《数字电路与数字系统实验》实验报告

**实验7: 存储器**

**姓名:** 尹浚宇

**学号:** 161130118

**班级:** 2018-2019第一学期数字电路与数字系统实验2班

**邮箱:** [908664035@qq.com](mailto:908664035@qq.com)

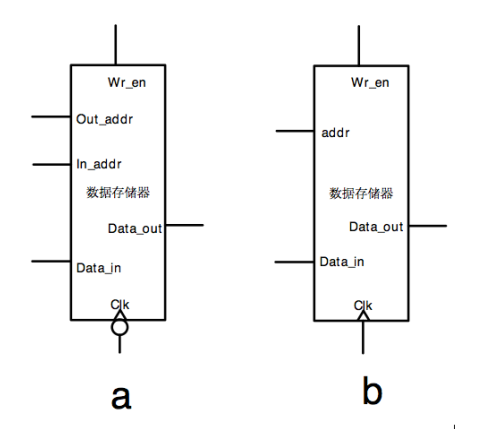
**实验时间:** 2018-10-30

1. **实验目的**
2. 了解FPGA芯片上存储器的存储特性.
3. 学会分析存储器的工作时序和结构.
4. 学会如何设计存储器.
5. **实验原理**

存储器(Memory)是电⼦设备中的记忆器件, ⽤来存放程序和数据. 电⼦设备中全部信息, 包括输⼊的原始数据、程序、中间运⾏结果和最终运⾏结果都保存在存储器中.

1. 存储器结构

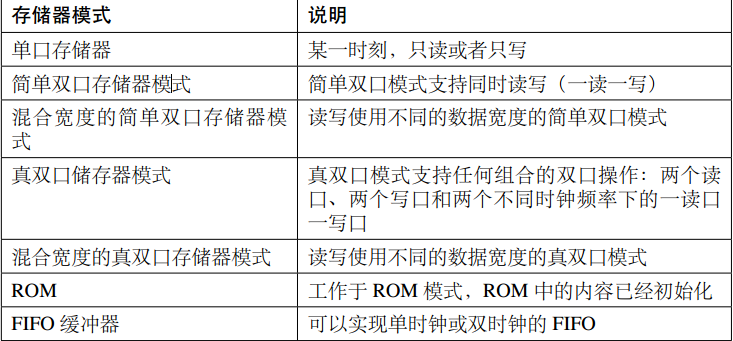
存储器是⼀组存储单元, ⽤于在计算机中存储⼆进制的数据, 如图所示.



存储器的端⼜可以分为输⼊端、输出端和控制端. 输⼊端又包括: 读/写地址端、数据输⼊端等; 输出端⼀般指的是输出数据端; 控制端⼜包括时钟端和读/写控制端. 存储器的⼯作过程如下:

写数据：在时钟(clk)有效沿(上升或下降沿), 如果写使能(Wr\_en，也可以没有使能端)有效, 则读取输⼊总线(Data\_in)上的数据, 将其存储到输⼊地址线(In\_addr)所指的存储单元中.  
读数据：存储器的输出可以受时钟和使能端的控制，也可以不受时钟和使  
能端的控制. 如果输出受时钟的控制，则在时钟有效沿，将输出地址所指示的单元中的数据，输出到输出总线上(Data\_out); 如果不受时钟的控制,则只要输出地址有效，就⽴即将此地址所指的单元中的数据送到输出总线上.

FPGA存储器的⼯作模式有很多, 如：真双口RAM、简单双口RAM、单口 RAM、 ROM或者FIFO缓存等. 常见的模式请参照下表.



1. 存储器实现

Cyclone V系列FPGA内部含有两种嵌⼊式存储器块:

10Kb的M10K存储器块——这是专⽤存储器资源块. M10K 存储器块是理想的⼤存储器阵列, 并提供⼤量独⽴端口.

