

## 实验二 FPGA 开发板与 Vivado 入门

### 一、 实验目的

- 学习 NEXYS A7 开发板的使用
- 学习基于 Vivado 软件的 FPGA 开发流程
- 使用 verilog 语言和 xdc 约束文件实现 FPGA 编程
- 初步了解组合逻辑和时序逻辑

### 二、 实验环境

Vivado 2019.1 开发工具

NEXYS A7 开发板

### 三、 实验内容

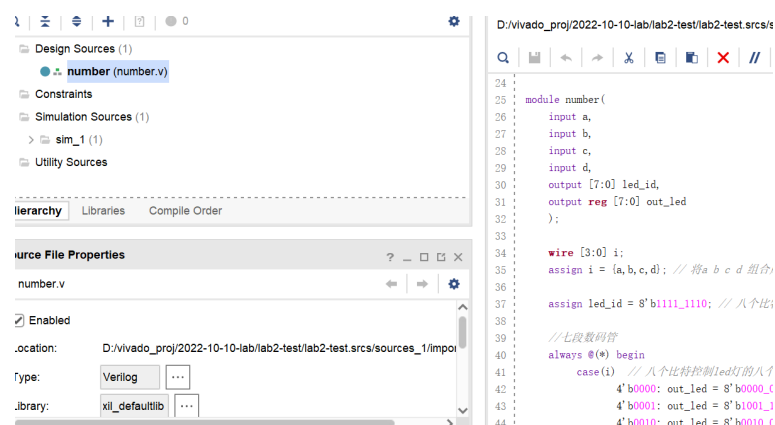
#### 1. NEXYS A7 入门实验

下载群文件, 打开“DIGILENT 官方指导手册\_你的 Nexys 4DDR 第一个入门实验.pdf”, 完成第一章的内容, 初步认识 NEXYS。接着完成第二章的内容, 最后可以实现打开开关 SW0 和 SW1, 对应的 LED 灯会亮。这个指导手册每一步的操作都介绍地很详细。

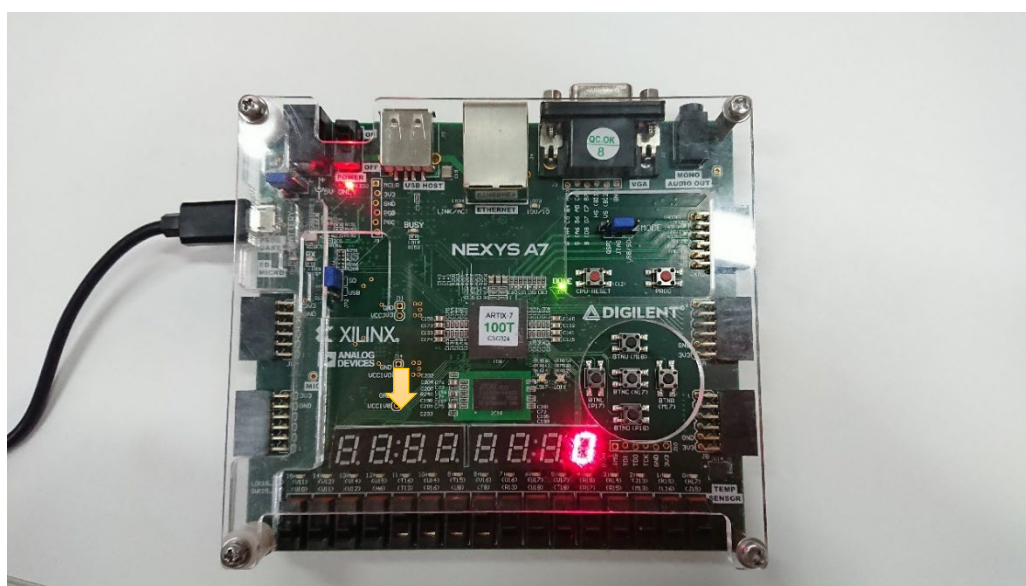
#### 2. NEXYS A7 第二个实验

接下来, 我们将会实现控制 SW0、SW1、SW2、SW3 四个开关, 将它们的组合当作一个二进制数, 使得上方的七段数码管会显示对应的十六进制数。以下是实验步骤:

- (1) 新建一个 project: 方法和第一个实验的方法一样。
- (2) 选择 Add or create design sources 添加 sources,  
选择 Add files, 选择刚刚下载的文件中的 numbers.v 文件添加,  
添加成功后 Finish, 双击点开应该能看见 verilog 代码和注释。



- (3) 选择 Add or create constraints 添加约束条件 xdc, 然后和上一步一样, 选择 numbers.xdc 文件添加。
- (4) 综合, 生成 bitstream 配置文件, 下载到 NEXYS。(与上个实验一样)
- (5) 最后 program 比特流。最右边的数字会亮起, 拨动最右边的四个开关, 会显示相应的十六进制数字



### 3. 作业题

更改第二个实验的代码, 使上一张图标有黄色箭头的数码管亮, 使它对应的开关为 SW8、SW9、SW10、SW11, 更改这些开关改变数码管的数字。完成作业的同学需要现场演示。(具体 I/O 引脚见英文手册 Page18) (注意: 每次更改 verilog 代码后都要重新进行 Synthesis 综合和 Generate Bitstream 生成比特流, 更改 xdc 约束文件后只需要重新生成比特流即可)

## 4. 附加题

**题目要求：**设计程序和约束文件，使拨动右边的 8 个开关，右边的四个数码管可以显示 8 位二进制对应的十进制数，这里的二进制要求是补码（最高为 1 的话会输出负数）。

**相关提示：**要使四个数码管显示不同的数，需要使用时序逻辑。在 input 中加入时钟，并在 xdc 文件中添加代码，将时钟绑定板子自带的 100Mhz 时钟，时钟的引脚接口可在英文用户手册中查到。

具体方法可参考时钟分频。

## 四、 实验注意事项与提交方法

1. 本实验使用 NEXYS A7 开发板，需要同学们分成**三人组或四人组**。
2. 本次实验需要大家提前掌握 verilog 的基础语法知识，能够看懂基本的 verilog 语言（至少看懂本实验文件提供的.v 文件），可参考群文件中的 ppt 学习。
3. 本次实验**不需要交实验报告**，大家完成作业题之后找助教进行现场演示即可。先做完的同学可以思考附加题，在下课前做出附加题的同学可以加分。
4. 严禁抄袭。
5. 爱护实验器材。**借用和归还开发板需遵守规章手续进行登记。**
6. Vivado 软件和 Hyper-V 会有冲突。如果报错 Unable to launch local hw\_server executable, 请从命令行关掉 Hyper-V 并重启电脑。命令为：bcdedit /set hypervisorlaunchtype off

## 附: [HDLbits](#) verilog 练习

从本周开始，后面的实验都需要用 verilog 实现相关设计，为了方便同学们快速熟悉 verilog 语言，大家可以在 HDLbits 网站快速入门，学习基本的 verilog 语法、编程思路、仿真方法等。如果没有足够的时间完成该网站的所有练习，在此给大家整理该网站中重点需要学习的内容。

首先需要了解最基础的 verilog 语法，以及组合逻辑、时序逻辑的基本实现方法，

**Verilog 语法：**建议把 **Verilog Language** 所有题练习一遍。

**组合逻辑与时序逻辑：**可参考实验相关文件中的 **verilog-hdlbits.md** 选择性练习，有兴趣的可以自行练习其他题目。

另外，verilog 编程的仿真调试和 c 语言不一样，必须手动编写 testbench 仿真，观察电路图才能进行程序调试，所以有必要学习观察波形图，编写 **testbench 仿真** 的方法：建议把 HDLbits 的最后一章 **Verification** 做完，并尝试在 Vivado 软件中进行仿真。