# 计算机组成原理课程实验

# 实验报告

09015101 程茜雅 09015120 胡 斌

## **实验一 寄存器组成实验**

#### 实验目的

1. 熟悉D触发器的功能及使用方法。
2. 掌握寄存器文件的逻辑组成及使用方法。

#### 实验内容

1. 掌握Quartus II的使用方法，能够进行数字电路的设计及仿真。
2. 验证Quartus II所提供D触发器的功能及使用方法。
3. 设计具有1个读端口、1个写端口的寄存器文件，并进行存取操作仿真/验证。

#### 三、实验原理及设计方案

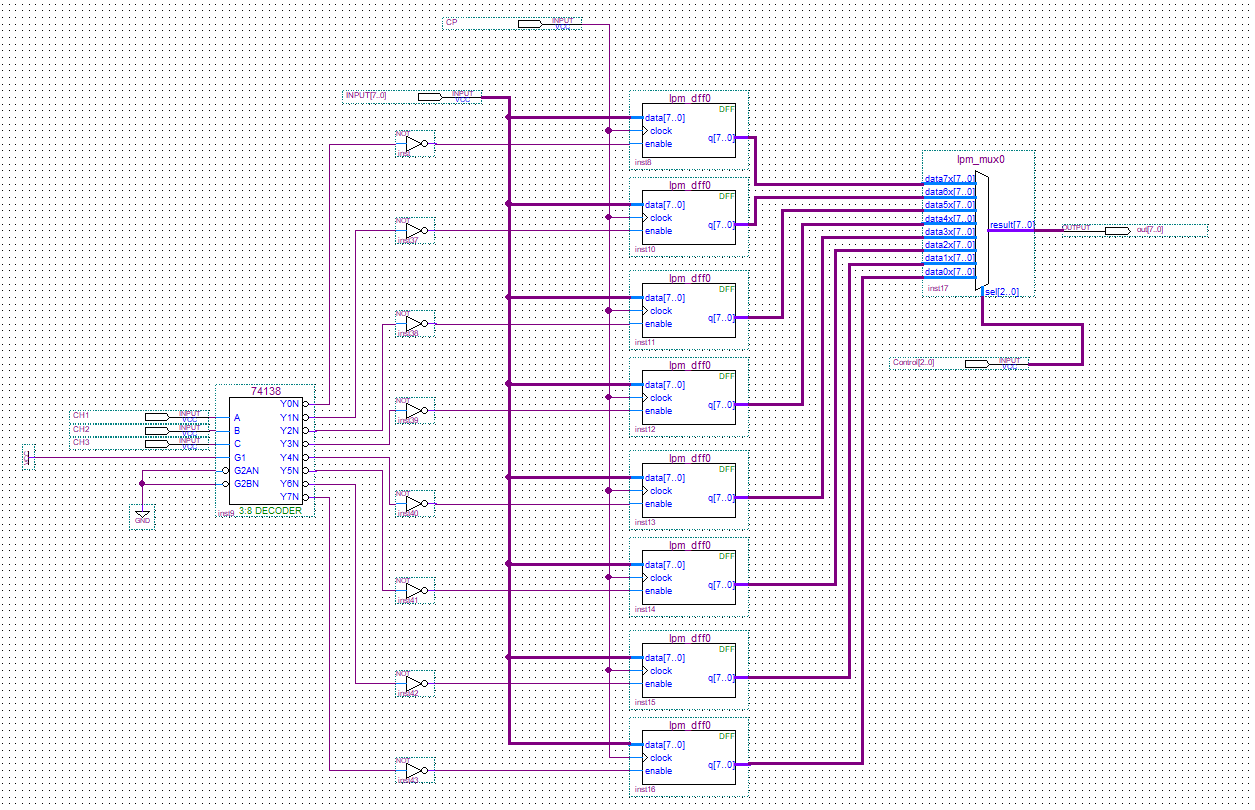
本实验的寄存器文件要求包含8个8位寄存器，具有1个读端口、1个写端口。因此，操作地址均为3位，数据均为8位。所以我们利用Quartus II提供的8位D触发器lpm\_dff（用8个）、74138译码器和数据选择器lpm\_mux()实现了一个寄存器组。

由于要求读写端口分离，因此，读操作的相关引脚有地址Control[2..0]、数据输出out[7..0]，写操作的相关引脚有地址CH[2..0]、数据输入INPUT [7..0].