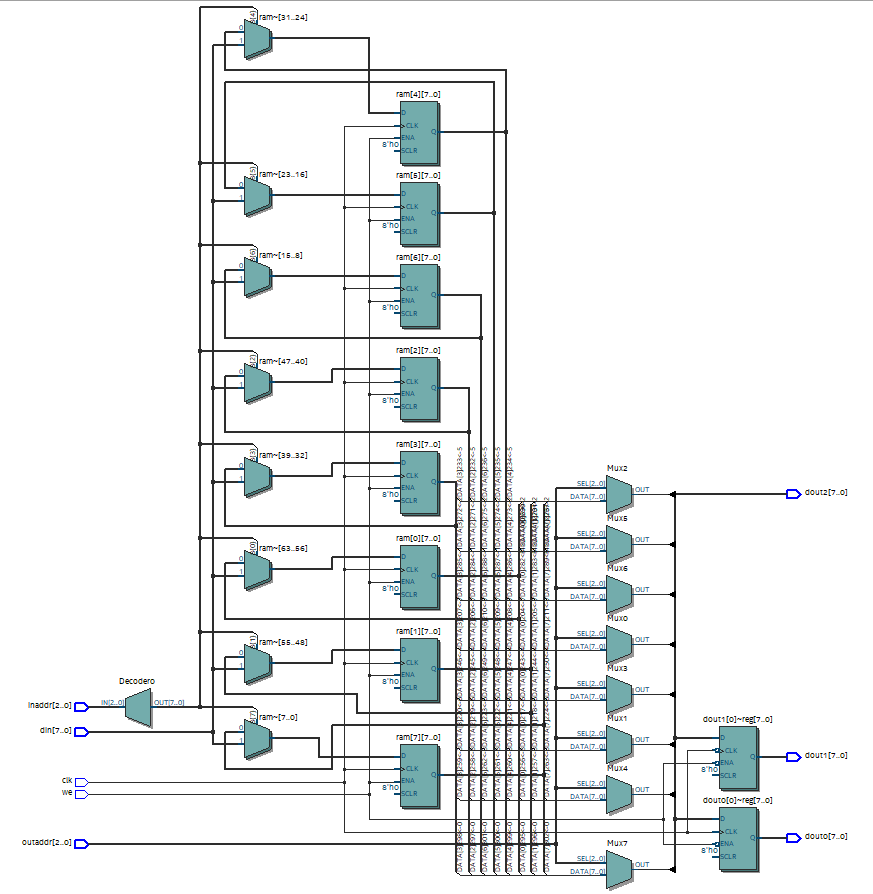
64位存储器逻辑阵列(MLABs)——是⼀种嵌⼊式存储器阵列是由双⽤途逻辑阵列块配置⽽来的. MLAB 是理想的宽⽽浅的存储阵列. MLAB是经过优化的可以⽤于实现数字信号处理(DSP)应⽤中的移位寄存器、宽浅FIFO缓存和滤波延迟线.每个MLAB都由10个⾃适应逻辑块(ALM)组成.

Quartus会根据⽤户存储器设计的速度与⼤⼩, 来⾃动划分硬件实现时存

储器模块的数量与配置. 例如, 为提供设计性能, Quartus可能将可以由1块RAM实现的存储器设计扩展为由多块RAM来实现.

1. 存储器实例分析

RTL图如下:



解释如下:

dout2的值始终和ram[outaddr]相同.

dout0的值在写使能无效且上升沿时和ram[outaddr]同步, 之后一直保持不变直到下一次满足条件,

dout1的值在写使能无效且下降沿时和ram[outaddr]同步, 之后一直保持不变直到下一次满足条件,

1. **实验环境/器材**

系统环境是window10, 硬件环境是DE10-Standard开发板, 软件环境是Quartus.

1. **程序代码**

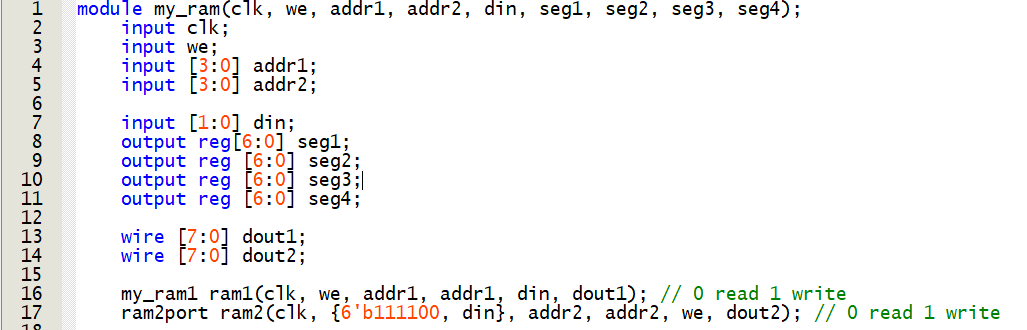
存储器设计思路如下:

该存储器总共分为三个模块, 在顶层模块my\_ram里对其他两个模块进行了实例化. 其他两个模块一个为自己设计的存储器, 一个为IP核自动生成的存储器.

