

计算机组成原理

实验报告

学号：57119101 姓名：王晨阳

学号：57119102 姓名：庄严

二0二0 年十二月

**实验一 寄存器组的设计**

**一、实验内容**

* 测试D触发器的功能。
* 设计具有1个读端口、1个写端口的4×8位寄存器组，并验证设计正确性。

**二、电路设计与实现**

实验需要设计一个寄存器组，本实验要求的寄存器组包含4 个8 位寄存器，具有1 个读端口、1 个写端口。实验小组通过仔细分析需求，设计出如下电路原理简图



**图1.1 电路原理简图**

实验通过译码器选择将数据输入到哪一个触发器，通过选择器选择输出哪一个触发器存储的数据。触发器会在时钟脉冲上升沿且译码器使其enable的情况下写入数据。实验同时设置了清零的操作，将触发器内数据清零。

该寄存器组共有7个引脚，其中输入引脚6个，输出引脚1个，具体如下表所示

**表1.1 引脚**

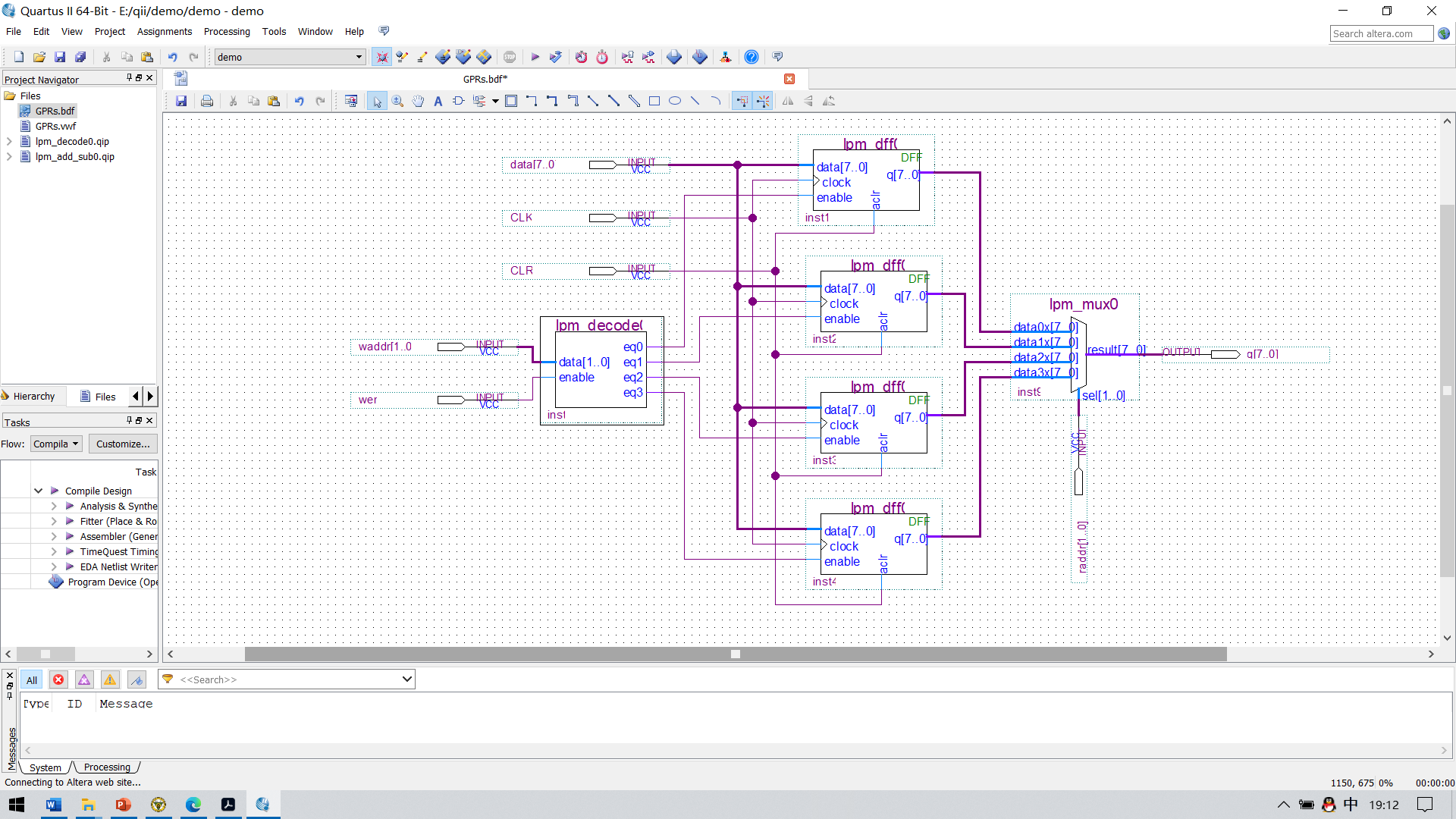
|  |  |  |
| --- | --- | --- |
| **名称** | **类型** | **功能** |
| clr | 输入（1位） | 寄存器组清零 |
| clk | 输入（1位） | 时钟信号 |
| data | 输入（8位） | 数据输入 |
| wen | 输入（1位） | 写使能 |
| waddr | 输入（2位） | 写地址 |
| raddr | 输入（2位） | 读地址 |
| q | 输出（8位） | 数据输出 |

