

南京大学《数字电路基础实验》（实验一）实验报告

实验日期: 2022 年 11 月 26 日至 2022 年 12 月 6 日

实验项目: 南京大学《数字电路基础实验》实验一

实验操作系统: Ubuntu Linux 22.04.1 LTS (GUI) amd64

实验操作人: 烟雨松 (一生一芯学号: ysyx_22041384)

实验内容: 使用 Verilog HDL 编写 2 选 1 多路选择器; 使用 Verilog HDL 编写 4 选 1 多路选择器并深入研究 4 选 1 多路选择器。

实验第一部分: 使用 Verilog HDL 编写 2 选 1 多路选择器

实验分为三个模块: 1) 使用数据流建模实现模块 2) 使用结构建模实现模块 3) 使用行为建模实现模块

实验步骤: 对每一个模块分别编写 Verilog HDL 文件, 同时使用 C++ 编写 simulator 文件, 通过 Verilator 进行编译

实验一: 使用数据流建模实现模块

实验环境:

- 1) 操作系统 Ubuntu Linux 22.04.1 LTS (Jammy)
- 2) 工作目录 "/home/ce/OneStudentOneChip/DigitalCircuitsExperiments/DCE01/mux_2to1_Datastream"
- 3) 编辑软件: Visual Studio Code
- 4) 软件环境: Verilator (4.210), g++ (11.2.0-1ubuntu1)

实验目的: 通过数据流建模 2 选 1 多路选择器, 了解多路选择器如何使用数据流建模并学习 Verilog HDL 语言的使用方式, 练习使用 Verilator (熟悉 Verilator 的相关调取方法, 比如说在运行的时候需要在命令行除了说明文件名称以外还需要在命令行输入那些指令)

实验基础: 2 选 1 多路选择器的逻辑结构—2 选 1 多路选择器有三个输入端口 (a,b,s) 和一个输出端口 (y)。输入端口中的 a 和 b 是两个要被选择的信号来源, s 的真值决定了选择器会选择哪一路信号: 如果 s 为 0, 那么选择 a; 如果 s 为 1, 那么选择 b。根据以上定义, 画出卡诺图, 标记真值为 1 的情况, 得出逻辑式: $y = (\sim s \& a) | (s \& b)$ 。根据以上逻辑式, 编写 Verilog HDL 硬件描述文件。

实验过程 1: 通过得出的逻辑式, 写出 Verilog HDL 硬件描述文件: (文件名称: mux_2to1_Datastream.v)

```
module mux_2to1_Datastream (a, b, s, y);
    input a, b, s;
    output y;
    assign y = (~s & a) | (s & b);
endmodule
```

实验过程 2: 给该模块设计一个 C++ 语言的激励文件, 内部包含了 8 种可能的端口情况 (a、b、s 各两种情况, 0 或 1)。(文件名称: mux_2to1_Datastream_sim.cpp)

```
//#include <verilated.h>
//#include <verilated_vcd_c.h>
//#include <obj_dir/Vmux_2to1_Datastream.h>

#include "/usr/local/share/verilator/include/verilated.h"
#include "/usr/local/share/verilator/include/verilated_vcd_c.h"
#include
"/home/ce/OneStudentOneChip/DigitalCircuitsExperiments/DCE01/mux_2to1_Datastream/obj_dir/Vmux_2to1_Datastream.h"

#include <iostream>

VerilatedContext* contextp = NULL;
VerilatedVcdC* tfp = NULL;

static Vmux_2to1_Datastream* top;

void step_and_dump_wave()
{
    top->eval();
    contextp->timeInc(1);
}
```

```

    tftp -> dump(contexttp -> time());
}

void sim_init() // Initilize the simulator
{
    contexttp = new VerilatedContext;
    tftp = new VerilatedVcdC;
    top = new Vmux_2to1_Datastream;
    contexttp -> traceEverOn(true);
    top -> trace (tftp, 0);
    tftp -> open("dump.vcd");
}

void sim_exit() // Terminate the simulator
{
    step_and_dump_wave();
    tftp -> close();
}

int main()
{
    // Print the Experiment Informations
    printf("Verilator Generated File simulator");
    printf("\n");
    printf("This is a Mutiplexer modeled by Datastream");
    printf("\n");

    // Simulator Main Parts
    sim_init();

    // Condition sab
    top -> s = 0;
    top -> a = 0;
    top -> b = 0;
    step_and_dump_wave();

    // Condition saB
    top -> s = 0;
    top -> a = 0;
    top -> b = 1;
    step_and_dump_wave();

    // Condition sAb
    top -> s = 0;
    top -> a = 1;
    top -> b = 0;
    step_and_dump_wave();

    // Condition sAB
    top -> s = 0;
    top -> a = 1;
    top -> b = 1;
    step_and_dump_wave();

    // Condition Sab
    top -> s = 1;
    top -> a = 0;
    top -> b = 0;
    step_and_dump_wave();

    // Condition SaB
    top -> s = 1;
    top -> a = 0;
    top -> b = 1;

```

```

step_and_dump_wave();

// Condition SAb
top -> s = 1;
top -> a = 1;
top -> b = 0;
step_and_dump_wave();

// Condition SAB
top -> s = 1;
top -> a = 1;
top -> b = 1;
step_and_dump_wave();

sim_exit();
}

```

实验过程 3: 使用 Verilator 进行编译文件 (需要同时导入写好的硬件描述文件和激励文件), 使用命令 “verilator mux_2to1_Datastream.v mux_2to1_Datastream_sim.cpp --cc --exe --build --trace --debug” 进行文件生成。在文件生成后, 使用指令 “.Vmux_2to1_Datastream” 运行 Verilator 编译好的文件, 目录 “{working directory}/obj_dir” 结构如下:

```

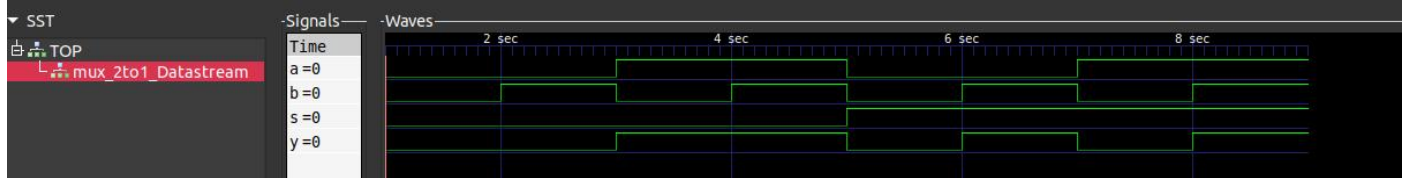
drwxrwxr-x 2 ce ce 12288 11 月 27 10:58 ./
drwxrwxr-x 3 ce ce 4096 11 月 27 10:57 ../
-rw-rw-r-- 1 ce ce 498 11 月 27 10:58 dump.vcd
-rw-rw-r-- 1 ce ce 553 11 月 27 10:57 mux_2to1_Datastream_sim.d
-rw-rw-r-- 1 ce ce 6304 11 月 27 10:57 mux_2to1_Datastream_sim.o
-rw-rw-r-- 1 ce ce 446 11 月 27 10:56 verilated.d
-rw-rw-r-- 1 ce ce 145408 11 月 27 10:56 verilated.o
-rw-rw-r-- 1 ce ce 472 11 月 27 10:56 verilated_vcd_c.d
-rw-rw-r-- 1 ce ce 50872 11 月 27 10:56 verilated_vcd_c.o
-rwxrwxr-x 1 ce ce 142216 11 月 27 10:57 Vmux_2to1_Datastream*
-rw-rw-r-- 1 ce ce 148 11 月 27 10:57 Vmux_2to1_Datastream_001_linkcells.dot
-rw-rw-r-- 1 ce ce 1865 11 月 27 10:57 Vmux_2to1_Datastream_002_cells.tree
-rw-rw-r-- 1 ce ce 2060 11 月 27 10:57 Vmux_2to1_Datastream_007_link.tree
-rw-rw-r-- 1 ce ce 85 11 月 27 10:57 Vmux_2to1_Datastream_008_linkInc.tree
-rw-rw-r-- 1 ce ce 85 11 月 27 10:57 Vmux_2to1_Datastream_010_paramlink.tree
-rw-rw-r-- 1 ce ce 2856 11 月 27 10:57 Vmux_2to1_Datastream_012_width.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_014_const.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_015_assertpre.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_016_assert.tree
-rw-rw-r-- 1 ce ce 4580 11 月 27 10:57 Vmux_2to1_Datastream_018_const.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_023_begin.tree
-rw-rw-r-- 1 ce ce 2717 11 月 27 10:57 Vmux_2to1_Datastream__024root.cpp
-rw-rw-r-- 1 ce ce 1004 11 月 27 10:57 Vmux_2to1_Datastream__024root.h
-rw-rw-r-- 1 ce ce 2411 11 月 27 10:57 Vmux_2to1_Datastream__024root_Slow.cpp
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_024_tristate.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_025_unknown.tree
-rw-rw-r-- 1 ce ce 5115 11 月 27 10:57 Vmux_2to1_Datastream_026_inline.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_028_const.tree
-rw-rw-r-- 1 ce ce 4622 11 月 27 10:57 Vmux_2to1_Datastream_029_deadDtypes.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_030_inst.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_031_const.tree
-rw-rw-r-- 1 ce ce 7567 11 月 27 10:57 Vmux_2to1_Datastream_032_scope.tree
-rw-rw-r-- 1 ce ce 7423 11 月 27 10:57 Vmux_2to1_Datastream_033_linkdot.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_034_class.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_035_const.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_036_deadDtypesScoped.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_037_case.tree
-rw-rw-r-- 1 ce ce 194 11 月 27 10:57 Vmux_2to1_Datastream_038_task_call.dot
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_039_task.tree