寄存器文件通过写地址CH [2..0]信号来实现触发器的写入控制，通过读地Control[2..0]信号来控制触发器的数据输出选择。其连接电路原理如图所示。通过写地址译码器使能对应的寄存器，就可以实现写地址选择。

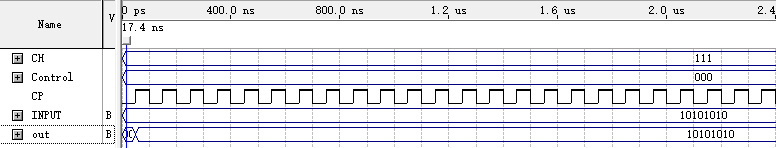


电路图如下：



#### 实验结果

功能仿真波形图如下：



#### **五、实验小结**

通过这次试验，我们不仅熟悉了D触发器的功能及使用方法，而且还掌握了寄存器文件的逻辑组成及使用方法。深感Quartus II功能之强大，我们可以根据实验需要自定义所用的器件，十分方便。

## 实验二 运算器组成实验

#### 实验目的

（1）熟悉加/减法器的功能及使用方法。

（2）掌握算术逻辑部件(ALU)的功能及其逻辑组成。

（3）加深对运算器工作原理的理解。

#### 二、实验内容

（1）验证Quartus II所提供加/减法器的功能及使用方法。

（2）设计具有加法、减法、逻辑与、逻辑非4种功能的ALU，并进行功能仿真/验证。

#### 三、实验原理及设计方案

运算器是计算机硬件进行数据加工的重要部件，算术逻辑部件ALU是其核心，还包括存放操作数和运算结果的寄存器/锁存器、存放运算结果状态的触发器等器件。

本实验所要求的ALU数据宽度为8位，具有4种算术及逻辑运算功能，其功能选择

引脚记SEL[1]～SEL[0]。该ALU实现的具体功能如表所示，其中，算术运算可以实现

有符号数、无符号数的运算功能。

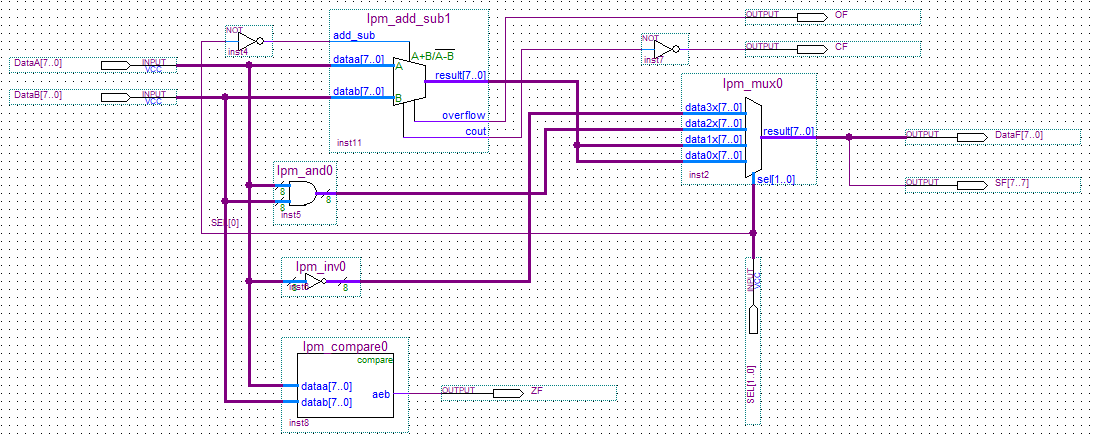
|  |  |  |
| --- | --- | --- |
| SEL[1] | SEL[0] | 实现功能 |
| 0 | 0 | 加法 |
| 0 | 1 | 减法 |
| 1 | 0 | 逻辑与 |
| 1 | 1 | 逻辑非 |

用lpm\_add\_sub模块实现ALU的加法、减法功能，用lpm\_and、lpm\_inv模块实现逻辑与、逻辑非功能。通过门电路、数据选择器、编码/译码器等实现ALU引脚与内部模块引脚之间的关联。

