青岛若贝电子有限公司

集成电路虚拟仿真实验平台介绍和相关案例

集成电路虚拟仿真实验室平台

支持数字电路、 EDA 、可编程逻辑器件、 计算机原理与系统、SOPC 片上系统实验课程

- **建设思想：**

指导思想：集成电路产业是信息技术产业的核心，是支撑经济社会发展和保•障国家安全的战略性、基础性和先导性产业，当前和今后一段时期是我国集成电路产业发展的重要战略机遇期和攻坚期。为主动适应和引领经济发展新常态，贯彻落实《国家集成电路产业发展推进纲要》《国家中长期教育改革和发展规划纲要（2010-2020 年）》，创新集成电路相关专业人才培养机制，提高人才培养质量，提升我国集成电路产业持续发展能力，教育部于 2016 发布了《教育部等七部门关于加强集成电路人才培养的意见》（教高〔2016〕1 号）文件。

意见指出，我国集成电路人才培养，需扩大集成电路相关学科专业人才培养规模、加强集成电路相关学科专业和院系建设、创新集成电路人才培养机制、建设集成电路人才培养公共实践平台、建设产学合作育人服务平台、提升集成电路从业人员专业能力、优化集成电路人才引进与使用、加大对集成电路人才培养的政策支持、加强对集成电路产业人才工作的领导共九个方面的建设。

筹建集成电路虚拟仿真实验室，与国内集成电路发展和人才培养是密不可分的。根据中国电子信息产业发展研究院 2017 年发布的白皮书，中国集成电路人才缺口 40 万。集成电路产业的发展依托于集成电路设计工具，但是国外提供的 EDA 工具基本上对一些常用的 IP 进行知识产权封锁，不利于高校人才培养。为增强学生 IP 设计能力，并解决初学者在集成电路设计方向的种种困惑，从感兴趣到迷茫、头疼、失去兴趣、悟道、加深兴趣一直到痴迷的过程。一种提供可视化、面向对象、透明 IP 的 EDA 设计工具将促进集成电路设计人员的设计能力和动手能力。

● 集成电路建设的必要性：

“集成电路设计建设项目”显著特征，是从产业需求出发，主导企业深度参与高校人才培养，实现企业与高校协同育人、协同创新和成果转化，为产业高速发展培育和储备更多“适销对路”的高质量人才。

在当前提倡“新工科”研究与实践的大环境下，强调学生的创新创业能力，数字电路的教学改革应该在以激发学生学以致用为主、将传统的理论灌输变成动手实践、将抽象的逻辑概念变成可视化模块化的虚拟设备，采用永不过时的虚拟实验环境代替被产业淘汰的逻辑器件，以可重构的 FPGA 数字实践代替老化稳定性差的芯片堆叠，以数字课堂与实践课堂相结合的方式，来重新认知学习数字电路基础。实践课程基于 Robei 集成电路平台可以实现集成电路快速入手，在电子设计大赛，毕业设计等项目有相当大的优势。

数字电路，数字电子技术基础和 EDA 课程等相关课程都包括理论和实践。教育部鼓励高校开设课程以理论和实践相结合。其他 EDA 工具 Modisim、synopsys 等 EDA 工具是国内引进工具，其中很多 IP 设置了黑匣子，最终只会用这几家 EDA 公司的 IP，不利于学生学习底层设计。同时，集成电路是需要了解底层设计，才能设计出更高端的 IP。Robei 实验平台重点突出学生将理论和实验实践课程相结合，增强学生的动手能力，把理论应用与实践案例。

- **集成电路虚拟仿真平台适合用于以下专业教学或者课程设计：**

电子信息、通信工程、自动化、集成电路、电气，光电、计算机、机电、微固等相关专业

❖ **集成电路虚拟仿真实验室：**

采用 Robei 可视化芯片设计软件和开发板以及配套教材和 IP 设计源代码。从设计到仿真，综合以及板级验证。基于实验室已有计算机，一台电脑配套一套实验开发平台。

❖ **集成电路虚拟仿真实验平台包括：**

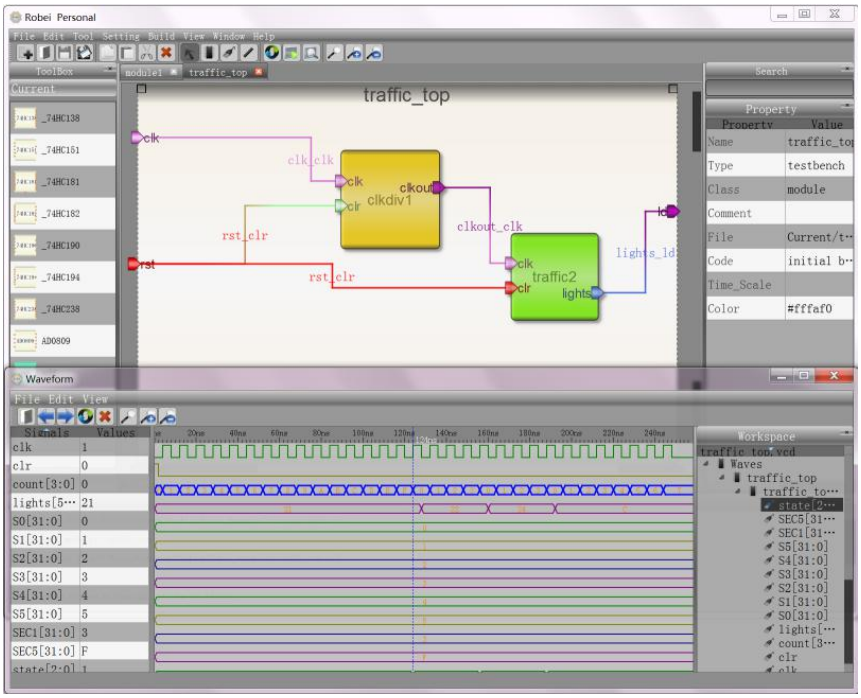
1、Robei EDA 软件　2、 核心开发板　3、实验箱　 4、 教材　5、设计源代码