```

-rw-rw-r-- 1 ce ce 7522 11 月 27 10:57 Vmux_2to1_Datastream_041_unroll.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_042_slice.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_043_const.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_044_life.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_045_table.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_046_const.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_047_deadDtypesScoped.tree
-rw-rw-r-- 1 ce ce 7784 11 月 27 10:57 Vmux_2to1_Datastream_048_active.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_049_split.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_050_splitas.tree
-rw-rw-r-- 1 ce ce 11119 11 月 27 10:57 Vmux_2to1_Datastream_051_tracedecl.tree
-rw-rw-r-- 1 ce ce 2875 11 月 27 10:57 Vmux_2to1_Datastream_052_gate_simp.dot
-rw-rw-r-- 1 ce ce 2650 11 月 27 10:57 Vmux_2to1_Datastream_053_gate_opt.dot
-rw-rw-r-- 1 ce ce 8206 11 月 27 10:57 Vmux_2to1_Datastream_054_gate.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_055_const.tree
-rw-rw-r-- 1 ce ce 6878 11 月 27 10:57 Vmux_2to1_Datastream_056_deadAllScoped.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_057_reorder.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_058_delayed.tree
-rw-rw-r-- 1 ce ce 8776 11 月 27 10:57 Vmux_2to1_Datastream_059_activetop.tree
-rw-rw-r-- 1 ce ce 1827 11 月 27 10:57 Vmux_2to1_Datastream_060_orderg_pre.dot
-rw-rw-r-- 1 ce ce 148 11 月 27 10:57 Vmux_2to1_Datastream_061_acyc_simp.dot
-rw-rw-r-- 1 ce ce 1828 11 月 27 10:57 Vmux_2to1_Datastream_062_orderg_acyc.dot
-rw-rw-r-- 1 ce ce 1882 11 月 27 10:57 Vmux_2to1_Datastream_063_orderg_order.dot
-rw-rw-r-- 1 ce ce 1883 11 月 27 10:57 Vmux_2to1_Datastream_064_orderg_domain.dot
-rw-rw-r-- 1 ce ce 405 11 月 27 10:57 Vmux_2to1_Datastream_065_order_edges.txt
-rw-rw-r-- 1 ce ce 1881 11 月 27 10:57 Vmux_2to1_Datastream_066_orderg_done.dot
-rw-rw-r-- 1 ce ce 9376 11 月 27 10:57 Vmux_2to1_Datastream_067_order.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_068_genclock.tree
-rw-rw-r-- 1 ce ce 9012 11 月 27 10:57 Vmux_2to1_Datastream_069_clock.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_070_const.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_071_life.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_072_life_post.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_073_const.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_074_deadAllScoped.tree
-rw-rw-r-- 1 ce ce 9377 11 月 27 10:57 Vmux_2to1_Datastream_075_changed.tree
-rw-rw-r-- 1 ce ce 15837 11 月 27 10:57 Vmux_2to1_Datastream_076_trace.tree
-rw-rw-r-- 1 ce ce 13136 11 月 27 10:57 Vmux_2to1_Datastream_078_descope.tree
-rw-rw-r-- 1 ce ce 11752 11 月 27 10:57 Vmux_2to1_Datastream_079_combine.tree
-rw-rw-r-- 1 ce ce 11611 11 月 27 10:57 Vmux_2to1_Datastream_080_const.tree
-rw-rw-r-- 1 ce ce 11295 11 月 27 10:57 Vmux_2to1_Datastream_081_deadAll.tree
-rw-rw-r-- 1 ce ce 11858 11 月 27 10:57 Vmux_2to1_Datastream_082_clean.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_083_premittree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_084_expand.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_085_const_cpp.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_086_subst.tree
-rw-rw-r-- 1 ce ce 87 11 月 27 10:57 Vmux_2to1_Datastream_087_const_cpp.tree
-rw-rw-r-- 1 ce ce 11753 11 月 27 10:57 Vmux_2to1_Datastream_088_deadAll.tree
-rw-rw-r-- 1 ce ce 12118 11 月 27 10:57 Vmux_2to1_Datastream_092_cast.tree
-rw-rw-r-- 1 ce ce 14995 11 月 27 10:57 Vmux_2to1_Datastream_093_cuse.tree
-rw-rw-r-- 1 ce ce 15888 11 月 27 10:57 Vmux_2to1_Datastream_990_final.tree
-rw-rw-r-- 1 ce ce 16746 11 月 27 10:57 Vmux_2to1_Datastream__ALL.a
-rw-rw-r-- 1 ce ce 360 11 月 27 10:57 Vmux_2to1_Datastream__ALL.cpp
-rw-rw-r-- 1 ce ce 692 11 月 27 10:57 Vmux_2to1_Datastream__ALL.d
-rw-rw-r-- 1 ce ce 13928 11 月 27 10:57 Vmux_2to1_Datastream__ALL.o
-rw-rw-r-- 1 ce ce 1775 11 月 27 10:57 Vmux_2to1_Datastream_classes.mk
-rw-rw-r-- 1 ce ce 5853 11 月 27 10:57 Vmux_2to1_Datastream.cpp
-rw-rw-r-- 1 ce ce 2880 11 月 27 10:57 Vmux_2to1_Datastream.h
-rw-rw-r-- 1 ce ce 1964 11 月 27 10:57 Vmux_2to1_Datastream.mk
-rw-rw-r-- 1 ce ce 216 11 月 27 10:57 Vmux_2to1_Datastream_mux_2to1_Datastream.vpp

```
-rw-rw-r-- 1 ce ce 21719 11 月 27 10:57 Vmux_2to1_Datastream__stats.txt
-rw-rw-r-- 1 ce ce 865 11 月 27 10:57 Vmux_2to1_Datastream__Syms.cpp
-rw-rw-r-- 1 ce ce 1223 11 月 27 10:57 Vmux_2to1_Datastream__Syms.h
-rw-rw-r-- 1 ce ce 1765 11 月 27 10:57 Vmux_2to1_Datastream__Trace.cpp
-rw-rw-r-- 1 ce ce 3368 11 月 27 10:57 Vmux_2to1_Datastream__Trace__Slow.cpp
-rw-rw-r-- 1 ce ce 4169 11 月 27 10:57 Vmux_2to1_Datastream__ver.d
-rw-rw-r-- 1 ce ce 11031 11 月 27 10:57 Vmux_2to1_Datastream__verFiles.dat
-rw-rw-r-- 1 ce ce 10972 11 月 27 10:57 Vmux_2to1_Datastream.xml
```

实验过程 4: 根据已经编译、生成好的文件, 使用指令 “gtkwave dump.vcd” 在模拟示波器中观察图像, 结果如下图。



实验二：通过结构建模实现模块

实验环境:

- 1) 操作系统 Ubuntu Linux 22.04.1 LTS (Jammy)
- 2) 工作目录 “/home/yanyusong/OneStudentOneChip/DigitalCircuitsExperiments/DCE01/mux_2to1_Structure”
- 3) 编辑软件 Visual Studio Code
- 4) 软件环境 Verilator (5.003), g++ (11.2.0-1ubuntu1)

实验目的: 通过结构建模 2 选 1 多路选择器, 了解多路选择器如何使用结构建模并学习 Verilog HDL 语言的使用方式, 练习使用 Verilator (熟悉 Verilator 的相关调用方法, 比如说在运行的时候需要在命令行除了说明文件名称以外还需要在命令行输入那些指令)

实验基础: 在实验一中已经得出了 2 选 1 多路选择器的逻辑表达式: $y = (\sim s \& a) | (s \& b)$, 发现其中有以下几个关键逻辑部件, 在表达式 $\sim s \& a$ 中, 我们可以发现一个非门和一个与门参与了这部分计算, 在表达式 $s \& b$ 中, 我们可以发现一个与门参与了计算, 整体来看, 表达式 $(\sim s \& a) | (s \& b)$ 还需要一个或门。由此, 我们分析出需要先写出或门、与门、非门的实现后通过逻辑结构的调用来实现这个 2 选 1 多路选择器。

实验过程 1: 通过数据流建模, 设计出一个与门的 Verilog HDL 描述文件。 (文件名: my_and.v)

```
module my_and (a, b, c);
    input a, b;
    output c;
    assign c = a & b;
endmodule
```

实验过程 2: 通过数据流建模, 设计出一个非门的 Verilog HDL 描述文件。 (文件名: my_not.v)

```
module my_not (a, b);
    input a;
    output b;
    assign b = ~a;
endmodule
```

实验过程 3: 通过数据流建模, 设计出一个或门的 Verilog HDL 描述文件。 (文件名: my_or.v)

```
module my_or (a, b, c);
    input a, b;
    output c;
    assign c = a | b;
endmodule
```

实验过程 4: 基于过程 1-3 设计出的描述文件, 写出通过结构建模的 2 选 1 多路选择器的 Verilog HDL 硬件描述文件: (文件名: mux_2to1_Structure.v)

```
module mux_2to1_Structure(a, b, s, y);
    input a, b, s;
    output y;
    wire l, r, s_n;
    my_not i1(.a(s), .b(s_n));
    my_and i2 (.a(s_n), .b(a), .c(l));
```

```

    my_and i3 (.a(s), .b(b), .c(r));
    my_or i4 (.a(l), .b(r), .c(y));
endmodule

```

实验过程 5: 给该模块设计一个 C++语言的激励文件, 内部包含了 8 中可能的端口情况 (a、b、s 各两种情况, 0 或 1)。(文件名称: mux_2to1_Structure_sim.cpp)

```

#include <verilated.h>
#include <verilated_vcd_c.h>
#include <obj_dir/Vmux_2to1_Datastream.h>

#include "/home/ce/OneStudentOneChip/include/verilated.h"
#include "/home/ce/OneStudentOneChip/include/verilated_vcd_c.h"
#include
"/home/ce/OneStudentOneChip/DigitalCircuitsExperiments/DCE01/mux_2to1_Structure/obj_dir/Vmux_2to1_Structur
e.h"

#include <iostream>

VerilatedContext* contextp = NULL;
VerilatedVcdC* tfp = NULL;

static Vmux_2to1_Structure* top;

void step_and_dump_wave()
{
    top -> eval();
    contextp -> timeInc(1);
    tfp -> dump(contextp -> time());
}

void sim_init() // Initilize the simulator
{
    contextp = new VerilatedContext;
    tfp = new VerilatedVcdC;
    top = new Vmux_2to1_Structure;
    contextp -> traceEverOn(true);
    top -> trace (tfp, 0);
    tfp -> open("dump.vcd");
}

void sim_exit() // Terminate the simulator
{
    step_and_dump_wave();
    tfp -> close();
}

int main()
{
    // Print the Experiment Informations
    printf("Verilator Generated File simulator");
    printf("\n");
    printf("This is a Mutiplexer modeled by Structure");
    printf("\n");

    // Simulator Main Parts
    sim_init();

    // Condition sab
    top -> s = 0;
    top -> a = 0;
    top -> b = 0;
    step_and_dump_wave();

    // Condition saB