器件选择如下表所示

**表1.2 器件**

|  |  |  |
| --- | --- | --- |
| **名称** | **型号** | **使用的引脚** |
| 芯片 | Cyclone III系列P3C16Q240C8芯片 |  |
| 译码器 | lpm\_decode | data[1..0]  enable  eq0  eq1  eq2  eq3 |
| 触发器 | lpm\_dff | data[7..0]  clock  enable  aclr  q[7..0] |
| 选择器 | lpm\_mux | data0x[7..0]  data1x[7..0]  data2x[7..0]  data3x[7..0]  sel[1..0]  result[7..0] |

电路设计如下图所示。其中有译码器1个，选择器1个，触发器4个。



**图1.2 电路图**

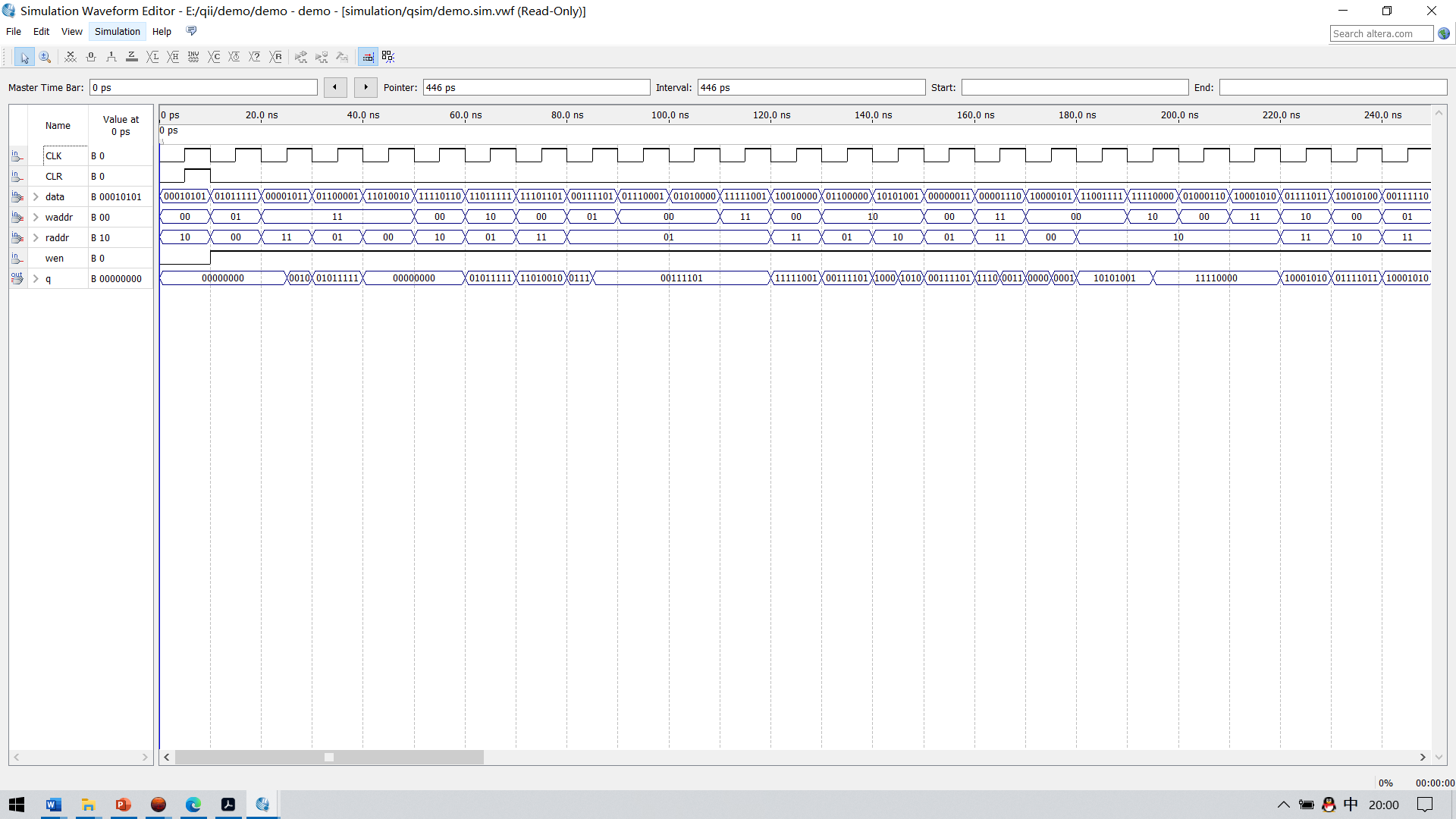
**三、电路正确性验证**

**1、电路仿真**

为了使仿真结果更加可靠，除一般情况外，实验的仿真涵盖了如下特殊情况

* 情况一：清零操作
* 情况二：在寄存器中没有存入数据时输出该寄存器
* 情况三：在输出寄存器数据的同时，该寄存器读入其他数据
* 情况四：寄存器多次读入不同数据后输出
* 情况五：寄存器数据不变的情况下多次输出

我们随机生成了data、waddr、raddr的内容，得到仿真结果如下所示



**图1.3 仿真波形图**

**2、结果分析**

截取实验中的典型时间段分析如下

**表1.3 结果分析**

|  |  |  |
| --- | --- | --- |
| **情况** | **时间/ns** | **具体内容** |
| 情况一 | 0-10 | 期间进行了清零操作，寄存器存储的数据均变为00000000 |
