## 实验课形式

1. 学生根据实验课安排时间准时参加实验
2. 学生需尽量提前做实验并进入实验室做实验、检查实验，学生可自己安排任意做哪一个实验。
3. 每次开放实验时间，实验老师都在，各教学班任课老师和助教分别共进入实验室16学时。
4. 每次实验就尽量现场进行检查，好评价实验过程的质量。

## 实验考核方式

1. 2-3个学生一组，每组领一块实验板，原则上同一个老师的学生组成一组。
2. 学生成绩由现场检查和实验报告两部分构成，都是以一组的成绩体现。学生以小组为单位，做好考核实验后主动向教师申请实验检查，教师从组员中随机选一名来陈述实验情况，包括设计思路、实验演示、结果分析等，他的表现将作为整组的现场检查成绩。实验报告一组提交一份。
3. 一共有4次实验和一次项目，由于实验课时间有限，各个小组应该尽早进行实验检查，不要拖到学期后面检查，否则时间不够。
4. 实验打分各个班都是统一标准的，因此任何一名在实验课当值的老师都可以检查学生的实验，也就是老师会检查到不是自己班的学生的实验并打分。

## 实验内容

实验内容分为基础类、组合逻辑电路设计、时序逻辑电路设计、有限状态机设计、FPGA综合应用设计五类。每一类包括了不同难度的实验项目，\*\*\*~\*\*\*\*\*难度逐级递增，高难度的实验需在掌握低难度实验的基础上完成。实验项目的不同难度对应了不同的最高可得成绩:

|  |  |
| --- | --- |
| 难度 | 最高可得成绩(百分制) |
| \*\*\* | 60 |
| \*\*\*\* | 80 |
| \*\*\*\*\* | 100 |

如:难度为\*\*\*的实验项目,最高可得的成绩为60分,以此类推。每一类实验项目中任选一个作为考核项目，考核项目需提交实验报告并通过现场检查。

* **基础类**

实验一 基本门电路——Vivado开发环境及Basys3开发板的熟悉（\*\*\*）

1. 使用Verilog HDL语言的数据流描述方法设计一个（数据宽度可在1~32之间变化的）2输入与门andgate（或门、非门、与非门等基本门）；
2. 利用仿真来验证所设计门电路的逻辑（学会看仿真波形图），填写门电路真值表，并将该与门封装成可变数据宽度参数的IP核。

实验二 2选1多路选择器——IP核的封装与调用（\*\*\*\*）

1. 将实验一中的与门（演示过程）、或门、非门、与非门等基本门电路封装成IP核；
2. 调用封装的IP核，用Block Design设计一个一位的2选1多路选择器；
3. 用verilog HDL语言编写一个一位的2选1多路选择器；
4. 分别用RTL分析2）、3）中的电路设计（学会看RTL电路图）；
5. 编写仿真文件，填写出2选1多路选择器真值表，验证所设计电路的逻辑；
6. 综合、实现、生成bit流，下载到Basys3开发板进行验证；

实验三 多路选择器（\*\*\*\*\*）

设计一个三位4选1多路选择器电路，分别用Block Design和verilog HDL语言编写代码两种方式实现，通过仿真、看RTL电路图、下载到板子验证其正确性，并比较不同方法实现的异同（逻辑资源、RTL电路图、仿真波形等方面）。

* **组合逻辑电路设计**

实验四 译码器（\*\*\*\*）

1. 根据3-8译码器真值表，使用Verilog HDL语言实现74LS138译码器decode38，下载到板子上进行验证。其中Y0~Y7分别接板上的LD0~LD7，G1接SW15，G2A接SW14，G2B接SW13，C、B、A分别接SW2，SW1和SW0；
2. 将1）中的3-8译码器封装成IP核，调用3-8译码器的IP核，用Block Design设计一个4-16译码器，并通过写仿真文件、看RTL电路图、下载到板子验证其正确性；

实验五 七段数码管显示（\*\*\*\*）

1. 将拨码开关SW3~SW0以十六进制数显示在1个七段数码管上（了解七段数码管显示原理，可提供参考代码）；
2. 设计一个3位的4选1多路选择器，并利用拨码开关SW15~SW14控制选择将拨码开关SW11~SW9、SW8~SW6、SW5~SW3、SW2~SW0以八进制数显示在1个七段数码管上；
3. 将拨码开关SW11~SW9、SW8~SW6、SW5~SW3、SW2~SW0分别以八进制数显示在4个七段数码管上；（学习数码管分时复用显示原理，可提供参考代码）
4. 封装一个显示模块IP核，该模块能将32位的二进制数以十六进制形式显示到8个七段数码管上，并编写顶层模块调用该IP核，下载至开发板验证其正确性。