```

```

top -> s = 0;
top -> a = 0;
top -> b = 1;
step_and_dump_wave();

// Condition sAb
top -> s = 0;
top -> a = 1;
top -> b = 0;
step_and_dump_wave();

// Condition sAB
top -> s = 0;
top -> a = 1;
top -> b = 1;
step_and_dump_wave();

// Condition Sab
top -> s = 1;
top -> a = 0;
top -> b = 0;
step_and_dump_wave();

// Condition SaB
top -> s = 1;
top -> a = 0;
top -> b = 1;
step_and_dump_wave();

// Condition SAb
top -> s = 1;
top -> a = 1;
top -> b = 0;
step_and_dump_wave();

// Condition SAB
top -> s = 1;
top -> a = 1;
top -> b = 1;
step_and_dump_wave();

sim_exit();
}

```

实验过程 6: 使用 Verilator 进行编译文件 (需要同时导入写好的硬件描述文件和激励文件), 使用命令“verilator mux_2to1_Structure.v myx_2to1_Structure_sim.cpp --cc --exe --build --trace --debug”进行文件生成。在文件生成后, 使用指令“./Vmux_2to1_Structure”运行 Verilator 编译好的文件, 目录“{working directory}/obj_dir”结构如下:

```

-rwxrwxr-x 1 yanyusong yanyusong 142160 12 月 5 23:22 Vmux_2to1_Structure
-rw-rw-r-- 1 yanyusong yanyusong 5803 12 月 5 23:22 Vmux_2to1_Structure.cpp
-rw-rw-r-- 1 yanyusong yanyusong 2868 12 月 5 23:22 Vmux_2to1_Structure.h
-rw-rw-r-- 1 yanyusong yanyusong 1955 12 月 5 23:22 Vmux_2to1_Structure.mk
-rw-rw-r-- 1 yanyusong yanyusong 15307 12 月 5 23:22 Vmux_2to1_Structure.xml
-rw-rw-r-- 1 yanyusong yanyusong 703 12 月 5 23:22 Vmux_2to1_Structure_001_linkcells.dot
-rw-rw-r-- 1 yanyusong yanyusong 6183 12 月 5 23:22 Vmux_2to1_Structure_002_cells.tree
-rw-rw-r-- 1 yanyusong yanyusong 7682 12 月 5 23:22 Vmux_2to1_Structure_007_link.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Structure_008_linkInc.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Structure_010_paramlink.tree
-rw-rw-r-- 1 yanyusong yanyusong 9499 12 月 5 23:22 Vmux_2to1_Structure_012_width.tree
-rw-rw-r-- 1 yanyusong yanyusong 9258 12 月 5 23:22 Vmux_2to1_Structure_014_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Structure_015_assertpre.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Structure_016_assert.tree
-rw-rw-r-- 1 yanyusong yanyusong 11126 12 月 5 23:22 Vmux_2to1_Structure_018_const.tree

```

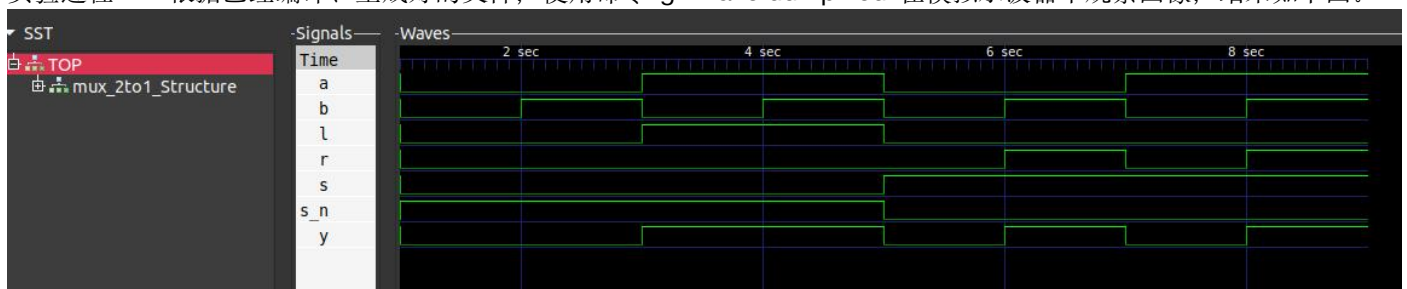
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Structure_023_begin.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Structure_024_tristate.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Structure_025_unknown.tree
-rw-rw-r-- 1 yanyusong yanyusong 15157 12 月 5 23:22 Vmux_2to1_Structure_026_inline.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Structure_028_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 13593 12 月 5 23:22 Vmux_2to1_Structure_029_deadDTypes.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Structure_030_inst.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Structure_031_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 22936 12 月 5 23:22 Vmux_2to1_Structure_032_scope.tree
-rw-rw-r-- 1 yanyusong yanyusong 22292 12 月 5 23:22 Vmux_2to1_Structure_033_linkdot.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_034_class.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_035_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_036_deadDTypesScoped.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_037_case.tree
-rw-rw-r-- 1 yanyusong yanyusong 193 12 月 5 23:22 Vmux_2to1_Structure_038_task_call.dot
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_039_task.tree
-rw-rw-r-- 1 yanyusong yanyusong 22664 12 月 5 23:22 Vmux_2to1_Structure_041_unroll.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_042_slice.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_043_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_044_life.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_045_table.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_046_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_047_deadDTypesScoped.tree
-rw-rw-r-- 1 yanyusong yanyusong 23047 12 月 5 23:22 Vmux_2to1_Structure_048_active.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_049_split.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_050_splitas.tree
-rw-rw-r-- 1 yanyusong yanyusong 32646 12 月 5 23:22 Vmux_2to1_Structure_051_tracedecl.tree
-rw-rw-r-- 1 yanyusong yanyusong 8564 12 月 5 23:22 Vmux_2to1_Structure_052_gate_simp.dot
-rw-rw-r-- 1 yanyusong yanyusong 8197 12 月 5 23:22 Vmux_2to1_Structure_053_gate_opt.dot
-rw-rw-r-- 1 yanyusong yanyusong 20748 12 月 5 23:22 Vmux_2to1_Structure_054_gate.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_055_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 14517 12 月 5 23:22 Vmux_2to1_Structure_056_deadAllScoped.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_057_reorder.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_058_delayed.tree
-rw-rw-r-- 1 yanyusong yanyusong 16423 12 月 5 23:22 Vmux_2to1_Structure_059_activetop.tree
-rw-rw-r-- 1 yanyusong yanyusong 1826 12 月 5 23:22 Vmux_2to1_Structure_060_orderg_pre.dot
-rw-rw-r-- 1 yanyusong yanyusong 147 12 月 5 23:22 Vmux_2to1_Structure_061_acyc_simp.dot
-rw-rw-r-- 1 yanyusong yanyusong 1827 12 月 5 23:22 Vmux_2to1_Structure_062_orderg_acyc.dot
-rw-rw-r-- 1 yanyusong yanyusong 1881 12 月 5 23:22 Vmux_2to1_Structure_063_orderg_order.dot
-rw-rw-r-- 1 yanyusong yanyusong 1882 12 月 5 23:22 Vmux_2to1_Structure_064_orderg_domain.dot
-rw-rw-r-- 1 yanyusong yanyusong 405 12 月 5 23:22 Vmux_2to1_Structure_065_order_edges.txt
-rw-rw-r-- 1 yanyusong yanyusong 1880 12 月 5 23:22 Vmux_2to1_Structure_066_orderg_done.dot
-rw-rw-r-- 1 yanyusong yanyusong 17033 12 月 5 23:22 Vmux_2to1_Structure_067_order.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_068_genclk.tree
-rw-rw-r-- 1 yanyusong yanyusong 16661 12 月 5 23:22 Vmux_2to1_Structure_069_clock.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_070_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_071_life.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_072_life_post.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_073_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_074_deadAllScoped.tree
-rw-rw-r-- 1 yanyusong yanyusong 17032 12 月 5 23:22 Vmux_2to1_Structure_075_changed.tree
-rw-rw-r-- 1 yanyusong yanyusong 21711 12 月 5 23:22 Vmux_2to1_Structure_076_trace.tree
-rw-rw-r-- 1 yanyusong yanyusong 18100 12 月 5 23:22 Vmux_2to1_Structure_078_descope.tree
-rw-rw-r-- 1 yanyusong yanyusong 16711 12 月 5 23:22 Vmux_2to1_Structure_079_combine.tree
-rw-rw-r-- 1 yanyusong yanyusong 16544 12 月 5 23:22 Vmux_2to1_Structure_080_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 16225 12 月 5 23:22 Vmux_2to1_Structure_081_deadAll.tree
-rw-rw-r-- 1 yanyusong yanyusong 17234 12 月 5 23:22 Vmux_2to1_Structure_082_clean.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_083_premittree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_2to1_Structure_084_expand.tree


```

-rw-rw-r-- 1 yanyusong yanyusong 89 12月 5 23:22 Vmux_2to1_Structure_085_const_cpp.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12月 5 23:22 Vmux_2to1_Structure_086_subst.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12月 5 23:22 Vmux_2to1_Structure_087_const_cpp.tree
-rw-rw-r-- 1 yanyusong yanyusong 17106 12月 5 23:22 Vmux_2to1_Structure_088_deadAll.tree
-rw-rw-r-- 1 yanyusong yanyusong 18383 12月 5 23:22 Vmux_2to1_Structure_092_cast.tree
-rw-rw-r-- 1 yanyusong yanyusong 21271 12月 5 23:22 Vmux_2to1_Structure_093_cuse.tree
-rw-rw-r-- 1 yanyusong yanyusong 22164 12月 5 23:22 Vmux_2to1_Structure_990_final.tree
-rw-rw-r-- 1 yanyusong yanyusong 18712 12月 5 23:22 Vmux_2to1_Structure__ALL.a
-rw-rw-r-- 1 yanyusong yanyusong 354 12月 5 23:22 Vmux_2to1_Structure__ALL.cpp
-rw-rw-r-- 1 yanyusong yanyusong 681 12月 5 23:22 Vmux_2to1_Structure__ALL.d
-rw-rw-r-- 1 yanyusong yanyusong 15952 12月 5 23:22 Vmux_2to1_Structure__ALL.o
-rw-rw-r-- 1 yanyusong yanyusong 857 12月 5 23:22 Vmux_2to1_Structure__Syms.cpp
-rw-rw-r-- 1 yanyusong yanyusong 1214 12月 5 23:22 Vmux_2to1_Structure__Syms.h
-rw-rw-r-- 1 yanyusong yanyusong 2036 12月 5 23:22 Vmux_2to1_Structure__Trace.cpp
-rw-rw-r-- 1 yanyusong yanyusong 4543 12月 5 23:22 Vmux_2to1_Structure__Trace_Slow.cpp
-rw-rw-r-- 1 yanyusong yanyusong 2690 12月 5 23:22 Vmux_2to1_Structure__024root.cpp
-rw-rw-r-- 1 yanyusong yanyusong 993 12月 5 23:22 Vmux_2to1_Structure__024root.h
-rw-rw-r-- 1 yanyusong yanyusong 2380 12月 5 23:22 Vmux_2to1_Structure__024root_Slow.cpp
-rw-rw-r-- 1 yanyusong yanyusong 22316 12月 5 23:22 Vmux_2to1_Structure__stats.txt
-rw-rw-r-- 1 yanyusong yanyusong 4218 12月 5 23:22 Vmux_2to1_Structure__ver.d
-rw-rw-r-- 1 yanyusong yanyusong 11497 12月 5 23:22 Vmux_2to1_Structure__verFiles.dat
-rw-rw-r-- 1 yanyusong yanyusong 1768 12月 5 23:22 Vmux_2to1_Structure_classes.mk
-rw-rw-r-- 1 yanyusong yanyusong 341 12月 5 23:22 Vmux_2to1_Structure_mux_2to1_Structure.vpp
-rw-rw-r-- 1 yanyusong yanyusong 158 12月 5 23:22 Vmux_2to1_Structure_my_and.vpp
-rw-rw-r-- 1 yanyusong yanyusong 149 12月 5 23:22 Vmux_2to1_Structure_my_not.vpp
-rw-rw-r-- 1 yanyusong yanyusong 155 12月 5 23:22 Vmux_2to1_Structure_my_or.vpp
-rw-rw-r-- 1 yanyusong yanyusong 1055 12月 5 23:22 dump.vcd
-rw-rw-r-- 1 yanyusong yanyusong 599 12月 5 23:22 mux_2to1_Structure_sim.d
-rw-rw-r-- 1 yanyusong yanyusong 6304 12月 5 23:22 mux_2to1_Structure_sim.o
-rw-rw-r-- 1 yanyusong yanyusong 446 12月 5 23:22 verilated.d
-rw-rw-r-- 1 yanyusong yanyusong 145408 12月 5 23:22 verilated.o
-rw-rw-r-- 1 yanyusong yanyusong 472 12月 5 23:22 verilated_vcd_c.d