❖ **Robei 简介：**

Robei EDA 芯片设计软件： 是一款全新的拥有自主知识产权的集成电路设计工具。同时也是一种低投资的集成电路设计软件。不仅具备传统设计工具的代码编写、编译、仿真功能，并且增加了可视化和模块化的设计理念，具有模块设计透明化，方便模块重新利用，加快设计进度等特点。可实现顶层跨平台，图形和代码相结合的设计优势，将 IC 设计简化到模块、端口、导线三个基本元素，并自动生成端口定义的 Verilog 代码以及约束文件。

Robei 可视化芯片设计软件可以提供 74 系类器件的仿真与模拟。可以仿真 FPGA、MCU、状态机、接口电路设计等，并提供了大量的 IP 库:计数器、Alu、流水灯自动售货机、移位寄存器、浮点计算单元（FPU）、FIFO、串口通信、RISC 处理器等。

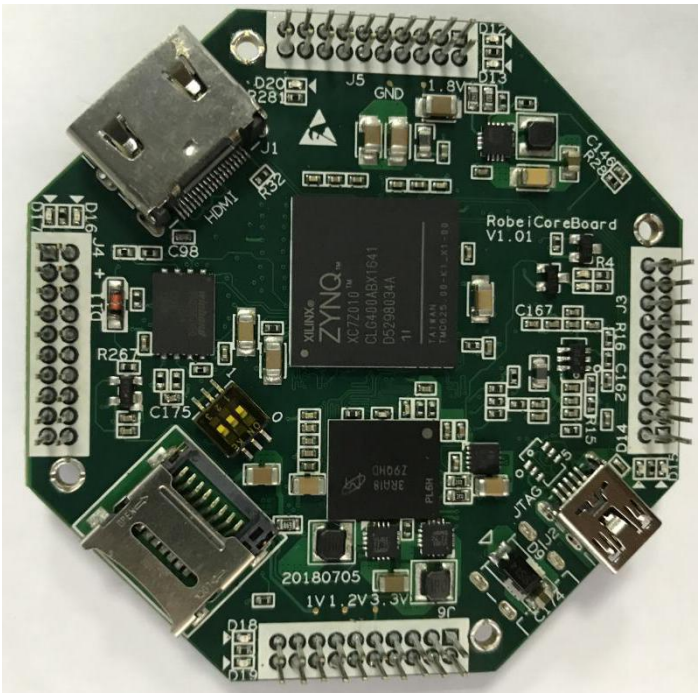
1、Robei　EDA 软件界面图



❖ **Robei 可视化芯片设计软件参数：**

	序号	参数
Robei EDA 软件技术指标	1	接口与例化代码自动生成
	2	引脚可移动，引脚颜色可改
	3	软件不超过 8M，下载方便
	4	可以分层设计，IP 模块双击打开
	5	提供搜索资源库的接口，配备免费的参考设计资源
	6	支持 Verilog 仿真
	7	支持波形查看，可打开 VCD 文件
	8	临近波形采用不同颜色进行区分
	9	生成的 Verilog 代码可以移植到不同的 FPGA 平台使用
	10	自动生成设计模块的文档

2、开发板



❖ **配套核心开发板参数：**

核心板组件	1	FPGA 型号	XC7Z010-1CLG400C，BGA400 封装，主频最高 633MHz
	2	编程接口	USB2.0/JTAG
	3	板载晶振	125MHZ
	4	数据存储器	LPDDR2 SDRAM，32 位宽，512MB 运行频率 800MHz
	5	程序存储器	flash，256Mb
	6	Serdes 接口	USB2.0 /USB OTG
	7	图形图像接口	HDMI
	8	指示灯	LED，4 个