实验六 加减法器（\*\*\*\*\*）

1. 设计一个1位全加器，然后由32个1位全加器串联组成32位加法器（画出电路原理图），并通过写仿真文件、看RTL电路图、下载至开发板验证其正确性；
2. 用全加器来构建全减器（画出电路原理图），实现一个32位加减法器，并通过写仿真文件、看RTL电路图、下载至开发板验证其正确性；
3. 编写顶层模块将加减法器模块和7段数码管显示模块连接起来，将SW15~SW12作为第一操作数低4位，SW11~SW8作为第二操作数低4位，SW0=0（1）时做加（减）法，添加选择控制SW1~SW2，能选择将操作数1或操作数2或结果显示在7段数码管上。
4. 在3）中的加减法器中增加进位标志和溢出标志，设计实现一个带进位标志和溢出标志的32位加减法器，将SW15~SW12作为第一操作数（可显示在左起第一个7段数码管上），SW11~SW8作为第二操作数（可显示在左起第二个7段数码管上），将结果显示在7段数码管上，进位和溢出标志通过LED灯显示；
5. 将所设计的加减法器与系统自带的“+\-”进行比较。

实验七 加法器扩展训练（\*\*\*\*\*）

完成超前进位加法器、流水线加法器的设计与实现，当输入的值每个周期都在变化并要求每个周期产生其和值的情况，与实验六中的加法器进行比较。

实验八 乘/除法器的设计与实现（\*\*\*\*\*）

设计实现一个32位的乘法器和除法器，并进行仿真验证和下板验证。

* **时序逻辑电路设计**

实验九 寄存器（\*\*\*）

1. 设计一个带有（同步或异步）复位的D触发器 ddf1。编写仿真文件观察仿真时序波形图，回答异步清零与同步清零的区别，并下载到板验证。信号和板上器件的关系是：输出q-LD0,输出qn-LD1,输入d-SW0,时钟clk-BTNC,复位reset-BTND。
2. 给1）中的D触发器dff1增加一个load信号就成为了一个一位寄存器reg\_1，当load为1时，那么在下一个时钟上升沿，输入值d就被存储到q中，编写代码实现reg\_1，并下载到板验证。
3. 把N个1位寄存器组合起来就构成了一个N位寄存器。只需要将1位寄存器的输入端和输出端定义为N位的数组。编写代码实现一个可变宽度的N位寄存器（默认为8），编写顶层模块实例化一个8位寄存器，并下载到板验证。

实验十 移位寄存器（\*\*\*\*）

1. 实现一个8位的通用移位寄存器，可以加载并行数据，通过两位控制信号ctrl可以选择控制将其内容向左移位、向右移位或保持原有状态。利用此移位寄存器实现并转串（先加载并行输入，然后移位），和串转并（首先移位，然后进行并行输出），并下载到板验证。
2. 实现一个8位的桶形移位寄存器，具有8位数据输入和8位数据输出，以及指定如何移动数据的控制输入，指定移位方向、移位类型（循环、算术还是逻辑移位）及移动的位数（0~7）。信号和板上器件的关系是：原始数据d[7:0]--SW[7:0]，输出q[7:0]--ld[7:0]，s[2:0]--SW[15:13]表示移动的位数，t[1:0]--SW[12:11]表示移动类型（00逻辑右移，01-算术右移，1X-左移）。

实验十一 寄存器堆的设计(\*\*\*\*)

实现一个具有8个8位寄存器的“两读一写”寄存器堆，该寄存器堆具有读写和寄存功能。写地址信号w\_addr指定了数据存储位置，读取地址信号r\_addr1、r\_addr2指定数据读取位置。信号和板上器件的关系是：输出数据r\_data1-LD7~LD0，r\_data2-LD15~LD8，写入数据w\_data[5:0]-SW5~SW0(w\_data[7:6]在顶层传入1或0)，时钟clk-BTNC，写控制wen-SW15,写寄存器选择w\_addr-SW14~SW12，读寄存器选r\_addr1-SW11~SW9，r\_addr2- SW8~SW6。