```

实验过程 7：根据已经编译、生成好的文件，使用命令“gtkwave dump.vcd”在模拟示波器中观察图像，结果如下图。



实验三：通过行为建模实现模块

- 1) 实验环境 Ubuntu Linux 22.04.1 LTS (Jammy)
- 2) 工作目录 “/home/ce/OneStudentOneChip/DigitalCircuitsExperiments/DCE01/mux_2to1_Behavior”
- 3) 编辑软件 Visual Studio Code
- 4) 软件环境 Verilator (4.210), g++ (11.2.0-1ubuntu1)

实验目的：通过行为建模 2 选 1 多路选择器，了解多路选择器如何使用行为建模并学习 Verilog HDL 语言的使用方式，练习使用 Verilator（熟悉 Verilator 的相关调取方法，比如说在运行的时候需要在命令行除了说明文件名称以外还需要在命令行输入那些指令）

实验基础：根据 Verilog HDL 的语法，行为建模可以使用 always 块并让逻辑电路对敏感事件的变化执行相关的指令，同时注意到如果使用 always 块并有信号需要在 always 块中被赋值，则需要在声明寄存器变量（本实验中输出信号因为在 always 块中被赋值，所以需要 reg 型变量来初始化输出信号 y）

实验过程 1：通过实验基础和相关的 Verilog HDL 语法知识，写出 Verilog HDL 硬件描述文件：（文件名称：mux_2to1_Behavior.v）

```

module mux_2to1_Behavior (a, b, s, y);
    input a, b, s;
    output reg y;

    always @(*) begin
        if (s==0)
            begin
                y = a;
            end
        else
            begin
                y = b;
            end
        end
    end
endmodule

```

实验过程 2: 给该模块设计一个 C++语言的激励文件，内部包含了 8 中可能的端口情况 (a、b、s 各两种情况，0 或 1)。(文件名称: mux_2to1_Behavior_sim.cpp)

```

#include <verilated.h>
#include <verilated_vcd_c.h>
#include <obj_dir/Vmux_2to1_Datastream.h>

#include "/home/ce/OneStudentOneChip/include/verilated.h"
#include "/home/ce/OneStudentOneChip/include/verilated_vcd_c.h"
#include
"/home/ce/OneStudentOneChip/DigitalCircuitsExperiments/DCE01/mux_2to1_Behavior/obj_dir/Vmux_2to1_Structur
e.h"

#include <iostream>

VerilatedContext* contextp = NULL;
VerilatedVcdC* tfp = NULL;

static Vmux_2to1_Behavior* top;

void step_and_dump_wave()
{
    top -> eval();
    contextp -> timeInc(1);
    tfp -> dump(contextp -> time());
}

void sim_init() // Initilize the simulator
{
    contextp = new VerilatedContext;
    tfp = new VerilatedVcdC;
    top = new Vmux_2to1_Behavior;
    contextp -> traceEverOn(true);
    top -> trace (tfp, 0);
    tfp -> open("dump.vcd");
}

void sim_exit() // Terminate the simulator
{
    step_and_dump_wave();
    tfp -> close();
}

int main()
{
    // Print the Experiment Informations
    printf("Verilator Generated File simulator");
    printf("\n");
    printf("This is a Mutiplexer modeled by Behavior");
}

```

```

printf("\n");

// Simulator Main Parts
sim_init();

// Condition sab
top -> s = 0;
top -> a = 0;
top -> b = 0;
step_and_dump_wave();

// Condition saB
top -> s = 0;
top -> a = 0;
top -> b = 1;
step_and_dump_wave();

// Condition sAb
top -> s = 0;
top -> a = 1;
top -> b = 0;
step_and_dump_wave();

// Condition sAB
top -> s = 0;
top -> a = 1;
top -> b = 1;
step_and_dump_wave();

// Condition Sab
top -> s = 1;
top -> a = 0;
top -> b = 0;
step_and_dump_wave();

// Condition SaB
top -> s = 1;
top -> a = 0;
top -> b = 1;
step_and_dump_wave();

// Condition SAb
top -> s = 1;
top -> a = 1;
top -> b = 0;
step_and_dump_wave();

// Condition SAB
top -> s = 1;
top -> a = 1;
top -> b = 1;
step_and_dump_wave();

sim_exit();
}

```

实验过程 3: 使用 Verilator 进行编译文件 (需要同时导入写好的硬件描述文件和激励文件), 使用命令“verilator mux_2to1_Behavior.v mux_2to1_Behavior_sim.cpp --cc --exe --build --trace --debug”进行文件生成。在文件生成后, 使用命令“./Vmux_2to1_Behavior”运行 Verilator 编译好的文件, 目录“{working directory}/obj_dir”结构如下:

```

-rwxrwxr-x 1 yanyusong yanyusong 142104 12 月 5 23:22 Vmux_2to1_Behavior
-rw-rw-r-- 1 yanyusong yanyusong 5753 12 月 5 23:22 Vmux_2to1_Behavior.cpp
-rw-rw-r-- 1 yanyusong yanyusong 2856 12 月 5 23:22 Vmux_2to1_Behavior.h
-rw-rw-r-- 1 yanyusong yanyusong 1946 12 月 5 23:22 Vmux_2to1_Behavior.mk