| 情况二 | 10-20 | 清零后，输出00寄存器，结果为00000000 |
| 情况三 | 80-90 | 01寄存器原数据为01011111，在第85ns时钟上升沿读入数据00111101，输出85ns前为01011111，85ns后为00111101 |
| 情况四 | 50-60  70-80  120-130  150-160  170-175 | 在前面4个时间段内，00寄存器分别读入数据11110110、11101101、10010000、00000011，在第170-175ns期间输出为00000011 |
| 情况五 | 80-120  130-140  150-160 | 在80-90ns期间01寄存器读入数据00111101，此后几次输出均为00111101 |

该结果与预期完全一致，成功实现了寄存器组的功能。

**四、实验小结**

实验小组成员通过共同讨论，在没有学习过数字电路的情况下，成功完成了电路的设计与仿真，并在验收时一次性通过。

通过本次实验，小组成员成功掌握了Quartus的基本使用方法，了解了电路设计的流程，巩固了所学习的计算机组成原理知识，对寄存器和寄存器组的工作原理有了更加深刻的认识。小组成员也通过本次实验认识到了合作的重要性，锻炼了勇于面对困难与未知的精神。

**五、教师评语**

教师签字：

日期：

**实验二 ALU的设计**

**一、实验内容**

* 测试加/减法器的功能。
* 设计具有加法、减法、逻辑与、逻辑非4种功能的8位ALU，ALU需产生结果状态标志ZF、CF、OF、SF，并验证设计正确性。

**二、电路设计与实现**

本实验要求ALU的数据宽度为8位，具有4种算术及逻辑运算功能，需产生结果状态标志ZF、CF、OF、SF。ALU的功能如下表所示，其中，A、B为数据入端，F为数据出端，SEL为功能选择（操作控制）信号。可见，SEL为2位（记为SEL[1..0]），F的位数与A及B相同。