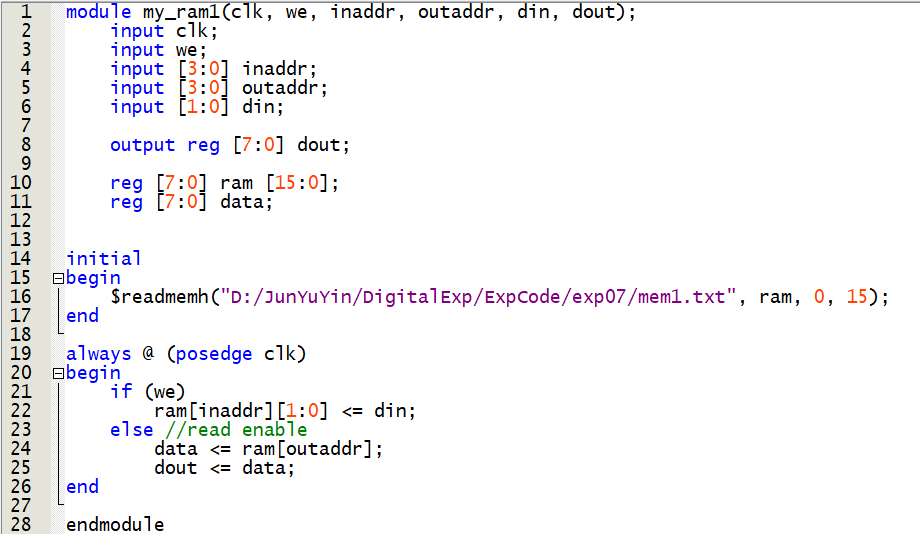
对于自己设计的存储器模块, 为了实现输出有缓存的功能, 加入了一个中间变量data, 对于IP核生成的存储器, 则按照教程配置, 未加改动. 最后在顶层模块里还有一个将结果映射到数码管的功能.

部分代码如下(完整代码见压缩包):

顶层模块实例化部分代码:



my\_ram1代码:



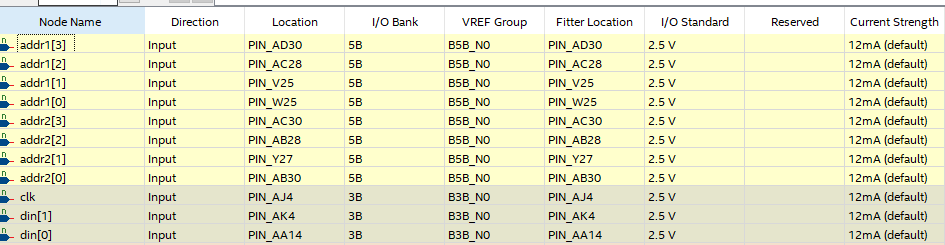
1. **实验步骤**

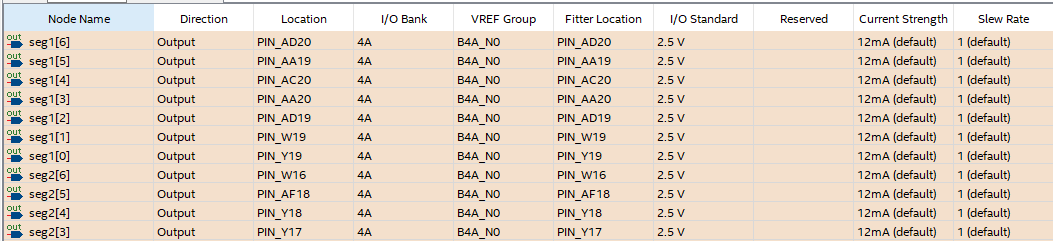
首先设计了一个可读写, 输出有缓存的存储器, 使用.txt文件进行存储器的初始化, 而后利用IP核生成了一个双口存储器, 使用.mif文件进行了初始化, 最后在顶层模块中将它们综合在一起(具体见四), 通过编译后又进行了仿真模拟(见六), 仿真模拟通过后分配了引脚(见下方), 最后将生成的二进制烧写文件导入开发板进行硬件验证, 并通过了助教的查验.

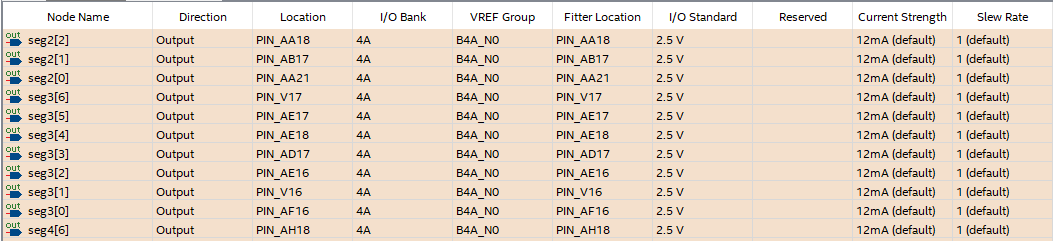
引脚分配思路如下:

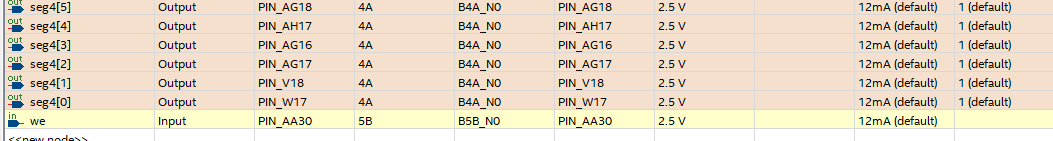
时钟端和数据输入用按钮实现, 使能端及两个存储器的地址都用开关实现, 输出用数码管实现.

引脚分配图片如下”









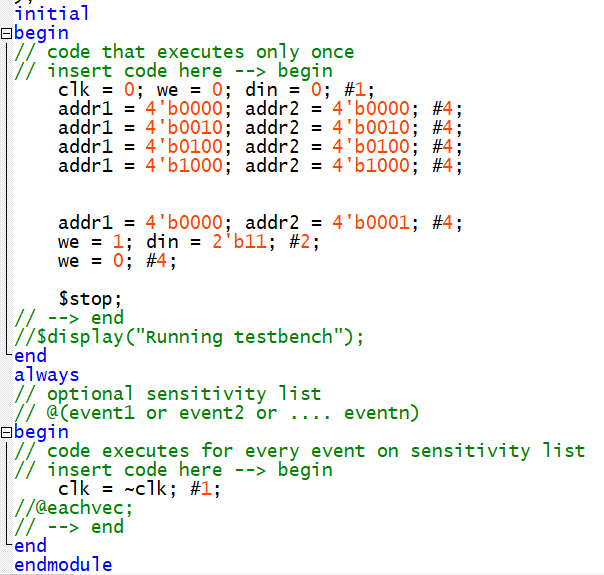
1. **测试方法**

采用仿真验证和硬件验证的测试方法, 首先编写了如下的测试代码(见下方)进行仿真验证(仿真波形图见七), 而后进行了硬件验证(见七的描述).

仿真模拟思路如下:

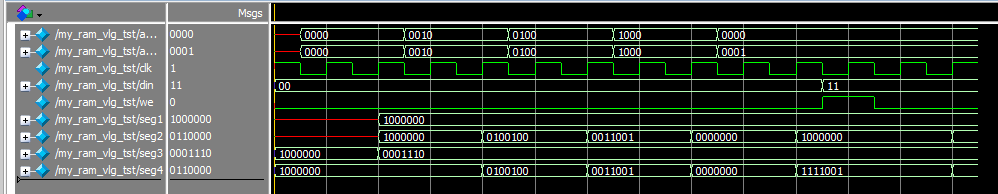
首先分别读取两个存储器每个地址的数据, 检测是否初始化成功和读取功能是否正确. 而后修改存储器某个地址的值后再读取, 检测写入功能是否正确.

部分仿真代码如下:



1. **实验结果**

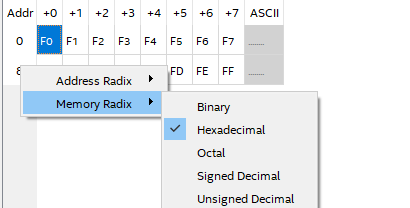
仿真波形图如下:



可以看到仿真结果基本符合预期.

硬件验证结果难以用图片文字说明, 且验收已通过, 可以将压缩包内的.sof文件写入开发板内进行验证, 此处不再赘述.

1. **实验中遇到的问题及解决方案**
2. 在设置.mif文件的初值时, 发现单元格里输入形如0xXY形式的数据会产生错误, 后来发现.mif文件单元格里数据的进制需要自己调整, 如下图所示.



1. IP核生成的存储器在输入输出长度不一致时会有一点问题, 后来用IP核生成了一个输入输出长度一致的存储器, 通过在输入的高位补特定数字达到预期效果. 具体如下图所示.



1. **实验得到的启示**
2. 可以用满足特定格式的.txt文件或者.mif文件初始化存储器.
3. Quartus提供了很多实用的IP核, 利用这些IP核可以简化我们的设计.
4. 利用非阻塞赋值语句和一些中间变量可以做到输出缓冲功能.
5. **意见和建议**
6. 讲义中可以提供.mif文件设置时怎么切换不同进制输入的方法.
7. 希望讲义更新后可以在群里发公告通知, 避免课上现场改代码.