```

-rw-rw-r-- 1 yanyusong yanyusong 10548 12 月 5 23:22 Vmux_2to1_Behavior.xml
-rw-rw-r-- 1 yanyusong yanyusong 146 12 月 5 23:22 Vmux_2to1_Behavior_001_linkcells.dot
-rw-rw-r-- 1 yanyusong yanyusong 2727 12 月 5 23:22 Vmux_2to1_Behavior_002_cells.tree
-rw-rw-r-- 1 yanyusong yanyusong 2836 12 月 5 23:22 Vmux_2to1_Behavior_007_link.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_008_linkInc.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_010_paramlink.tree
-rw-rw-r-- 1 yanyusong yanyusong 3480 12 月 5 23:22 Vmux_2to1_Behavior_012_width.tree
-rw-rw-r-- 1 yanyusong yanyusong 3438 12 月 5 23:22 Vmux_2to1_Behavior_014_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_015_assertpre.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_016_assert.tree
-rw-rw-r-- 1 yanyusong yanyusong 5226 12 月 5 23:22 Vmux_2to1_Behavior_018_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 5032 12 月 5 23:22 Vmux_2to1_Behavior_023_begin.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_024_tristate.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_025_unknown.tree
-rw-rw-r-- 1 yanyusong yanyusong 5568 12 月 5 23:22 Vmux_2to1_Behavior_026_inline.tree
-rw-rw-r-- 1 yanyusong yanyusong 5169 12 月 5 23:22 Vmux_2to1_Behavior_028_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 4256 12 月 5 23:22 Vmux_2to1_Behavior_029_deadDtypes.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_030_inst.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_031_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 7090 12 月 5 23:22 Vmux_2to1_Behavior_032_scope.tree
-rw-rw-r-- 1 yanyusong yanyusong 6951 12 月 5 23:22 Vmux_2to1_Behavior_033_linkdot.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_034_class.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_035_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_036_deadDtypesScoped.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_037_case.tree
-rw-rw-r-- 1 yanyusong yanyusong 192 12 月 5 23:22 Vmux_2to1_Behavior_038_task_call.dot
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_039_task.tree
-rw-rw-r-- 1 yanyusong yanyusong 7042 12 月 5 23:22 Vmux_2to1_Behavior_041_unroll.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_042_slice.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_043_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_044_life.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_045_table.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_046_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_047_deadDtypesScoped.tree
-rw-rw-r-- 1 yanyusong yanyusong 7298 12 月 5 23:22 Vmux_2to1_Behavior_048_active.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_049_split.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_050_splitas.tree
-rw-rw-r-- 1 yanyusong yanyusong 10601 12 月 5 23:22 Vmux_2to1_Behavior_051_tracedecl.tree
-rw-rw-r-- 1 yanyusong yanyusong 2865 12 月 5 23:22 Vmux_2to1_Behavior_052_gate_simp.dot
-rw-rw-r-- 1 yanyusong yanyusong 2640 12 月 5 23:22 Vmux_2to1_Behavior_053_gate_opt.dot
-rw-rw-r-- 1 yanyusong yanyusong 7736 12 月 5 23:22 Vmux_2to1_Behavior_054_gate.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_055_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 6432 12 月 5 23:22 Vmux_2to1_Behavior_056_deadAllScoped.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_057_reorder.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_058_delayed.tree
-rw-rw-r-- 1 yanyusong yanyusong 7892 12 月 5 23:22 Vmux_2to1_Behavior_059_activetop.tree
-rw-rw-r-- 1 yanyusong yanyusong 1823 12 月 5 23:22 Vmux_2to1_Behavior_060_orderg_pre.dot
-rw-rw-r-- 1 yanyusong yanyusong 146 12 月 5 23:22 Vmux_2to1_Behavior_061_acyc_simp.dot
-rw-rw-r-- 1 yanyusong yanyusong 1824 12 月 5 23:22 Vmux_2to1_Behavior_062_orderg_acyc.dot
-rw-rw-r-- 1 yanyusong yanyusong 1878 12 月 5 23:22 Vmux_2to1_Behavior_063_orderg_order.dot
-rw-rw-r-- 1 yanyusong yanyusong 1879 12 月 5 23:22 Vmux_2to1_Behavior_064_orderg_domain.dot
-rw-rw-r-- 1 yanyusong yanyusong 405 12 月 5 23:22 Vmux_2to1_Behavior_065_order_edges.txt
-rw-rw-r-- 1 yanyusong yanyusong 1877 12 月 5 23:22 Vmux_2to1_Behavior_066_orderg_done.dot
-rw-rw-r-- 1 yanyusong yanyusong 8374 12 月 5 23:22 Vmux_2to1_Behavior_067_order.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_068_genclk.tree
-rw-rw-r-- 1 yanyusong yanyusong 8008 12 月 5 23:22 Vmux_2to1_Behavior_069_clock.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_070_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_071_life.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_072_life_post.tree

```

-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_073_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_074_deadAllScoped.tree
-rw-rw-r-- 1 yanyusong yanyusong 8373 12 月 5 23:22 Vmux_2to1_Behavior_075_changed.tree
-rw-rw-r-- 1 yanyusong yanyusong 14815 12 月 5 23:22 Vmux_2to1_Behavior_076_trace.tree
-rw-rw-r-- 1 yanyusong yanyusong 12296 12 月 5 23:22 Vmux_2to1_Behavior_078_descope.tree
-rw-rw-r-- 1 yanyusong yanyusong 11321 12 月 5 23:22 Vmux_2to1_Behavior_079_combine.tree
-rw-rw-r-- 1 yanyusong yanyusong 11180 12 月 5 23:22 Vmux_2to1_Behavior_080_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 10864 12 月 5 23:22 Vmux_2to1_Behavior_081_deadAll.tree
-rw-rw-r-- 1 yanyusong yanyusong 11407 12 月 5 23:22 Vmux_2to1_Behavior_082_clean.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_083_premitt.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_084_expand.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_085_const_cpp.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_086_subst.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_2to1_Behavior_087_const_cpp.tree
-rw-rw-r-- 1 yanyusong yanyusong 11306 12 月 5 23:22 Vmux_2to1_Behavior_088_deadAll.tree
-rw-rw-r-- 1 yanyusong yanyusong 11573 12 月 5 23:22 Vmux_2to1_Behavior_092_cast.tree
-rw-rw-r-- 1 yanyusong yanyusong 14454 12 月 5 23:22 Vmux_2to1_Behavior_093_cuse.tree
-rw-rw-r-- 1 yanyusong yanyusong 15339 12 月 5 23:22 Vmux_2to1_Behavior_990_final.tree
-rw-rw-r-- 1 yanyusong yanyusong 16512 12 月 5 23:22 Vmux_2to1_Behavior__ALL.a
-rw-rw-r-- 1 yanyusong yanyusong 348 12 月 5 23:22 Vmux_2to1_Behavior__ALL.cpp
-rw-rw-r-- 1 yanyusong yanyusong 670 12 月 5 23:22 Vmux_2to1_Behavior__ALL.d
-rw-rw-r-- 1 yanyusong yanyusong 13808 12 月 5 23:22 Vmux_2to1_Behavior__ALL.o
-rw-rw-r-- 1 yanyusong yanyusong 849 12 月 5 23:22 Vmux_2to1_Behavior__Syms.cpp
-rw-rw-r-- 1 yanyusong yanyusong 1205 12 月 5 23:22 Vmux_2to1_Behavior__Syms.h
-rw-rw-r-- 1 yanyusong yanyusong 1735 12 月 5 23:22 Vmux_2to1_Behavior__Trace.cpp
-rw-rw-r-- 1 yanyusong yanyusong 3302 12 月 5 23:22 Vmux_2to1_Behavior__Trace_Slow.cpp
-rw-rw-r-- 1 yanyusong yanyusong 2634 12 月 5 23:22 Vmux_2to1_Behavior__024root.cpp
-rw-rw-r-- 1 yanyusong yanyusong 982 12 月 5 23:22 Vmux_2to1_Behavior__024root.h
-rw-rw-r-- 1 yanyusong yanyusong 2349 12 月 5 23:22 Vmux_2to1_Behavior__024root_Slow.cpp
-rw-rw-r-- 1 yanyusong yanyusong 21799 12 月 5 23:22 Vmux_2to1_Behavior__stats.txt
-rw-rw-r-- 1 yanyusong yanyusong 3983 12 月 5 23:22 Vmux_2to1_Behavior__ver.d
-rw-rw-r-- 1 yanyusong yanyusong 10833 12 月 5 23:22 Vmux_2to1_Behavior__verFiles.dat
-rw-rw-r-- 1 yanyusong yanyusong 1761 12 月 5 23:22 Vmux_2to1_Behavior_classes.mk
-rw-rw-r-- 1 yanyusong yanyusong 364 12 月 5 23:22 Vmux_2to1_Behavior_mux_2to1_Behavior.vpp
-rw-rw-r-- 1 yanyusong yanyusong 496 12 月 5 23:22 dump.vcd
-rw-rw-r-- 1 yanyusong yanyusong 545 12 月 5 23:22 mux_2to1_Behavior_sim.d
-rw-rw-r-- 1 yanyusong yanyusong 6296 12 月 5 23:22 mux_2to1_Behavior_sim.o
-rw-rw-r-- 1 yanyusong yanyusong 446 12 月 5 23:22 verilated.d
-rw-rw-r-- 1 yanyusong yanyusong 145408 12 月 5 23:22 verilated.o
-rw-rw-r-- 1 yanyusong yanyusong 472 12 月 5 23:22 verilated_vcd_c.d
-rw-rw-r-- 1 yanyusong yanyusong 50872 12 月 5 23:22 verilated_vcd_c.o

```

实验过程 4：根据已经编译、生成好的文件，使用命令“gtkwave dump.vcd”在模拟示波器中观察图像，结果如下图：



实验第二部分：使用 Verilog HDL 编写 4 选 1 多路选择器

实验一：使用 Case 语句实现模块

实验环境：

- 1) 操作系统 Ubuntu Linux 22.04.1 LTS (Jammy)
- 2) 工作目录 “/home/yanyusong/OneStudentOneChip/DigitalCircuitsExperiments/DCE01/mux_4to1_Case”
- 3) 编辑软件 Visual Studio Code
- 4) 软件环境 Verilator (5.003), g++ (11.2.0-1ubuntu1)

实验目的：通过 case 语句建模 4 选 1 多路选择器，了解 Verilog HDL 中 case 语句的用法，并学习巩固 Verilator 的使用。

实验基础：Verilog HDL 中 case 语句的用法，4 选 1 多路选择器用 Case 是因为其组件需要反馈选择的信号输入，因此可以用 case，对选择端的不同信号给出不同的反馈。

实验过程 1：写出通过 case 语句实现的 4 选 1 多路选择器的 Verilog HDL 硬件描述文件（文件名 mux_4to1_Case.v）

```
module mux_4to1_Case (a, s, y);
    input [3: 0] a;
    input [1: 0] s;
    output reg y;

    always @(s or a) begin
        case (s)
            0: y = a[0];
            1: y = a[1];
            2: y = a[2];
            3: y = a[3];
            default: y = 1'b0;
        endcase
    end
endmodule
```

实验过程 2：给该模块设计一个 C++ 语言的激励文件，内部包含了 8 种可能的端口情况（只是给出了部分）

```
##include <verilated.h>
##include <verilated_vcd_c.h>
##include <obj_dir/Vmux_4to1_Case.h>

#include "/usr/local/share/verilator/include/verilated.h"
#include "/usr/local/share/verilator/include/verilated_vcd_c.h"
#include
"/home/ce/OneStudentOneChip/DigitalCircuitsExperiments/DCE01/mux_4to1_Case/obj_dir/Vmux_4to1_Case.h"

#include <iostream>

VerilatedContext* contextp = NULL;
VerilatedVcdC* tfp = NULL;

static Vmux_4to1_Case* top;

void step_and_dump_wave()
{
    top -> eval();
    contextp -> timeInc(1);
    tfp -> dump(contextp -> time());
}

void sim_init() // Initilize the simulator
{
    contextp = new VerilatedContext;
    tfp = new VerilatedVcdC;
    top = new Vmux_4to1_Case;
    contextp -> traceEverOn(true);
    top -> trace (tfp, 0);
    tfp -> open("dump.vcd");
}

void sim_exit() // Terminate the simulator
{
    step_and_dump_wave();
    tfp -> close();
}
```

```

int main()
{
    // Print the Experiment Informations
    printf("Verilator Generated File simulator");
    printf("\n");
    printf("This is a Mutiplexer modeled by Case");
    printf("\n");

    // Simulator Main Parts
    sim_init();

    top -> s = 0b00;
    top -> a = 0b0001;
    step_and_dump_wave();

    top -> s = 0b00;
    top -> a = 0b0001;
    step_and_dump_wave();

    top -> s = 0b01;
    top -> a = 0b1110;
    step_and_dump_wave();

    top -> s = 0b01;
    top -> a = 0b0010;
    step_and_dump_wave();

    top -> s = 0b10;
    top -> a = 0b1010;
    step_and_dump_wave();

    top -> s = 0b10;
    top -> a = 0b0100;
    step_and_dump_wave();

    top -> s = 0b11;
    top -> a = 0b0111;
    step_and_dump_wave();

    top -> s = 0b11;
    top -> a = 0b1001;
    step_and_dump_wave();

    sim_exit();
}