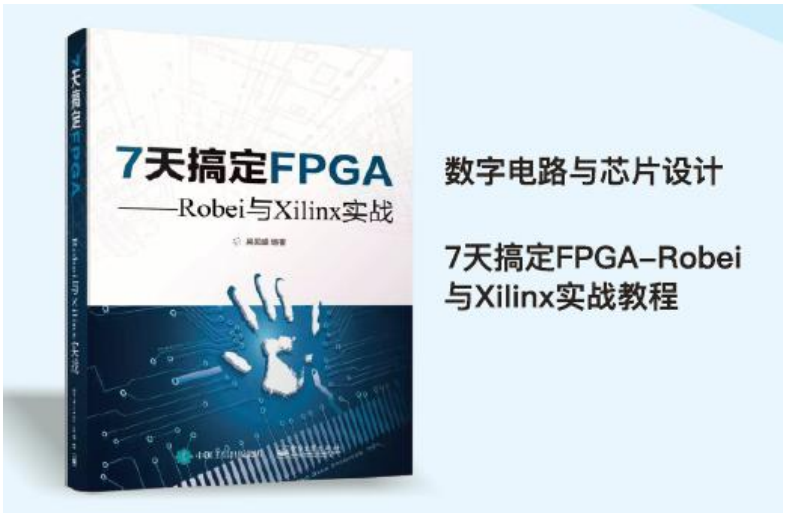
3、实验箱

❖ **实验箱**

实验箱组件	1	显示组件 1	LED 灯>=8 个
	2	显示组件 2	共阴极数码管，≥2 个
	3	LCD 显示屏	一块
	4	SPI 读写	flash controller
	5	电机接口	1 路步进电机/2 路直流电机
	6	按键输入	按键开关（≥4 个）/ 4X4 键盘
	7	传感器	温度型，1 线接口
	8	VGA 接口	VGA 接口

4、配套教材

1、 7 天搞定 FPGA---Robei 与 Xilinx 实战； 2、数字电路与芯片设计 （在编教材）



5、设计源代码

❖ **可实现案例：**

软件案例	1	流水灯控制
	2	基本 ALU 设计
	3	小数加减法
	4	除法器设计
	5	FIFO 设计
	6	SPI 总线设计
	7	串口通信设计
	8	8 位 RISC CPU 设计
	9	FIR 滤波器

● 建设方案：

根据高校情况，公司配有教育部产学研项目，对课程改革、师资培训等在数字电路、数字电子技术基础、EDA 课程等相关课程协助高校培养人才，并提供配套教材和开发板；为了更好的落实平台和课程建设，公司定期组织工程师对相关老师进行培训。

实验课时设计：根据专业需求选择不同课程

实施阶段	课程	内容	目标	理论+实践（课时）
第一阶段	数字电路基础	74 系类器件	掌握逻辑电路、时序电路、触发器、计数器等基础逻辑	48+16
第二阶段	EDA	Verilog 语言	掌握 Verilog 语言，并利用 Verilog 语言设计逻辑计算与接口电路	24+16
第三阶段	可编程逻辑器件	FPGA 设计	掌握 FPGA 器件，理解 FPGA 开发流程，学习电路接口协议，SPI、I2C、UART 等协议	32+16
第四阶段	计算机原理与系统或 SOPC 片上系统	MIPS 架构	掌握计算机组成结构、理解 CPU 设计原理，熟悉 MIPS 架构与指令集	32+16

- **实践课时安排**

第一阶段:

数字电路基础	内容	课时
	Robei 可视化芯片设计软件操作入门	2
	逻辑运算电路设计（74LS 系列逻辑门）	2
	算数运算单元	2
	时序电路设计（触发器、寄存器等）	2
	编码器与译码器	2
	移位寄存器	2
	计数器设计	2
	FSM 状态机	2

第二阶段：

EDA	内容	课时
	Verilog 语言基础演练	3
	同步与异步移位寄存器	2
	SRAM 设计	3
	FPGA 与开发工具	2
	流水灯设计	2
	自动售货机	2
	SPI Flash 读写	2

第三阶段：

可编程逻辑器件	内容	课时
	Vivado 开发环境	2
	同步与异步 FIFO	2
	I2C 协议	2
	乘法器设计	2
	UART 串口通信	2
	FIR 滤波器	3
	傅里叶变换 FFT	3

第四阶段

计算机原理与系统或 SOC 片上系统	内容	课时
	计算机原理简介	2
	RISC CPU 设计	3
	MIPS 架构设计与仿真	4
	AI 芯片设计	3
	MIPS+AI 的 SOC 设计	4

基于 Robei EDA 软件案例

(1) 集成电路逻辑基础（与门、或门、非门、数据选择器、编译码器、数字计算单元：加法、减法、乘法）	(2) Verilog 实现 UART 之发送与接收模块设计
(3) 集成电路中时序电路基础（计数器、触发器、锁存器、SRAM）	(4) 改进的 booth 编码和 wallace 树部分积压缩法设计 8*8 乘法器
(5) 流水灯控制	(6) 七段数码管译码器
(7) 基本 ALU 设计	(8) 电话振铃音调可调设计
(9) 小数加减法	(10) HDB3 译码设计
(11) 除法器设计	(12) 模拟简易 SCCB 总线传输
(13) FIFO 设计	(14) 多功能数字时钟
(15) SPI 总线设计	(16) 8 位 LED 字幕滚动控制
(17) 串口通信设计	(18) FIFO 存储器设计
(19) 8 位 RISC CPU 设计	(20) 七人表决器设计
(21) FIR 滤波器	(22) OV9712 输出信号模拟实验
(23) IIR 滤波器	(24) SPI 总线接口的 verilog 的实现
(25) DFT Cordic 运算	(26) 异步 FIFO
(27) 小波变换	(28) 2ASK 调制设计
(29) 卷积运算	(30) 4x4 键盘扫描设计
(31) 基于 CORDIC 算法的可调相 DDS 设计	(32) 时钟发生器设计
(33) CRC 即循环冗余校验码	(34) 键盘消抖
(35) 看门狗电路设计	(36) 格雷码计数器
(37) 基于 SDRAM 读写的接口设计	(38) 电梯控制模块设计
(39) 微波炉计时器设计	(40) 空调温度控制器设计
(41) 基于 FLASH 的数据读写设计	(42) “梁祝”乐曲演奏设计
(43) 开 3 次方电路设计	(44) 带符号位小数乘法的设计
(45) 随机数产生器	(46) 卷积码