**表2.1 ALU功能**

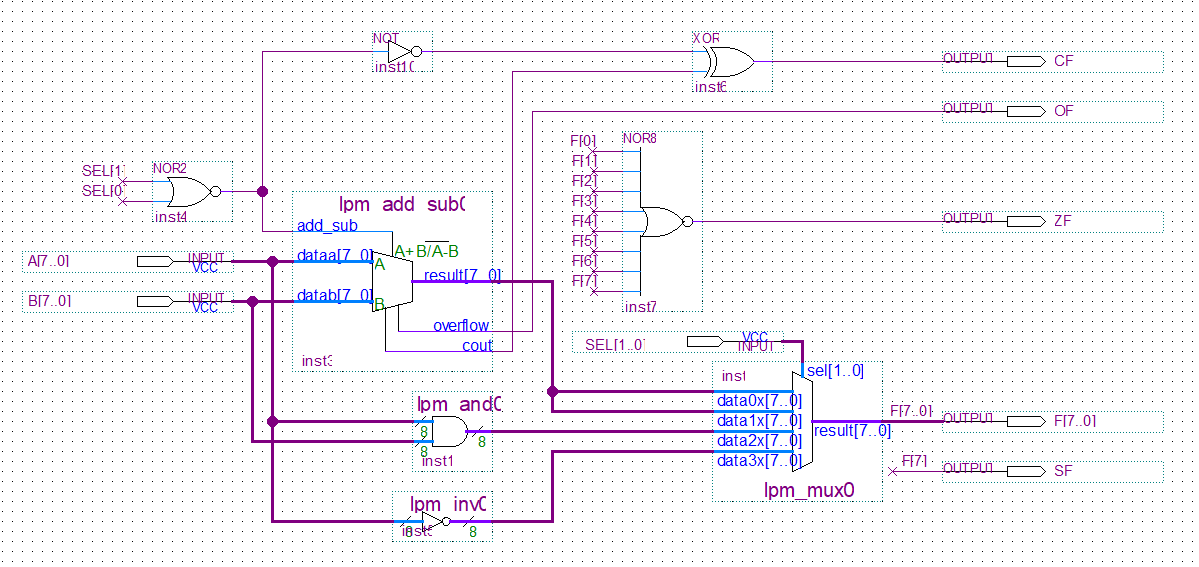
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能选择 | | 实现功能 | | | |
| SEL[1] SEL[0] | | 操作 | 助记符 | 功能函数 | 影响的状态标志 |
| 0 | 0 | 加法 | ADD | F＝A＋B | ZF、CF、OF、SF |
| 0 | 1 | 减法 | SUB | F＝A－B | ZF、CF、OF、SF |
| 1 | 0 | 逻辑与 | AND | F＝A·B | ZF |
| 1 | 1 | 逻辑非 | NOT | F＝ | ZF |

由上表可知，ALU的数据引脚A、B及F都为8位，控制引脚SEL为2位，还包含4根结果状态标志引脚。ALU中，加法、减法运算可用加/减法器来实现，逻辑与、逻辑非运算可用与门、非门来实现，当前操作的结果输出可用选择器来实现，ALU的内部组成如下图所示。



**图2.1 ALU组成**

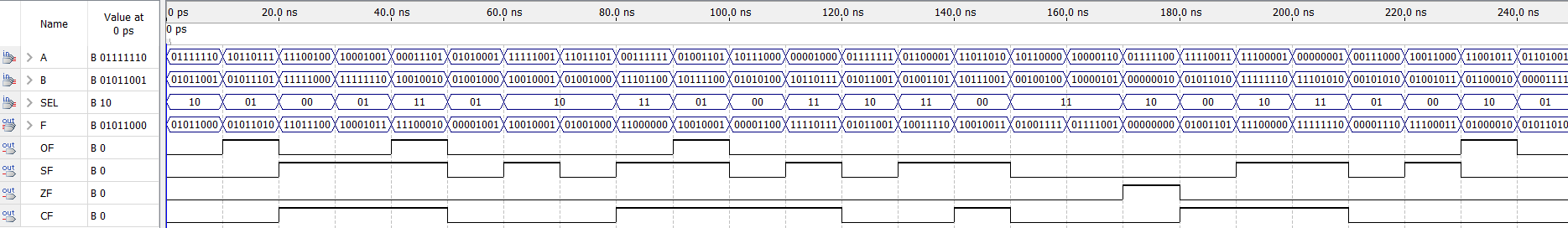
电路图如下：



**图2.2 电路图**

**三、电路正确性验证**

我们随机生成了A、B、sel的内容，得到仿真结果如下所示



**图2.3 仿真波形图**

**四、实验小结**

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Quartus II中其他的器件，如lpm\_compare。

通过本次实验，小组成员巩固了所学习的计算机组成原理知识，对ALU的工作原理有了更加深刻的认识。小组成员也通过本次实验认识到了合作的重要性，锻炼了勇于面对困难与未知的精神。