```

实验过程 3: 使用 Verilator 进行编译文件（需要同事导入写好的硬件描述文件和激励文件），使用命令“verilator mux_4to1_Case.v mux_4to1_Case_sim.cpp --cc --exe --build --trace --debug”进行文件生成。在文件生成后，使用指令“./Vmux_4to1_Case”运行 Verilator 编译好的文件，目录“{working directory}/obj_dir”结构如下：

```

-rwxrwxr-x 1 yanyusong yanyusong 221248 12 月 5 23:22 Vmux_4to1_Case
-rw-rw-r-- 1 yanyusong yanyusong 5164 12 月 5 23:22 Vmux_4to1_Case.cpp
-rw-rw-r-- 1 yanyusong yanyusong 3040 12 月 5 23:22 Vmux_4to1_Case.h
-rw-rw-r-- 1 yanyusong yanyusong 1910 12 月 5 23:22 Vmux_4to1_Case.mk
-rw-rw-r-- 1 yanyusong yanyusong 36854 12 月 5 23:22 Vmux_4to1_Case.xml
-rw-rw-r-- 1 yanyusong yanyusong 5159 12 月 5 23:22 Vmux_4to1_Case_001_cells.tree
-rw-rw-r-- 1 yanyusong yanyusong 6551 12 月 5 23:22 Vmux_4to1_Case_006_linkjump.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_007_linkinc.tree
-rw-rw-r-- 1 yanyusong yanyusong 6511 12 月 5 23:22 Vmux_4to1_Case_009_linkdotparam.tree
-rw-rw-r-- 1 yanyusong yanyusong 7232 12 月 5 23:22 Vmux_4to1_Case_011_width.tree
-rw-rw-r-- 1 yanyusong yanyusong 7164 12 月 5 23:22 Vmux_4to1_Case_013_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_014_assertpre.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_015_assert.tree

```

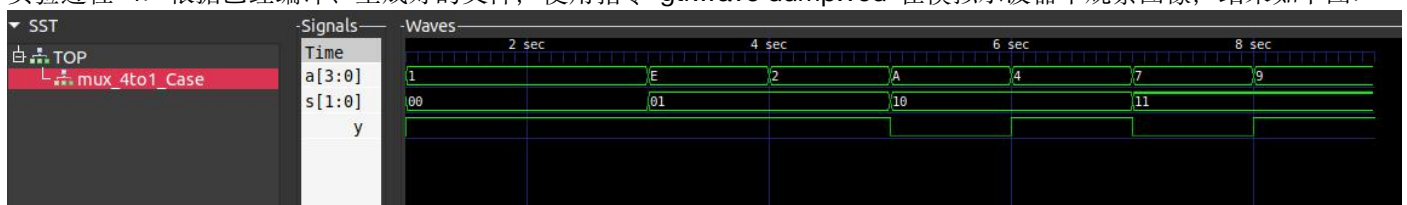
-rw-rw-r-- 1 yanyusong yanyusong 8603 12 月 5 23:22 Vmux_4to1_Case_017_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 8456 12 月 5 23:22 Vmux_4to1_Case_022_begin.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_023_tristate.tree
-rw-rw-r-- 1 yanyusong yanyusong 8552 12 月 5 23:22 Vmux_4to1_Case_024_unknown.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_025_dfg-extract.tree
-rw-rw-r-- 1 yanyusong yanyusong 8615 12 月 5 23:22 Vmux_4to1_Case_026_dfg-optimize.tree
-rw-rw-r-- 1 yanyusong yanyusong 8980 12 月 5 23:22 Vmux_4to1_Case_027_inline.tree
-rw-rw-r-- 1 yanyusong yanyusong 9059 12 月 5 23:22 Vmux_4to1_Case_029_dfg-optimize.tree
-rw-rw-r-- 1 yanyusong yanyusong 8972 12 月 5 23:22 Vmux_4to1_Case_030_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 8418 12 月 5 23:22 Vmux_4to1_Case_031_deadDtypes.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_032_inst.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_033_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 12060 12 月 5 23:22 Vmux_4to1_Case_034_scope.tree
-rw-rw-r-- 1 yanyusong yanyusong 11926 12 月 5 23:22 Vmux_4to1_Case_035_linkdot.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_036_class.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_037_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_038_deadDtypesScoped.tree
-rw-rw-r-- 1 yanyusong yanyusong 16547 12 月 5 23:22 Vmux_4to1_Case_039_case.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_040_task.tree
-rw-rw-r-- 1 yanyusong yanyusong 16438 12 月 5 23:22 Vmux_4to1_Case_042_unroll.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_043_slice.tree
-rw-rw-r-- 1 yanyusong yanyusong 11771 12 月 5 23:22 Vmux_4to1_Case_044_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_045_life.tree
-rw-rw-r-- 1 yanyusong yanyusong 25412 12 月 5 23:22 Vmux_4to1_Case_046_table.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_047_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 15105 12 月 5 23:22 Vmux_4to1_Case_048_deadDtypesScoped.tree
-rw-rw-r-- 1 yanyusong yanyusong 14642 12 月 5 23:22 Vmux_4to1_Case_049_active.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_050_split.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_051_splitas.tree
-rw-rw-r-- 1 yanyusong yanyusong 17711 12 月 5 23:22 Vmux_4to1_Case_052_tracedecl.tree
-rw-rw-r-- 1 yanyusong yanyusong 15457 12 月 5 23:22 Vmux_4to1_Case_053_gate.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_054_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 14105 12 月 5 23:22 Vmux_4to1_Case_055_deadAllScoped.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_056_reorder.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_057_delayed.tree
-rw-rw-r-- 1 yanyusong yanyusong 14097 12 月 5 23:22 Vmux_4to1_Case_058_activetop.tree
-rw-rw-r-- 1 yanyusong yanyusong 56691 12 月 5 23:22 Vmux_4to1_Case_059_sched.tree
-rw-rw-r-- 1 yanyusong yanyusong 54027 12 月 5 23:22 Vmux_4to1_Case_060_clock.tree
-rw-rw-r-- 1 yanyusong yanyusong 54392 12 月 5 23:22 Vmux_4to1_Case_061_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_062_life.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_063_life_post.tree
-rw-rw-r-- 1 yanyusong yanyusong 54983 12 月 5 23:22 Vmux_4to1_Case_064_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_065_deadAllScoped.tree
-rw-rw-r-- 1 yanyusong yanyusong 61241 12 月 5 23:22 Vmux_4to1_Case_066_trace.tree
-rw-rw-r-- 1 yanyusong yanyusong 48249 12 月 5 23:22 Vmux_4to1_Case_068_descope.tree
-rw-rw-r-- 1 yanyusong yanyusong 46538 12 月 5 23:22 Vmux_4to1_Case_069_combine.tree
-rw-rw-r-- 1 yanyusong yanyusong 46400 12 月 5 23:22 Vmux_4to1_Case_070_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 45504 12 月 5 23:22 Vmux_4to1_Case_071_deadAll.tree
-rw-rw-r-- 1 yanyusong yanyusong 47769 12 月 5 23:22 Vmux_4to1_Case_072_clean.tree
-rw-rw-r-- 1 yanyusong yanyusong 50007 12 月 5 23:22 Vmux_4to1_Case_073_premitt.tree
-rw-rw-r-- 1 yanyusong yanyusong 50144 12 月 5 23:22 Vmux_4to1_Case_074_expand.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_075_const_cpp.tree
-rw-rw-r-- 1 yanyusong yanyusong 48196 12 月 5 23:22 Vmux_4to1_Case_076_subst.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_077_const_cpp.tree
-rw-rw-r-- 1 yanyusong yanyusong 47563 12 月 5 23:22 Vmux_4to1_Case_078_deadAll.tree
-rw-rw-r-- 1 yanyusong yanyusong 48487 12 月 5 23:22 Vmux_4to1_Case_082_cast.tree
-rw-rw-r-- 1 yanyusong yanyusong 51546 12 月 5 23:22 Vmux_4to1_Case_083_common.tree
-rw-rw-r-- 1 yanyusong yanyusong 51601 12 月 5 23:22 Vmux_4to1_Case_084_variableorder.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_085_cuse.tree


```

-rw-rw-r-- 1 yanyusong yanyusong 53038 12 月 5 23:22 Vmux_4to1_Case_990_final.tree
-rw-rw-r-- 1 yanyusong yanyusong 24108 12 月 5 23:22 Vmux_4to1_Case__ALL.a
-rw-rw-r-- 1 yanyusong yanyusong 589 12 月 5 23:22 Vmux_4to1_Case__ALL.cpp
-rw-rw-r-- 1 yanyusong yanyusong 1019 12 月 5 23:22 Vmux_4to1_Case__ALL.d
-rw-rw-r-- 1 yanyusong yanyusong 20544 12 月 5 23:22 Vmux_4to1_Case__ALL.o
-rw-rw-r-- 1 yanyusong yanyusong 676 12 月 5 23:22 Vmux_4to1_Case__ConstPool_0.cpp
-rw-rw-r-- 1 yanyusong yanyusong 818 12 月 5 23:22 Vmux_4to1_Case__Syms.cpp
-rw-rw-r-- 1 yanyusong yanyusong 1163 12 月 5 23:22 Vmux_4to1_Case__Syms.h
-rw-rw-r-- 1 yanyusong yanyusong 1884 12 月 5 23:22 Vmux_4to1_Case__Trace_0.cpp
-rw-rw-r-- 1 yanyusong yanyusong 3379 12 月 5 23:22 Vmux_4to1_Case__Trace_0__Slow.cpp
-rw-rw-r-- 1 yanyusong yanyusong 1105 12 月 5 23:22 Vmux_4to1_Case__024root.h
-rw-rw-r-- 1 yanyusong yanyusong 1527 12 月 5 23:22 Vmux_4to1_Case__024root__DepSet_h2310cbeb__0.cpp
-rw-rw-r-- 1 yanyusong yanyusong 933 12 月 5 23:22
Vmux_4to1_Case__024root__DepSet_h2310cbeb__0__Slow.cpp
-rw-rw-r-- 1 yanyusong yanyusong 5698 12 月 5 23:22 Vmux_4to1_Case__024root__DepSet_hdea51003__0.cpp
-rw-rw-r-- 1 yanyusong yanyusong 5917 12 月 5 23:22
Vmux_4to1_Case__024root__DepSet_hdea51003__0__Slow.cpp
-rw-rw-r-- 1 yanyusong yanyusong 728 12 月 5 23:22 Vmux_4to1_Case__024root__Slow.cpp
-rw-rw-r-- 1 yanyusong yanyusong 57740 12 月 5 23:22 Vmux_4to1_Case__stats.txt
-rw-rw-r-- 1 yanyusong yanyusong 3549 12 月 5 23:22 Vmux_4to1_Case__ver.d
-rw-rw-r-- 1 yanyusong yanyusong 10103 12 月 5 23:22 Vmux_4to1_Case__verFiles.dat
-rw-rw-r-- 1 yanyusong yanyusong 1813 12 月 5 23:22 Vmux_4to1_Case__classes.mk
-rw-rw-r-- 1 yanyusong yanyusong 422 12 月 5 23:22 Vmux_4to1_Case_mux_4to1_Case.v.vpp
-rw-rw-r-- 1 yanyusong yanyusong 502 12 月 5 23:22 dump.vcd
-rw-rw-r-- 1 yanyusong yanyusong 692 12 月 5 23:22 mux_4to1_Case_sim.d
-rw-rw-r-- 1 yanyusong yanyusong 8864 12 月 5 23:22 mux_4to1_Case_sim.o
-rw-rw-r-- 1 yanyusong yanyusong 677 12 月 5 23:22 verilated.d
-rw-rw-r-- 1 yanyusong yanyusong 198344 12 月 5 23:22 verilated.o
-rw-rw-r-- 1 yanyusong yanyusong 410 12 月 5 23:22 verilated_threads.d
-rw-rw-r-- 1 yanyusong yanyusong 30144 12 月 5 23:22 verilated_threads.o
-rw-rw-r-- 1 yanyusong yanyusong 697 12 月 5 23:22 verilated_vcd_c.d
-rw-rw-r-- 1 yanyusong yanyusong 128952 12 月 5 23:22 verilated_vcd_c.o