(47) 双向移位寄存器	(48) 带符号位小数的加法设计
(49) 分频设计	(50) 全自动洗衣机的设计
(51) CRC 校验码生成器	(52) 约翰逊计数器
(53) 抢答器	(54) 电话计费器
(55) 乒乓操作	(56) 奇偶校验
(57) 独热码设计	(58) 8b/10b 编码
(59) 序列检测器的设计	(60) RS232
(61) 曼彻斯特编码设计	(62) PCM30 基群帧同步电路
(63) 施密特触发器	(64) 位置式 PID 控制器设计
(65) 汽车尾灯控制系统设计	(66) 8B/10B 编码器的设计
(67) 出租车计价器	(68) FPGA 实现串口通信
(69) 函数发生器	(70) OV7670 摄像头图像采集和处理
(71) 正负脉宽数控调制信号发生器	(72) CMOS 摄像头驱动设计
(73) 2FSK 调制设计	(74) 摄像头驱动显示设计
(75) 8B/10B 编码器的设计	(76) 基于 FPGA 的摄像处理与显示功能展示
(77) 格雷码计数器	(78) 步进电机驱动设计
(79) 基于目标跟踪的三阶 kalman 滤波器设计	(80) AES 加密算法编码设计
(81) 位置式 PID 控制器设计	(82) 低功耗时序电路的设计
(83) FPGA 实现浮点数减法	(84) 基于 ov7670 摄像头的像素采集器
(85) 基于 iic 协议的数码管显示	

案例一

汽车尾灯控制系统设计

Robei LLC

1. 实验目的

要求掌握汽车尾灯的工作原理，掌握由硬件语言控制 LED 灯的亮灭,并根据原理设计阶乘模块以及设计 test_bench，最后在 Robei 可视化仿真软件进行功能实现和仿真验证。

2.实验原理

（1）汽车尾部左右两侧各有 3 只尾灯，用作汽车行驶状态的方向指示标志。
（2）当汽车正常向前行驶时，6 只尾灯全部熄灭。
（3）当汽车要向左或向右转弯时，相应侧的 3 只尾灯依次由左至右闪亮。每个灯亮 400ns，每个周期为 1200ns，另一侧的 3 只灯不亮。
（4）紧急刹车时，6 只尾灯全部闪亮，闪动频率为 10kHz。

3.实验内容

3.1 light 模型设计

1) 新建一个模型命名为 light，类型为 module，同时具备 5 输入 2 输出。每个引脚的属性和名称参照下图 1 进行对应的修改。

Name	Inout	DataType	Datasize	Function
clock	input	wire	1	clock
turnl	input	wire	1	turn left
turnr	input	wire	1	turn right
ordinary	input	wire	1	ordinary
brake	input	wire	1	brake
lightr	output	reg	3	right light
lightl	output	reg	3	left light