本实验所要求ALU的逻辑结构如下图所示，控制形成电路负责产生各功能模块的控制信号，本实验中与门、非门无需控制是特例；状态形成电路负责产生4个状态标志，分别是零标志ZF（用lpm\_compare判断数据DataA和DataB是否相等）、进位/借位标志CF、溢出标志OF、结果符号标志SF。

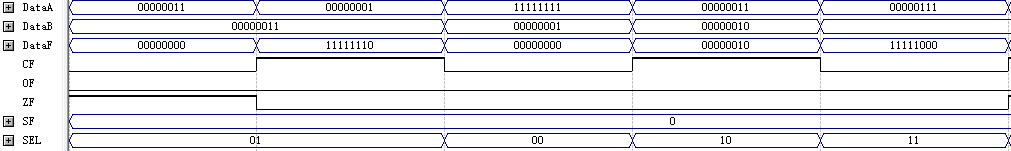


电路图如下：



#### 实验结果

功能仿真波形图如下：



#### 实验心得

通过这次试验，我们掌握了算术逻辑部件(ALU)的功能及其逻辑组成，熟悉了加/减法器的功能及使用方法，也接触到了一些很方便很实用的器件，如lpm\_compare（）可以比较大小。

## 实验三 存储器组成实验

#### 实验目的

（1）熟悉半导体存储器的存取方法。

（2）掌握存储器的扩展方法。

（3）掌握存储器与总线的连接方法。

#### 二、实验内容

（1）验证Quartus II所提供半导体存储器的功能及使用方法。

（2）设计一个读/写端口分离的256×8bit的存储器，地址空间中前一半只读、后一半可读可写，并进行存取操作仿真/验证。

（3）将上述存储器连接到地址/数据复用的总线上，并进行存、取操作仿真/验证。

#### 三、实验原理及设计方案

1、半导体存储器的操作

本次实验我们用到的存储器是lpm\_dq，其特性是8位字长、256个存储单元、读/写端口分离，其I/O引脚如图所示。其中，data[7..0]、q[7..0]分别为数据输入、数据输出引脚，wren为写操作（写使能）信号、高电平有效，clock为同步操作的时钟信号。



lpm\_dq引脚示例

对lpm\_dq的操作有读、写两种。写操作（wren=1）时，数据在clock上升沿时锁存并写入到存储单元中。读操作（wren=0）时，数据在clock上升沿后一段时间输出到引脚（功能仿真时延迟可忽略、实时仿真时延迟为1个时钟左右）

2、存储器扩展的设计与验证

本实验的存储器扩展要求ROM与RAM一起进行字扩展，两者地址空间相同，均为128个单元，地址引脚为address[6..0]，数据引脚为q[7..0]。

设扩展后存储器的地址引脚为A[7..0]、数据输出引脚为Q[7..0]，则A[7]用于选中ROM或RAM，即控制ROM或RAM的clock时钟信号产生、哪个q[7..0]连接到Q[7..0]。

3、存储器与总线连接的设计与验证

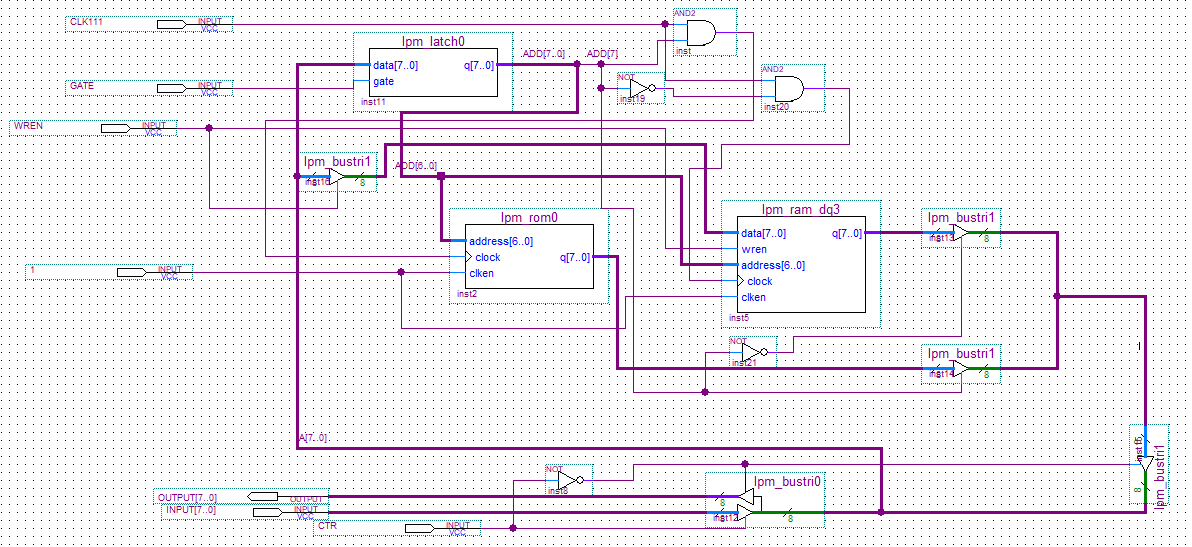
本实验中存储器lpm\_dq连接的总线要求是地址/数据复用总线，即存储器的地址引脚、数据引脚连接到同一组总线上。而存储器lpm\_dq是读/写端口分离的，即数据输入引脚、数据输出引脚是不同的引脚。因此，本实验中存储器lpm\_dq的address[7..0]、data[7..0]、q[7..0]引脚应同时连接到同一组总线bus[7..0]上。