```

实验过程 4：根据已经编译、生成好的文件，使用指令“gtkwave dump.vcd”在模拟示波器中观察图像，结果如下图：



实验二：更深入研究—在激励文件中增加更多的测试样例

实验环境：

- 1) 操作系统 Ubuntu Linux 22.04.1 LTS (Jammy)
- 2) 工作目录“/home/yanyusong/OneStudentOneChip/DigitalCircuitsExperiments/DCE01/mux_4to1_case_further”
- 3) 编辑软件：Visual Studio Code
- 4) 软件环境：Verilator (5.003), g++ (11.2.0-1ubuntu1)

实验目的：通过增加激励文件中的测试样例数量，从实验中要求的 8 个样例改为 64 个样例，进而模仿全部的电路、端口信号情况。

实验过程 1：复制实验一中写好的 4 选 1 多路选择器的硬件描述代码（同样使用 case 进行建模），并对其重新命名

实验过程 2：修改激励文件：

```

//#include <verilated.h>
//#include <verilated_vcd_c.h>
//#include <obj_dir/Vmux_4to1_Case.h>

```

```

#include "/usr/local/share/verilator/include/verilated.h"
#include "/usr/local/share/verilator/include/verilated_vcd_c.h"

```

```

#include
"/home/ce/OneStudentOneChip/DigitalCircuitsExperiments/DCE01/mux_4to1_case_further/obj_dir/Vmux_4to1_Case.
h"

#include <iostream>

VerilatedContext* contextp = NULL;
VerilatedVcdC* tfp = NULL;

static Vmux_4to1_Case* top;

void step_and_dump_wave()
{
    top -> eval();
    contextp -> timeInc(1);
    tfp -> dump(contextp -> time());
}

void sim_init() // Initilize the simulator
{
    contextp = new VerilatedContext;
    tfp = new VerilatedVcdC;
    top = new Vmux_4to1_Case;
    contextp -> traceEverOn(true);
    top -> trace (tfp, 0);
    tfp -> open("dump.vcd");
}

void sim_exit() // Terminate the simulator
{
    step_and_dump_wave();
    tfp -> close();
}

int main()
{
    // Print the Experiment Informations
    printf("Verilator Generated File simulator");
    printf("\n");
    printf("This is a Mutiplexer modeled by Case");
    printf("\n");

    // Simulator Main Parts
    sim_init();

    top -> s = 0b00;
    top -> a = 0b0000; // 0
    step_and_dump_wave();
    top -> s = 0b00;
    top -> a = 0b0001; // 1
    step_and_dump_wave();
    top -> s = 0b00;
    top -> a = 0b0010; // 2
    step_and_dump_wave();
    top -> s = 0b00;
    top -> a = 0b0011; // 3
    step_and_dump_wave();
    top -> s = 0b00;
    top -> a = 0b0100; // 4
    step_and_dump_wave();
    top -> s = 0b00;
    top -> a = 0b0101; // 5
    step_and_dump_wave();
    top -> s = 0b00;

```

```
top -> a = 0b0110; // 6
step_and_dump_wave();
top -> s = 0b00;
top -> a = 0b0111; // 7
step_and_dump_wave();
top -> s = 0b00;
top -> a = 0b1000; // 8
step_and_dump_wave();
top -> s = 0b00;
top -> a = 0b1001; // 9
step_and_dump_wave();
top -> s = 0b00;
top -> a = 0b1010; // 10
step_and_dump_wave();
top -> s = 0b00;
top -> a = 0b1011; // 11
step_and_dump_wave();
top -> s = 0b00;
top -> a = 0b1100; // 12
step_and_dump_wave();
top -> s = 0b00;
top -> a = 0b1101; // 13
step_and_dump_wave();
top -> s = 0b00;
top -> a = 0b1110; // 14
step_and_dump_wave();
top -> s = 0b00;
top -> a = 0b1111; // 15
step_and_dump_wave();
```

```
top -> s = 0b01;
top -> a = 0b0000; // 0
step_and_dump_wave();
top -> s = 0b01;
top -> a = 0b0001; // 1
step_and_dump_wave();
top -> s = 0b01;
top -> a = 0b0010; // 2
step_and_dump_wave();
top -> s = 0b01;
top -> a = 0b0011; // 3
step_and_dump_wave();
top -> s = 0b01;
top -> a = 0b0100; // 4
step_and_dump_wave();
top -> s = 0b01;
top -> a = 0b0101; // 5
step_and_dump_wave();
top -> s = 0b01;
top -> a = 0b0110; // 6
step_and_dump_wave();
top -> s = 0b01;
top -> a = 0b0111; // 7
step_and_dump_wave();
top -> s = 0b01;
top -> a = 0b1000; // 8
step_and_dump_wave();
top -> s = 0b01;
top -> a = 0b1001; // 9
step_and_dump_wave();
top -> s = 0b01;
top -> a = 0b1010; // 10
step_and_dump_wave();
top -> s = 0b01;
```

```
top -> a = 0b1011; // 11
step_and_dump_wave();
top -> s = 0b01;
top -> a = 0b1100; // 12
step_and_dump_wave();
top -> s = 0b01;
top -> a = 0b1101; // 13
step_and_dump_wave();
top -> s = 0b01;
top -> a = 0b1110; // 14
step_and_dump_wave();
top -> s = 0b01;
top -> a = 0b1111; // 15
step_and_dump_wave();
```

```
top -> s = 0b10;
top -> a = 0b0000; // 0
step_and_dump_wave();
top -> s = 0b10;
top -> a = 0b0001; // 1
step_and_dump_wave();
top -> s = 0b10;
top -> a = 0b0010; // 2
step_and_dump_wave();
top -> s = 0b10;
top -> a = 0b0011; // 3
step_and_dump_wave();
top -> s = 0b10;
top -> a = 0b0100; // 4
step_and_dump_wave();
top -> s = 0b10;
top -> a = 0b0101; // 5
step_and_dump_wave();
top -> s = 0b10;
top -> a = 0b0110; // 6
step_and_dump_wave();
top -> s = 0b10;
top -> a = 0b0111; // 7
step_and_dump_wave();
top -> s = 0b10;
top -> a = 0b1000; // 8
step_and_dump_wave();
top -> s = 0b10;
top -> a = 0b1001; // 9
step_and_dump_wave();
top -> s = 0b10;
top -> a = 0b1010; // 10
step_and_dump_wave();
top -> s = 0b10;
top -> a = 0b1011; // 11
step_and_dump_wave();
top -> s = 0b10;
top -> a = 0b1100; // 12
step_and_dump_wave();
top -> s = 0b10;
top -> a = 0b1101; // 13
step_and_dump_wave();
top -> s = 0b10;
top -> a = 0b1110; // 14
step_and_dump_wave();
top -> s = 0b10;
top -> a = 0b1111; // 15
step_and_dump_wave();
```

```

top -> s = 0b11;
top -> a = 0b0000; // 0
step_and_dump_wave();
top -> s = 0b11;
top -> a = 0b0001; // 1
step_and_dump_wave();
top -> s = 0b11;
top -> a = 0b0010; // 2
step_and_dump_wave();
top -> s = 0b11;
top -> a = 0b0011; // 3
step_and_dump_wave();
top -> s = 0b11;
top -> a = 0b0100; // 4
step_and_dump_wave();
top -> s = 0b11;
top -> a = 0b0101; // 5
step_and_dump_wave();
top -> s = 0b11;
top -> a = 0b0110; // 6
step_and_dump_wave();
top -> s = 0b11;
top -> a = 0b0111; // 7
step_and_dump_wave();
top -> s = 0b11;
top -> a = 0b1000; // 8
step_and_dump_wave();
top -> s = 0b11;
top -> a = 0b1001; // 9
step_and_dump_wave();
top -> s = 0b11;
top -> a = 0b1010; // 10
step_and_dump_wave();
top -> s = 0b11;
top -> a = 0b1011; // 11
step_and_dump_wave();
top -> s = 0b11;
top -> a = 0b1100; // 12
step_and_dump_wave();
top -> s = 0b11;
top -> a = 0b1101; // 13
step_and_dump_wave();
top -> s = 0b11;
top -> a = 0b1110; // 14
step_and_dump_wave();
top -> s = 0b11;
top -> a = 0b1111; // 15
step_and_dump_wave();

sim_exit();
}

```

实验过程 3: 使用 Verilator 进行编译文件 (方法同实验一, 这里不再阐述), 最后目录结构为:

```

-rwxrwxr-x 1 yanyusong yanyusong 221248 12 月 5 23:22 Vmux_4to1_Case
-rw-rw-r-- 1 yanyusong yanyusong 5164 12 月 5 23:22 Vmux_4to1_Case.cpp
-rw-rw-r-- 1 yanyusong yanyusong 3040 12 月 5 23:22 Vmux_4to1_Case.h
-rw-rw-r-- 1 yanyusong yanyusong 1910 12 月 5 23:22 Vmux_4to1_Case.mk
-rw-rw-r-- 1 yanyusong yanyusong 36854 12 月 5 23:22 Vmux_4to1_Case.xml
-rw-rw-r-- 1 yanyusong yanyusong 5159 12 月 5 23:22 Vmux_4to1_Case_001_cells.tree
-rw-rw-r-- 1 yanyusong yanyusong 6551 12 月 5 23:22 Vmux_4to1_Case_006_linkjump.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_007_linkinc.tree
-rw-rw-r-- 1 yanyusong yanyusong 6511 12 月 5 23:22 Vmux_4to1_Case_009_linkdotparam.tree
-rw-rw-r-- 1 yanyusong yanyusong 7232 12 月 5 23:22 Vmux_4to1_Case_011_width.tree

