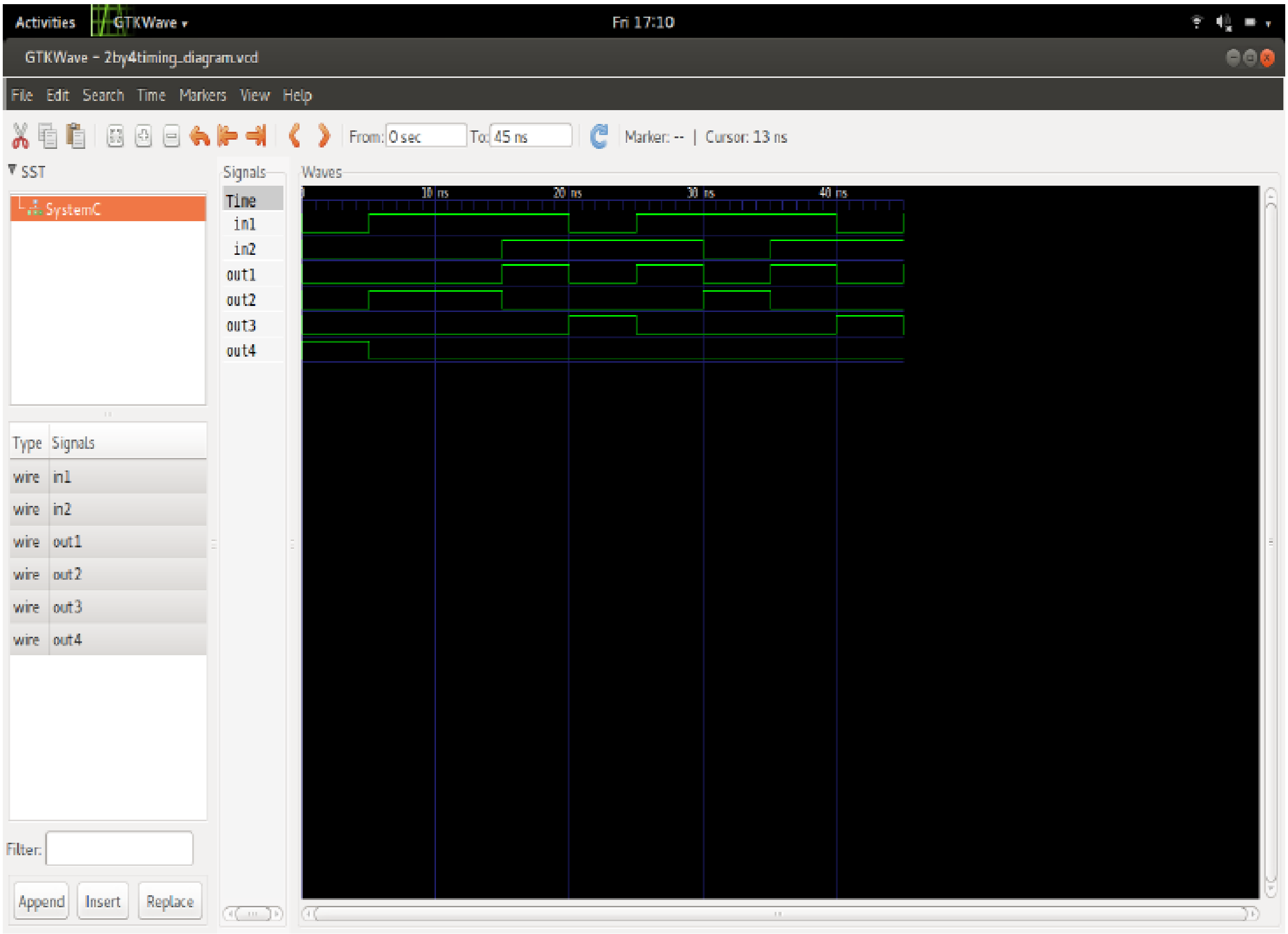


Fig 1.3 The expected Gtkwave waveform (Achieved when run on other machine)