由于只允许一个设备向总线发送信息，允许多个设备从总线接收信息。因此，本实验中，存储器lpm\_dq的地址信号、数据输入信号、数据输出信号须分时与总线交互。

存储器lpm\_dq与地址/数据复用总线连接时，地址信号、数据输入信号的分时写入可通过增设地址锁存器（lpm\_latch）实现，输入信号、输出信号的分时传送可通过增设三态门（lpm\_bustri）实现；同样地，输入部件、存储器的分时传送可通过增设三态门实现。其连接电路原理如图所示。



电路图如下：

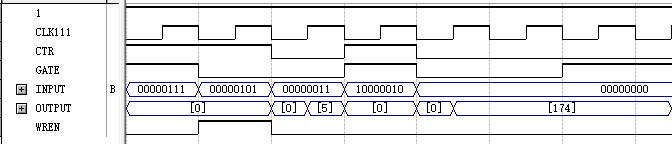


在设计lpm\_rom时，添加了hex文件，如下所示：



#### 四、实验结果

功能仿真波形图如下：



#### 五、实验小结

通过这次实验，我们熟悉了半导体存储器的存取方法，掌握了存储器的扩展方法以及存储器和总线的连接方法。不仅如此，我们还实现了数据/地址分时复用的总线，对存储器有了更加深入的了解。