**五、教师评语**

教师签字：

日期：

**实验三 存储器设计及总线互联**

**一、实验内容**

* 测试RAM的功能。
* 设计一个读/写端口分离的128×8位的存储模块，前64B为只读空间，并验证设计正确性。
* 将所设计存储模块、输入部件、输出部件连接到地址线/数据线复用的8位总线上，通过输入部件对存储器进行操作、通过输出部件查看操作结果，来验证设计正确性。

**二、电路设计与实现**

本次实验我们用到的存储器是lpm\_ram\_dq，其特性是8位字长、128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，Clken的作用是允许、禁止clock信号进入芯片内部。



**图3.1 lpm\_ram\_dq引脚示例**

对lpm\_ram\_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）

由设计要求可知，存储模块MEM可使用1片64×8位ROM、1片64×8位RAM进行字扩展来实现。感兴趣的同学，可用64×4位ROM、64×4位RAM来实现。

读/写端口分离的64×8位ROM的引脚为：6位地址、8位数据输出、时钟脉冲clock、片选clken，64×8位的RAM的引脚还包含8位数据输入、写使能wren。

假设存储模块MEM的地址引脚为A[6..0]、数据输入引脚为D[7..0]、数据输出引脚为Q[7..0]、片选引脚为CS（高电平有效），则连接ROM、RAM时，地址引脚连接A[5..0]，数据输入、clock、wren引脚直接连接芯片对应引脚，各芯片数据输出引脚通过MUX连接到Q、用A[6]进行选择，ROM的clken＝CS·A[6]、RAM的clken＝CS·A[6]。

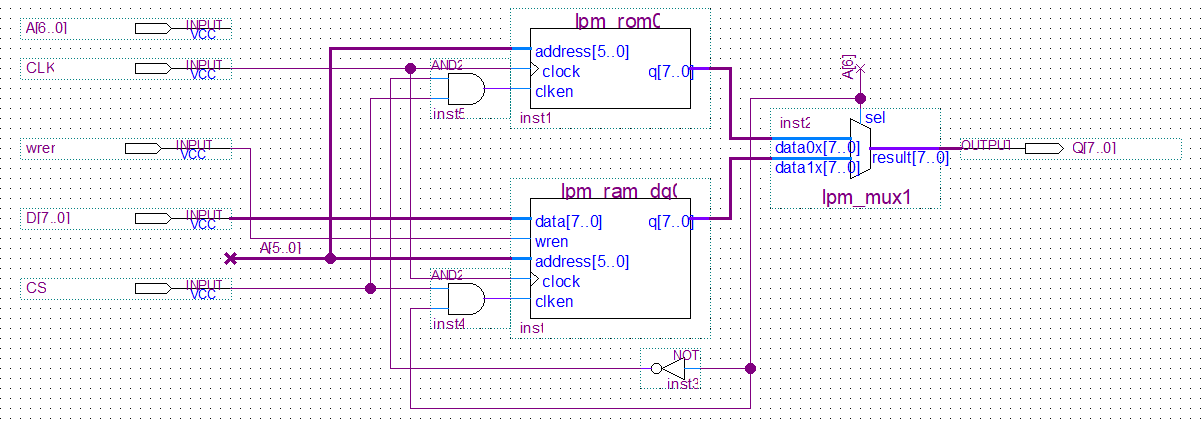
由于MEM是读/写端口分离的，其数据输出引脚Q需通过三态门连接到总线，且无需设置锁存器（Q与数据输入信号无关）；由于总线是地址线/数据线复用总线，MEM的数据输入引脚D直接连接到总线时，其地址引脚A需通过锁存器连接到总线。

同样，输入部件也需要通过三态门连接到总线，3个部件通过地址线/数据线复用总线互连的原理图如图7所示，图中还标出了所需的操作控制信号，电路C为MEM的片选信号CS的连接电路（MEM地址引脚A为7位、总线宽度为8位）。