图 1.light 引脚的属性

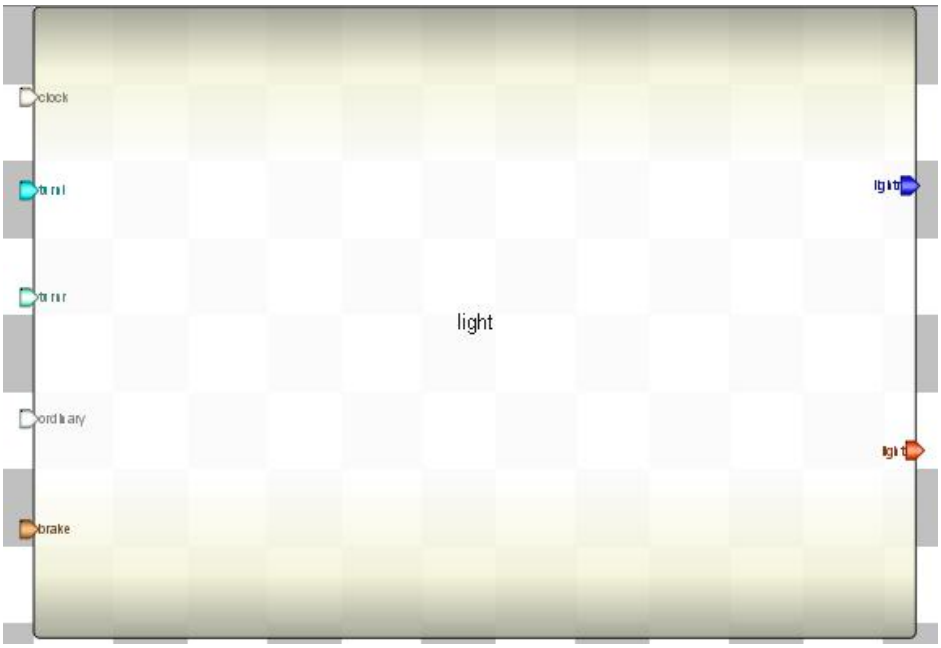


图 2.light 界面图

2) 添加代码。点击模型下方的 Code 添加代码。

代码：

```
/*********************************/
integer temp0=0;
integer a_temp=0;
integer b_temp=0;
reg f1=0,f2=0,i=0;
//分频模块
always@(posedge clock)
begin
if(a_temp==9)
begin
f1=~f1;
a_temp<=0;
end
else
a_temp<=a_temp+1;
end
always@(posedge clock)
begin
if(b_temp==99)
begin f2=~f2;
b_temp<=0;
end
else
b_temp<=b_temp+1;
end
```

```
always@(posedge f2)
begin
if(temp0<2)
temp0<=temp0+1;
else
temp0<=0;
end
always@(posedge f2)
begin
i<=~i;
end
always@(posedge f1)
begin
if(turnl==1&&turnr==0)
begin
case(temp0)
0:begin lightl<=3'b100;lightr<=3'b000;end
1:begin lightl<=3'b010;lightr<=3'b000;end
2:begin lightl<=3'b001;lightr<=3'b000;end
endcase
end
else if(brake==1)
begin
case(i)
0:
begin
lightl<=3'b111;
lightr<=3'b111;
end
1:
begin
lightl<=3'b000;
lightr<=3'b000;
end
endcase
end
else if(turnr==1&&turnl==0)
begin
case(temp0)
0:begin lightr<=3'b100;lightl<=3'b000;end
1:begin lightr<=3'b010;lightl<=3'b000;end
2:begin lightr<=3'b001;lightl<=3'b000;end
endcase
end
```

```
else
begin
lightl<=3'b000;
lightr<=3'b000;
end
end
/*******************************/
```

3) 保存模型到一个文件夹(文件夹路径不能有空格和中文)中，运行并检查有无错误输出。

3.2 light_test 测试文件的设计

1) 新建一个 5 输入 2 输出的 light_test 测试文件,记得将 Module Type 设置为 “testbench”，各个引脚配置如图 3 所示。

Name	Inout	DataType	Datasize	Function
clock	input	reg	1	clock
turnl	input	reg	1	turn left
turnr	input	reg	1	turn right
ordinary	input	reg	1	ordinary
brake	input	reg	1	brake
lightr	output	wire	3	right light
lightl	output	wire	3	left light

图3.light_test 测试文件引脚的属性

2) 另存为测试文件。将测试文件保存到上面创建的模型所在的文件夹下。

3) 加入模型。 在 Toolbox 工具箱的 Current 栏里，会出现模型，单击该模型并在 light _test 上添加，并连接引脚，如下图 4 所示：

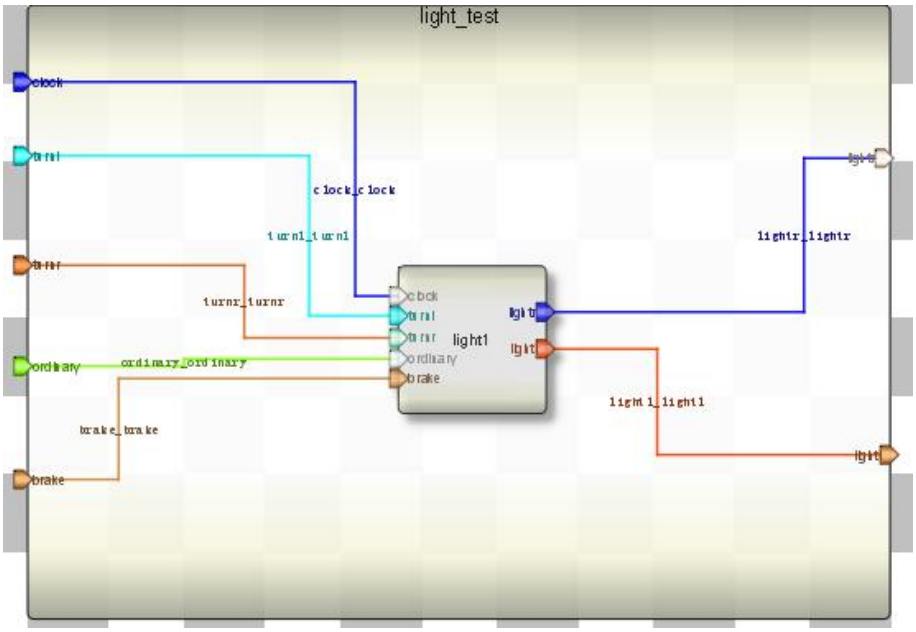


图 4. light_test 界面图

4) 输入激励。点击测试模块下方的“Code”，输入激励算法。激励代码在结束

的时候要用$finish 结束。

测试代码：

```
/**********************************/
initial begin
clock=0;
turnl=0;
turnr=0;
ordinary=0;
brake=0;
#1000
turnl=1;
#3500
turnl=0;
#100
turnr=1;
#4000
turnr=0;
#1000
ordinary=1;
#4000
ordinary=0;
#200
brake=1;
#4000
$finish;
end
always begin
#1 clock=~clock;
end
/**********************************/
```