## 实验四 CPU数据通路实验

#### 实验目的

（1）掌握CPU数据通路的逻辑组成。

（2）了解指令功能的实现过程及其控制方法。

#### 二、实验内容

（1）设计一个单总线结构的CPU数据通路，部件包括4种功能的8位ALU、4×8位的寄存器文件、256×8位的RAM、8位计数器各一个。

（2）给出相关部件控制信号，分别实现取数、加法、条件转移指令的功能。

#### 三、实验原理及设计方案

1、CPU数据通路的设计

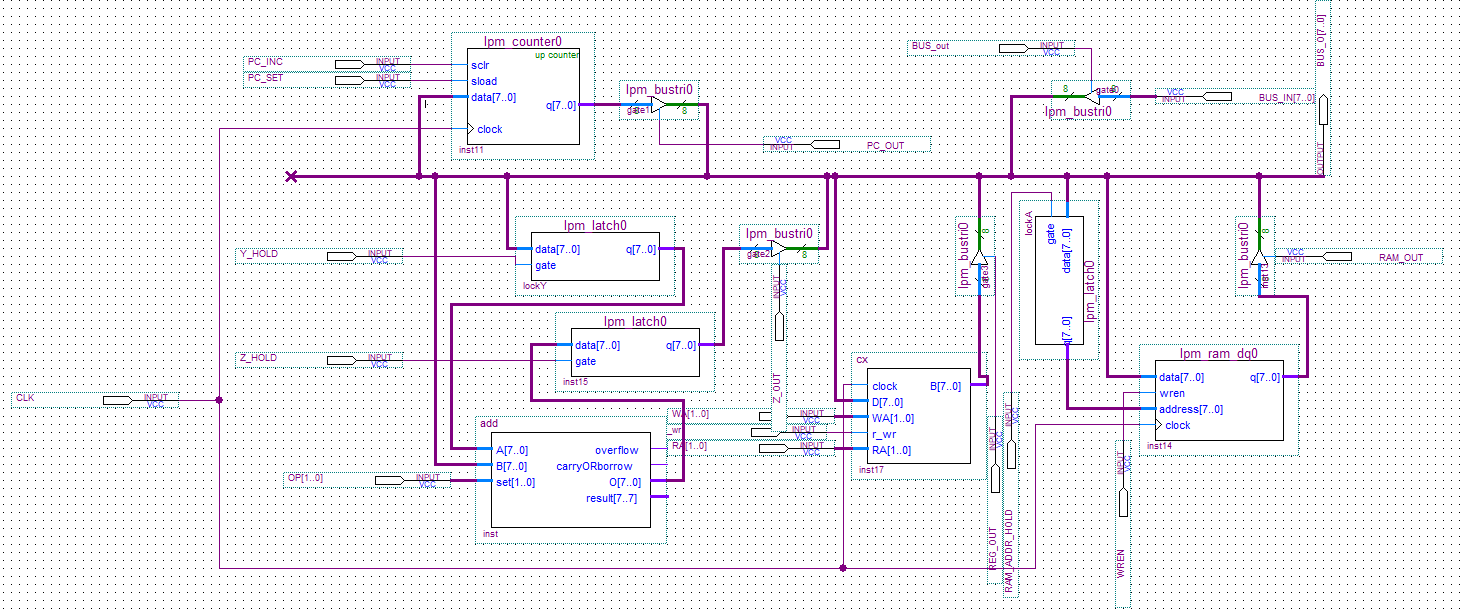
本实验要求的是设计单总线结构数据通路，数据宽度为8位，包括ALU、寄存器文件、RAM、计数器4个功能部件。各部件的功能分别是，ALU支持加法、减法、逻辑与、逻辑非4种操作（同实验二），寄存器文件用作通用寄存器组（记为REGs），由4个具有独立读端口、写端口的寄存器组成（同实验一）；RAM用作Cache，地址空间为8位、读/写端口分离（同实验三）；计数器用作程序计数器（记为PC），具有置数、计数功能。

实验的数据通路可以采用如图6所示的方案，其中，三态门（记为TSL）是依据总线操作特性（同时只有一个部件能发送数据）而设置的；锁存器是为解决部件的多个端口在单总线上数据接收冲突而设置的；REG\_S为状态寄存器，存放关系运算所需的标志位（如ZF）；其它部件与总线的数据输入、数据输出接口记为BUS\_I、BUS\_O，输入、输出分开便于观察实验结果。



另外，我们还需增设锁存器Y，解决ALU 的两个入端间的输入冲突；需增设锁存器Z，解决ALU的入端-出端间的环路冲突。同理，需增设锁存器A，解决RAM的地址-数据引脚间的输入冲突。

其中，ALU的操作控制信号线(2根)为op[2]，REGs的写地址信号线(2根)、读地址信号线(2根)分别为WA[2]、RA[2]，REGs、RAM的写操作控制信号线分别为r\_wr、m\_wr，计数器的置数控制信号线、时钟信号线分别为ldn、cp。

电路图如下

2、指令功能的实现

本实验要求的是实现取数、加法、条件转移指令的功能，各指令功能如下：

取数LD -- RD←[(RS)]

加法ADD -- RD←(RD)+(RS)及RD←(RD)+[(RS)]

条件转移JZ -- ZF=1时，PC←(PC)+disp；ZF=0时，PC←(PC)+1

其中，RD、RS表示寄存器编号，(Rx)、[(Rx)]表示寄存器寻址、寄存器间接寻址方式，(PC)+disp表示相对寻址方式（disp为偏移量）。可见，ADD指令的源操作数支持两种寻址方式，JZ指令的PC值依据状态寄存器中ZF位的状态形成。

基于数据通路，每个指令的功能可表示为一个微操作步序列，每个微操作步由一个或几个可同时实现的微操作组成。只要按序实现某微操作步序列中各微操作的功能，即可实现该指令的功能。微操作步序列如下：