```

-rw-rw-r-- 1 yanyusong yanyusong 7164 12 月 5 23:22 Vmux_4to1_Case_013_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_014_assertpre.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_015_assert.tree
-rw-rw-r-- 1 yanyusong yanyusong 8603 12 月 5 23:22 Vmux_4to1_Case_017_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 8456 12 月 5 23:22 Vmux_4to1_Case_022_begin.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_023_tristate.tree
-rw-rw-r-- 1 yanyusong yanyusong 8552 12 月 5 23:22 Vmux_4to1_Case_024_unknown.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_025_dfg-extract.tree
-rw-rw-r-- 1 yanyusong yanyusong 8615 12 月 5 23:22 Vmux_4to1_Case_026_dfg-optimize.tree
-rw-rw-r-- 1 yanyusong yanyusong 8980 12 月 5 23:22 Vmux_4to1_Case_027_inline.tree
-rw-rw-r-- 1 yanyusong yanyusong 9059 12 月 5 23:22 Vmux_4to1_Case_029_dfg-optimize.tree
-rw-rw-r-- 1 yanyusong yanyusong 8972 12 月 5 23:22 Vmux_4to1_Case_030_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 8418 12 月 5 23:22 Vmux_4to1_Case_031_deadDtypes.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_032_inst.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_033_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 12060 12 月 5 23:22 Vmux_4to1_Case_034_scope.tree
-rw-rw-r-- 1 yanyusong yanyusong 11926 12 月 5 23:22 Vmux_4to1_Case_035_linkdot.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_036_class.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_037_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 87 12 月 5 23:22 Vmux_4to1_Case_038_deadDtypesScoped.tree
-rw-rw-r-- 1 yanyusong yanyusong 16547 12 月 5 23:22 Vmux_4to1_Case_039_case.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_040_task.tree
-rw-rw-r-- 1 yanyusong yanyusong 16438 12 月 5 23:22 Vmux_4to1_Case_042_unroll.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_043_slice.tree
-rw-rw-r-- 1 yanyusong yanyusong 11771 12 月 5 23:22 Vmux_4to1_Case_044_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_045_life.tree
-rw-rw-r-- 1 yanyusong yanyusong 25412 12 月 5 23:22 Vmux_4to1_Case_046_table.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_047_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 15105 12 月 5 23:22 Vmux_4to1_Case_048_deadDtypesScoped.tree
-rw-rw-r-- 1 yanyusong yanyusong 14642 12 月 5 23:22 Vmux_4to1_Case_049_active.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_050_split.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_051_splitas.tree
-rw-rw-r-- 1 yanyusong yanyusong 17711 12 月 5 23:22 Vmux_4to1_Case_052_tracedecl.tree
-rw-rw-r-- 1 yanyusong yanyusong 15457 12 月 5 23:22 Vmux_4to1_Case_053_gate.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_054_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 14105 12 月 5 23:22 Vmux_4to1_Case_055_deadAllScoped.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_056_reorder.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_057_delayed.tree
-rw-rw-r-- 1 yanyusong yanyusong 14097 12 月 5 23:22 Vmux_4to1_Case_058_activetop.tree
-rw-rw-r-- 1 yanyusong yanyusong 56691 12 月 5 23:22 Vmux_4to1_Case_059_sched.tree
-rw-rw-r-- 1 yanyusong yanyusong 54027 12 月 5 23:22 Vmux_4to1_Case_060_clock.tree
-rw-rw-r-- 1 yanyusong yanyusong 54392 12 月 5 23:22 Vmux_4to1_Case_061_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_062_life.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_063_life_post.tree
-rw-rw-r-- 1 yanyusong yanyusong 54983 12 月 5 23:22 Vmux_4to1_Case_064_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_065_deadAllScoped.tree
-rw-rw-r-- 1 yanyusong yanyusong 61241 12 月 5 23:22 Vmux_4to1_Case_066_trace.tree
-rw-rw-r-- 1 yanyusong yanyusong 48249 12 月 5 23:22 Vmux_4to1_Case_068_descope.tree
-rw-rw-r-- 1 yanyusong yanyusong 46538 12 月 5 23:22 Vmux_4to1_Case_069_combine.tree
-rw-rw-r-- 1 yanyusong yanyusong 46400 12 月 5 23:22 Vmux_4to1_Case_070_const.tree
-rw-rw-r-- 1 yanyusong yanyusong 45504 12 月 5 23:22 Vmux_4to1_Case_071_deadAll.tree
-rw-rw-r-- 1 yanyusong yanyusong 47769 12 月 5 23:22 Vmux_4to1_Case_072_clean.tree
-rw-rw-r-- 1 yanyusong yanyusong 50007 12 月 5 23:22 Vmux_4to1_Case_073_premittree
-rw-rw-r-- 1 yanyusong yanyusong 50144 12 月 5 23:22 Vmux_4to1_Case_074_expand.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_075_const_cpp.tree
-rw-rw-r-- 1 yanyusong yanyusong 48196 12 月 5 23:22 Vmux_4to1_Case_076_subst.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_077_const_cpp.tree
-rw-rw-r-- 1 yanyusong yanyusong 47563 12 月 5 23:22 Vmux_4to1_Case_078_deadAll.tree
-rw-rw-r-- 1 yanyusong yanyusong 48487 12 月 5 23:22 Vmux_4to1_Case_082_cast.tree

```

-rw-rw-r-- 1 yanyusong yanyusong 51546 12 月 5 23:22 Vmux_4to1_Case_083_common.tree
-rw-rw-r-- 1 yanyusong yanyusong 51601 12 月 5 23:22 Vmux_4to1_Case_084_variableorder.tree
-rw-rw-r-- 1 yanyusong yanyusong 89 12 月 5 23:22 Vmux_4to1_Case_085_cuse.tree
-rw-rw-r-- 1 yanyusong yanyusong 53038 12 月 5 23:22 Vmux_4to1_Case_990_final.tree
-rw-rw-r-- 1 yanyusong yanyusong 24108 12 月 5 23:22 Vmux_4to1_Case__ALL.a
-rw-rw-r-- 1 yanyusong yanyusong 589 12 月 5 23:22 Vmux_4to1_Case__ALL.cpp
-rw-rw-r-- 1 yanyusong yanyusong 1019 12 月 5 23:22 Vmux_4to1_Case__ALL.d
-rw-rw-r-- 1 yanyusong yanyusong 20544 12 月 5 23:22 Vmux_4to1_Case__ALL.o
-rw-rw-r-- 1 yanyusong yanyusong 676 12 月 5 23:22 Vmux_4to1_Case__ConstPool_0.cpp
-rw-rw-r-- 1 yanyusong yanyusong 818 12 月 5 23:22 Vmux_4to1_Case__Syms.cpp
-rw-rw-r-- 1 yanyusong yanyusong 1163 12 月 5 23:22 Vmux_4to1_Case__Syms.h
-rw-rw-r-- 1 yanyusong yanyusong 1884 12 月 5 23:22 Vmux_4to1_Case__Trace_0.cpp
-rw-rw-r-- 1 yanyusong yanyusong 3379 12 月 5 23:22 Vmux_4to1_Case__Trace_0__Slow.cpp
-rw-rw-r-- 1 yanyusong yanyusong 1105 12 月 5 23:22 Vmux_4to1_Case__024root.h
-rw-rw-r-- 1 yanyusong yanyusong 1527 12 月 5 23:22 Vmux_4to1_Case__024root__DepSet_h2310cbeb__0.cpp
-rw-rw-r-- 1 yanyusong yanyusong 933 12 月 5 23:22
Vmux_4to1_Case__024root__DepSet_h2310cbeb__0__Slow.cpp
-rw-rw-r-- 1 yanyusong yanyusong 5698 12 月 5 23:22 Vmux_4to1_Case__024root__DepSet_hdea51003__0.cpp
-rw-rw-r-- 1 yanyusong yanyusong 5917 12 月 5 23:22
Vmux_4to1_Case__024root__DepSet_hdea51003__0__Slow.cpp
-rw-rw-r-- 1 yanyusong yanyusong 728 12 月 5 23:22 Vmux_4to1_Case__024root__Slow.cpp
-rw-rw-r-- 1 yanyusong yanyusong 57740 12 月 5 23:22 Vmux_4to1_Case__stats.txt
-rw-rw-r-- 1 yanyusong yanyusong 3549 12 月 5 23:22 Vmux_4to1_Case__ver.d
-rw-rw-r-- 1 yanyusong yanyusong 10103 12 月 5 23:22 Vmux_4to1_Case__verFiles.dat
-rw-rw-r-- 1 yanyusong yanyusong 1813 12 月 5 23:22 Vmux_4to1_Case_classes.mk
-rw-rw-r-- 1 yanyusong yanyusong 422 12 月 5 23:22 Vmux_4to1_Case_mux_4to1_Case.v.vpp
-rw-rw-r-- 1 yanyusong yanyusong 1257 12 月 5 23:22 dump.vcd
-rw-rw-r-- 1 yanyusong yanyusong 700 12 月 5 23:22 mux_4to1_Case_sim.d
-rw-rw-r-- 1 yanyusong yanyusong 13008 12 月 5 23:22 mux_4to1_Case_sim.o
-rw-rw-r-- 1 yanyusong yanyusong 677 12 月 5 23:22 verilated.d
-rw-rw-r-- 1 yanyusong yanyusong 198344 12 月 5 23:22 verilated.o
-rw-rw-r-- 1 yanyusong yanyusong 410 12 月 5 23:22 verilated_threads.d
-rw-rw-r-- 1 yanyusong yanyusong 30144 12 月 5 23:22 verilated_threads.o
-rw-rw-r-- 1 yanyusong yanyusong 697 12 月 5 23:22 verilated_vcd_c.d
-rw-rw-r-- 1 yanyusong yanyusong 128952 12 月 5 23:22 verilated_vcd_c.o

```

实验过程 4: 根据已经编译、生成好的文件, 使用指令“gtkwave dump.vcd”在模拟示波器中观察图像, 结果如下图:



实验总结

- 1) 通过本次实验, 对 Verilog HDL 语言的相关建模方法有了更为深刻的掌握
- 2) 通过本次实验, 对 Verilator 编译器的使用有了初步的了解, 已经从不会使用到可以熟练地编译代码并生成波形图文件
- 3) 通过本次实验, 学会了使用 GTKWave 分析、查看波形图文件的方式