5) 执行仿真并查看波形。查看输出信息。检查没有错误之后查看波形。点击右侧 Workspace 中的信号，进行添加并查看分析仿真结果。如图5所示：

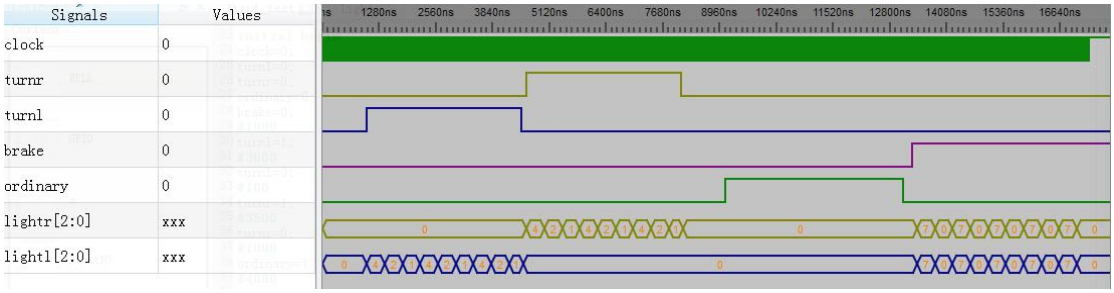


图 5.light_test 的仿真波形

案例二

自动售饮料机

1.实验目的

了解自动售货机的工作流程以及各个工作状态，以及其 test_bench,最后在 Robei 可视化仿真软件经行功能实现和仿真验证。

2.实验原理

自动售货机的信号定义：clk： 时钟输入； reset： 为系统复位信号； half_dollar：代表投入 5 角硬币； one_dollar： 代表投入 1 元硬币； half_out： 表示找零信号； dispense： 表示机器售出一瓶饮料； collect： 该信号用于提示投币者取走饮料。

当 reset=0 时，售货机处于工作状态，此时连续往售货机中投硬币（可以是 5 毛也可以是一元），投入最后一枚硬币时，如果之前投入的银币总和为 2.5 元则可以取走一瓶饮料，如果少于 2.5 元则继续投币，如果为 3 元则则显示可以取出一瓶饮料而且找零显示信号为高电平。