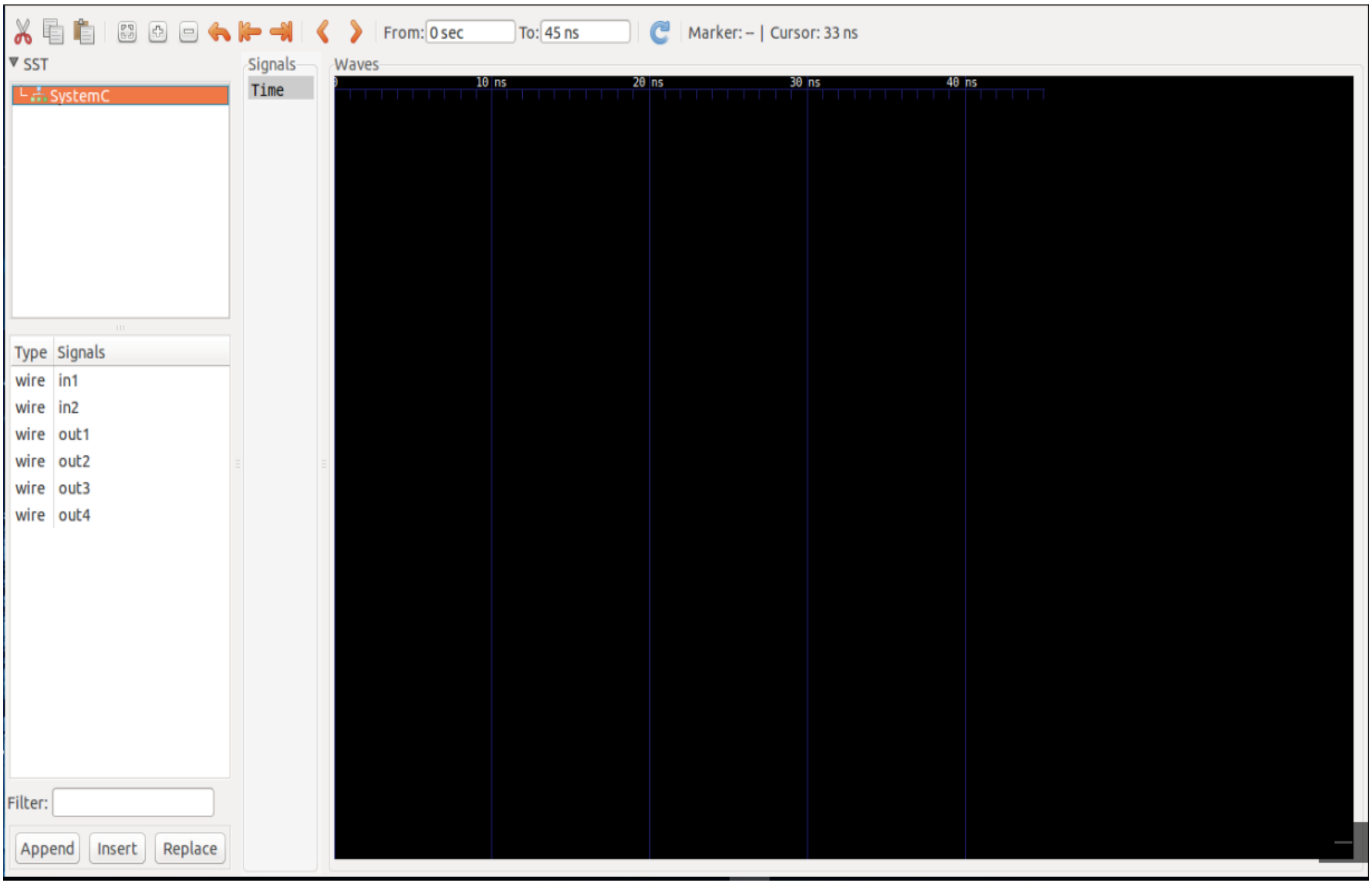


Fig 1.4 The undesired Gtkwave waveform (Achieved when run on my linux machine)

DISCUSSION

The 2 to 4 binary decoder implementation was a little bit of challenging due to the misbehaviour of my linux machine but I still managed to do it with the help of another linux machine with the same necessary required tools.
The VCD trace file in figure 1.3 was the expected one and I managed to achieve it on a different machine but on my machine, I was receiving the one in figure1.3. I would like to know why this happened.
The 2 to 4 decoder gave an output of logic level 1 at the port corresponding to the binary number at the inputs thus confirming that the modules were correctly modelled.

CONCLUSION

The overall objective was achieved since I was able to model and simulate a 2x4 binary decoder using SystemC.

REFERENCES

1.https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0CDcQFjAGahUKEwjCzryx_vDIAhXIqxoKHUrECZc&url=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FDecoder&usg=AFQjCNEUKlgVKVWqHkZT-GHhH60NIn49Lg&sig2=9ryY2BnfWiMNOENKOV8L2g
4.https://www.google.com/search?q=2x4+decoder+truth+table&revid=1992735984&sa=X&ved=0CGAQ1QIoAGoVChMIws68sf7wyAIVyKsaCh1KxAmX&biw=1366&bih=667