（注：对于控制信号，不加标记代表置高有效，加下划线代表置低无效。 ）

取数 LD ：RD←[(RS)]

1） RS→RA，REG\_OUT，RAM\_ADDR\_HOLD，REG\_OUT

2） RD→WA，r\_wr，RAM\_OUT

3） RAM\_OUT，RAM\_ADDR\_HOLD，结束

加法 ADD：RD←(RD)+[(RS)]

1） RD→RA，REG\_OUT，Y\_HOLD，REG\_OUT

2） RS→RA，REG\_OUT，RAM\_ADDR\_HOLD，REG\_OUT

3） RAM\_OUT，ADD→OP，Z\_HOLD，RAM\_OUT，RAM\_ADDR\_HOLD，Y\_HOLD

4） RD→WA，r\_wr，Z\_OUT

5） Z\_OUT，结束

加法 ADD：RD←(RD)+ (RS)

1） RD→RA，REG\_OUT，Y\_HOLD，REG\_OUT

2） RS→RA，REG\_OUT，ADD→OP，Z\_HOLD，REG\_OUT，Y\_HOLD

3） RD→WA，r\_wr，Z\_OUT

4） Z\_OUT，结束

条件转移 JZ（ZF=1）： PC←(PC)+disp

1） PC\_OUT，Y\_HOLD，PC\_OUT

2） BUS\_IN，ADD→OP，Z\_HOLD，BUS\_IN，Y\_HOLD

3） Z\_OUT，PC\_SET，Z\_OUT，结束

条件转移 JZ（ZF=0）： PC←(PC)+1

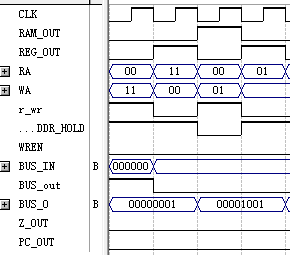
1） PC\_INS

据此，即可设计波形并仿真。

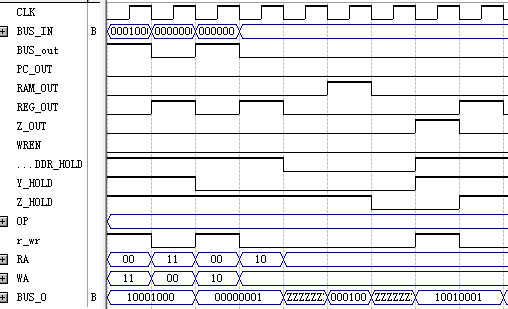
#### 四、实验结果

各指令的仿真波形图如下：

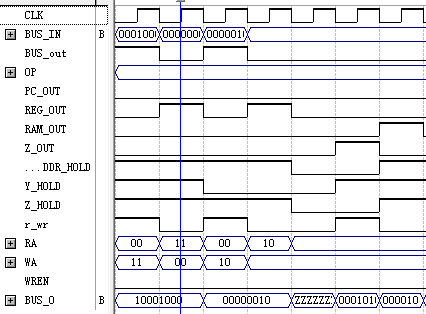
取数 LD ：RD←[(RS)] ：

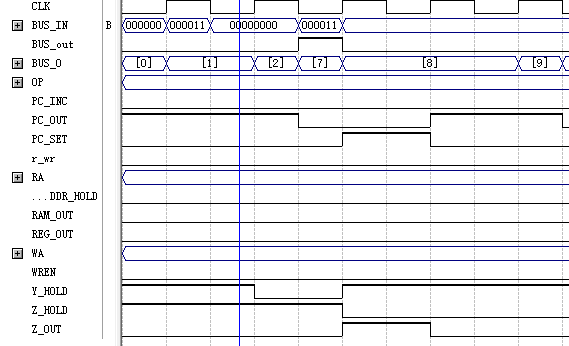


加法 ADD：RD←(RD)+[(RS)] ：



加法 ADD：RD←(RD)+ (RS)：



条件转移JZ（PC←(PC)+1 和 PC←(PC)+disp）：

#### 五、实验小结

通过这次实验，我们不仅加深了对之前做过的三个实验的理解，将ALU运算器，寄存器，ram存储器组合应用；还掌握了CPU数据通路的逻辑组成，了解了指令功能的实现过程及其控制方法，练习了微操作序列的写法。