投入硬币的总额	自动售饮料机给出的信号
<2.5 元	继续投币
=2.5 元	可以取出一瓶饮料
=3 元	可以取出一瓶饮料，并且找零

3.实验内容

3. 1、sell 模块的设计

1）、 新建一个模型命名为 sell，类型为 module，同时具备 4 输入 3 输出，每个引脚的属性和名称参照下图 1 经行对应的修改。

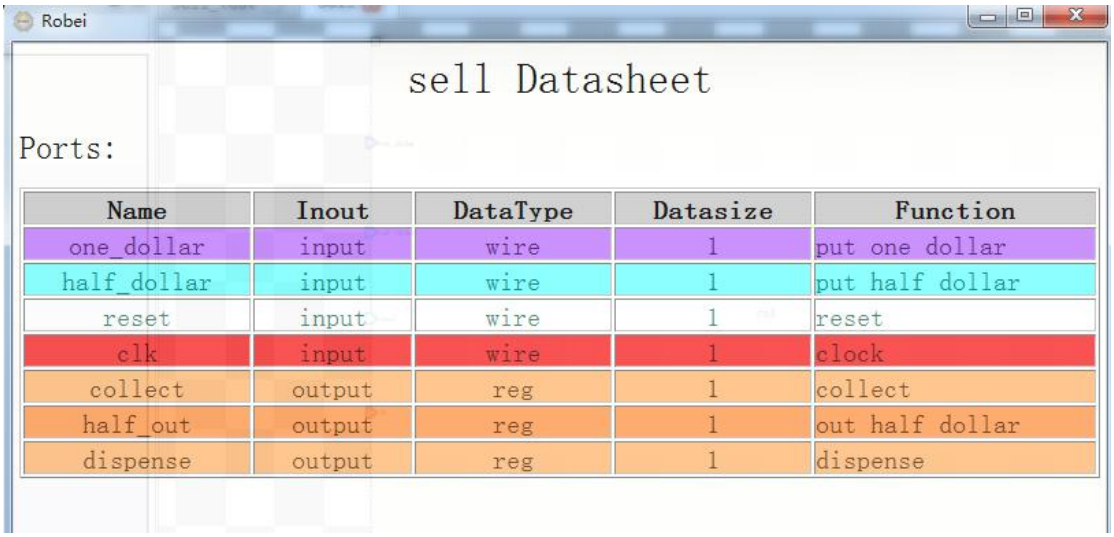


图 1 sell 引脚的属性

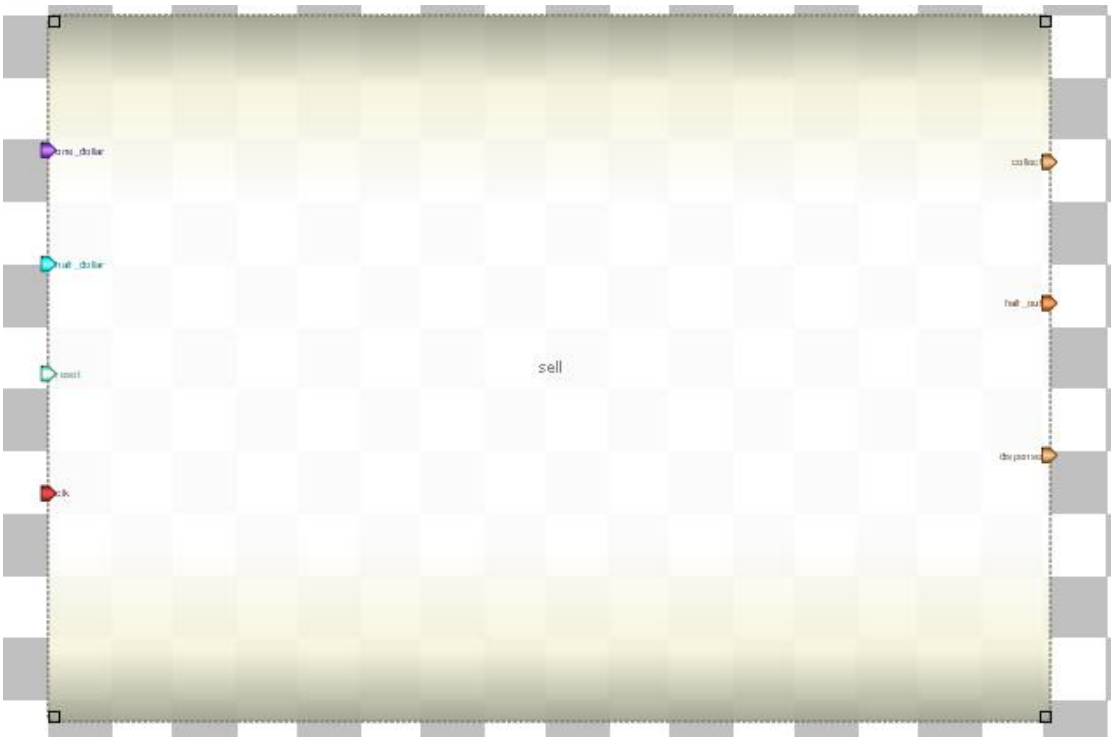


图 2sell 界面图

2）、添加代码。点击模型下方的 Code 添加代码。

代码：

```
/*********************************/
parameter idle=0,one=2,half=1,two=3,three=4;
reg[2:0] D;
always @(posedge clk)
begin
if(reset)
begin
dispense=0; collect=0;
```

```
half_out=0; D=idle;
end
case(D)
idle:
if(half_dollar) D=half;
else if(one_dollar)
D=one;
half:
if(half_dollar) D=one;
else if(one_dollar)
D=two;
one:
if(half_dollar) D=two;
else if(one_dollar)
D=three;
two:
if(half_dollar) D=three;
else if(one_dollar)
begin
dispense=1;
collect=1;
D=idle;
end
three:
if(half_dollar)
begin
dispense=1;
collect=1;
D=idle;
end
else if(one_dollar)
begin
dispense=1;
collect=1;
half_out=1;
D=idle;
end
endcase
end
/*********************************/
```

3）、保存模型到一个文件夹(文件夹路径不能有空格和中文)中，运行并检查有无错误输出。

3.2sell_test 测试文件的设计

1) 新建一个 4 输入 3 输出的 sell_test 测试文件,记得将 Module Type 设置为 “testbench”，各个引脚配置如图 3 所示。

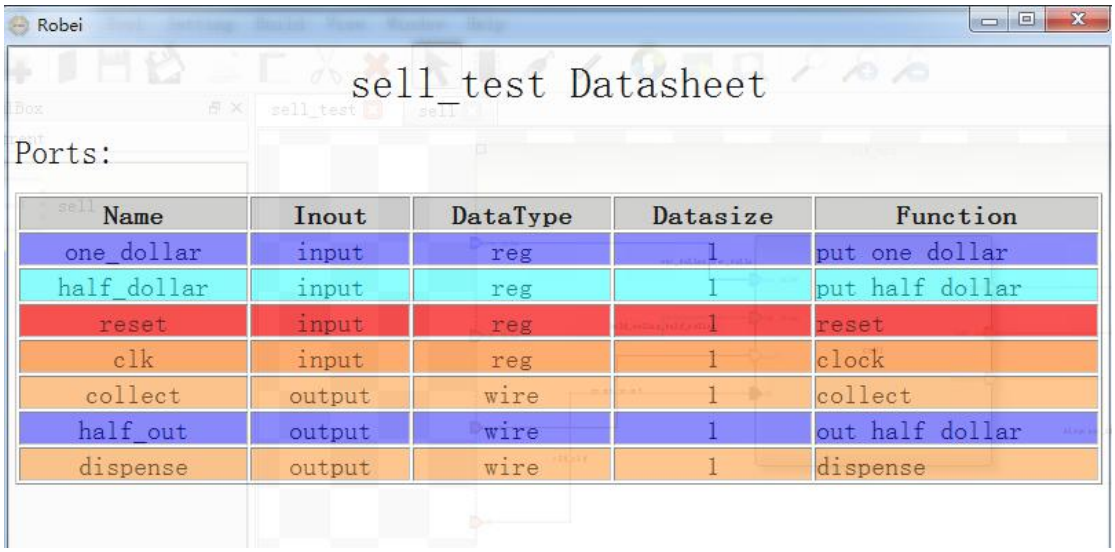


图 3sell_test 引脚的属性

4) 、另存为测试文件。将测试文件保存到上面创建的模型所在的文件夹下。加入模型。 在 Toolbox 工具箱的 Current 栏里，会出现模型，单击该模型并在 sell _test 上添加，并连接引脚，如下图 4 所示：