实验十二 时钟分频设计（\*\*\*）

实现一个分频器 clock\_div，输入的是系统提供的100MHz频率，输出的是1Hz频率，占空比是1:2。下载到板子上进行验证，信号与板上器件的关系是：输入内部时钟clk管脚W5，输出clkout-LD0。要求画出分频时序波形原理图。

实验十三 递增递减计数器（\*\*\*\*）

设计一个计数器能实现每一秒在0~7范围内进行递增或递减，up\_down选择控制递增或递减，计数器当前值显示到数码管上。

实验十四 存储器实验（\*\*\*\*\*）

1. 利用BASYS3片内存储器单元实现单端口RAM设计（带异步读和同步读两种模式），在时钟（clk）上升沿，采集地址（addr）、输入数据（data\_in）、执行相关控制信息。当写使能（we）有效，则执行写操作，否则执行读取操作。同步与异步设计仅针对读操作：对于异步RAM而言，读操作为异步，即地址信号有效时，控制器直接读取RAM阵列；对于同步RAM而言，地址信号在时钟上升沿被采集。并保存在寄存器中，然后使用该地址信号读取RAM阵列。
2. 实现双端口（同步与异步）RAM设计，相对于单端口RAM而言，双端口RAM存在两个存取端口，并且可独立进行读写操作，具有自己的地址（addr\_a、addr\_b）、数据输入（din\_a、din\_b）/输出端口（dout\_a、dout\_b）。
3. 实现FIFO设计，FIFO由存储单元队列或阵列构成，和RAM不同的是FIFO没有地址，第一个被写入队列的数据也是第一个从队列中读出的数据。FIFO可以在输入输出速率不匹配时，作为临时存储单元；可用于不同时钟域中间的同步；输入数据路径和输出数据路径之间数据宽度不匹配时，可用于数据宽度调整电路。

* **有限状态机设计**

实验十五 摩尔状态机序列检测器(\*\*\*\*\*)

1. 设计“1101”序列检测的状态转换图;
2. 设计一个 8 位并转串输出模块 par2ser。该器件有 8 位输入 d[7:0]，1 位输出 q，另有一个 clk 端，一个 set 端。set端上升沿将 8位输入锁存到逻辑右移移位寄存器中。
3. 调用并转串输出模块，使用Verilog HDL语言的行为描述方式实现一个摩尔状态机，能检测一个8位的二进制数据中是否存在“1101”序列，如果检测到该序列则指定的LED灯亮；
4. 综合、实现、生成bit流，下载到Basys3开发板进行验证；

实验十六 米里状态机序列检测器(\*\*\*\*\*)

设计实现一个米里Mealy状态机，能检测一个8位的二进制数据中是否存在“1011”序列。

实验十七 交通灯信号控制器设计（\*\*\*\*\*）

实现一个十字路口（南北和东西方向）交通灯信号控制器。该十字路口的南北和东西方向都有红、黄、绿三种颜色的信号灯，分别用LED灯代替。请根据交通规则，给出交通灯的状态表和状态转换图，延迟时间以秒计算。

* **FPGA综合应用设计**
* 项目一 流水灯设计（\*\*\*\*）

实现Led灯从左向右循环依次每隔1秒点亮。

* 项目二 数字钟设计（\*\*\*\*）

实现一个六十进制数字时钟，秒到 60 则归零重加，同时让分加1，分加到60归零重加，并让小时加 1，小时加到24归零重加。要求用数码管1,0显示秒值，数码管 2,3 显示分值，数码管 4,5 显示小时（可以十六进制形式显示在led灯上）。

* 项目三 密码锁设计（\*\*\*\*\*）

设计一个密码锁电路，可以通过拨码开关设置初始密码，通过按钮来输入密码，密码位数至少为4位，判断输入密码与设置的密码是否一致。

* 项目四 电梯控制器设计（\*\*\*\*\*）

设计一个多楼层的电梯控制器系统，并能在开发板上模拟电梯运行状态。可以利用按键作为呼叫按键，数码管显示电梯运行时电梯所在楼层，led灯显示楼层叫梯状态。

* 项目五 屏幕保护应用设计（\*\*\*\*\*）

利用FPGA开发板和VGA显示器设计制作一个屏幕保护应用，包括图像存储模块、VGA显示控制器、图像移动轨迹的设计。

* 项目六 FPGA综合应用设计之扩展训练（\*\*\*\*\*）

基于BASYS3的俄罗斯方块、贪吃蛇游戏等复杂数字系统开发设计