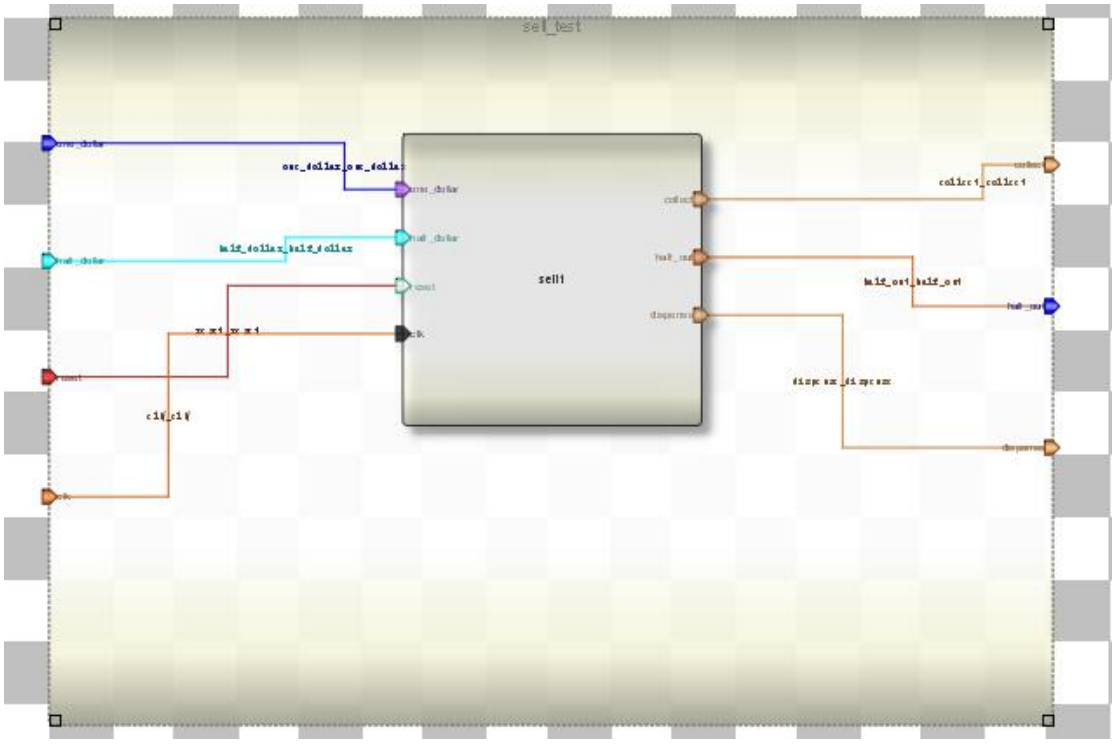


图 4sell_test 工作界面

4) 输入激励。点击测试模块下方的“Code”，输入激励算法。激励代码在结束的时候要用$finish 结束。

测试代码：

```
/**********************************/
initial begin
one_dollar=0;
half_dollar=0;
```

```
reset=1;
clk=0;
#100 reset=0;
repeat(2)@(posedge clk);
#2 one_dollar=1;
repeat(1)@(posedge clk);
#2 one_dollar=0;
repeat(2)@(posedge clk);
#2 one_dollar=1;
repeat(1)@(posedge clk);
#2 one_dollar=0;
repeat(2)@(posedge clk);
#2 one_dollar=1;
repeat(1)@(posedge clk);
#2 one_dollar=0;
#20 reset=1;
#100 reset=0;
repeat(2)@(posedge clk);
#2 one_dollar=1;
repeat(1)@(posedge clk);
#2 one_dollar=0;
repeat(2)@(posedge clk);
#2 one_dollar=1;
repeat(1)@(posedge clk);
#2 one_dollar=0;
repeat(2)@(posedge clk);
#2 half_dollar=1;
repeat(1)@(posedge clk);
#2 half_dollar=0;
#20 reset=1;
#5 $finish;
end
always #10 clk=~clk;
/***********************************/
```

5) 执行仿真并查看波形。查看输出信息。检查没有错误之后查看波形。点击右侧 Workspace 中的信号，进行添加并查看分析仿真结果。如图 6 所示：

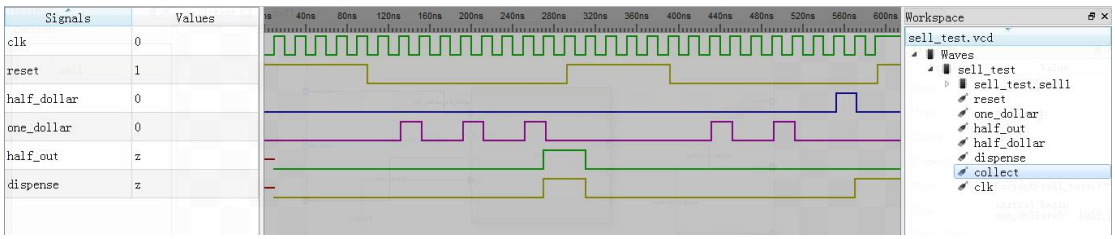


图 6　sell_test 仿真波形

教材

7 天搞定 FPGA--Robei 与 Xilinx 实战

目 录