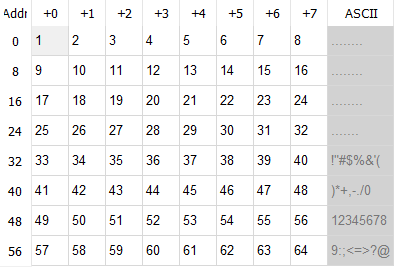


**图3.2 总线连接**

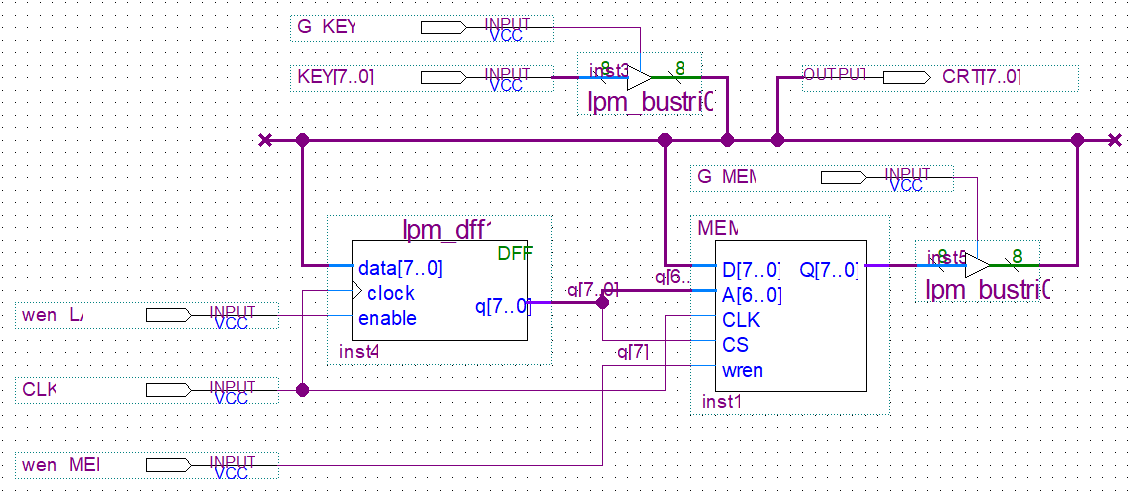
电路图如下：



**图3.3 MEM电路图**



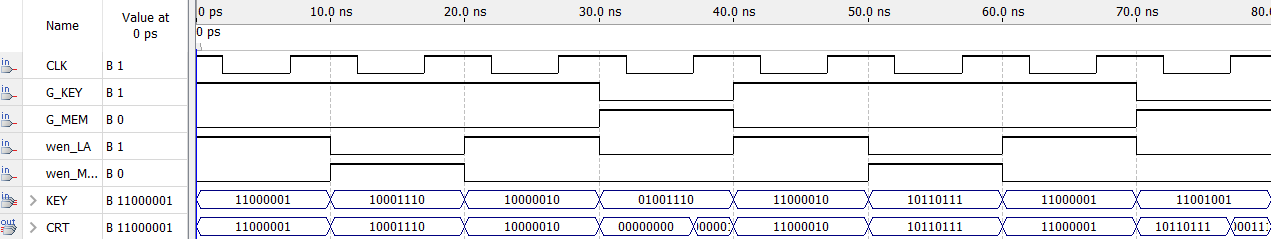
**图3.4 rom内容**



**图3.5 BUS电路图**

**三、电路正确性验证**

得到仿真结果如下所示



**图3.6 仿真波形图**

**四、实验小结**

通过这次试验，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。

通过本次实验，小组成员巩固了所学习的计算机组成原理知识，对ALU的工作原理有了更加深刻的认识。小组成员也通过本次实验认识到了合作的重要性，锻炼了勇于面对困难与未知的精神。

**五、教师评语**

教师签字：

日期：

**实验四 数据通路的组织**

**一、实验内容**

* 设计一个单总线结构的数据通路，支持Demo\_IS指令系统的取数(LD)、减法 (SUB)、双字长分支 (JNZ)指令。
* 将测试程序存入主存，根据程序执行过程的OPCmd微操作命令）序列，控制所设计数据通路，来验证数据通路的正确性。

**二、电路设计与实现**

**1、指令功能分析**

本实验要求支持Demo\_IS指令系统中的取数(LD)、减法(SUB)、双字长分支(JNZ)指令，其功能分别为：RD←M[(RS)]、RD←(RD)－(RS)、ZF＝0时PC←Addr。其中，RD、RS表示寄存器编号，(Rx)、M[(Rx)]表示寄存器、存储单元的内容，Addr为直接寻址方式的地址码，ZF为上一条指令所产生的结果状态标志。

**表4.1 Demo\_IS指令**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Demo\_IS Instructions | | | | | |
| Code | Byte 1 | | | Byte 2 | Specification |
| op | rd | rs | immediate/address |
| 7..4 | 3..2 | 1..0 | 7..0 |
| MOV | 0000 | rd | 00 | imme | rd ← imme |
| LD | 0010 | rd | rs | - | rd ← M[(rs)] |
| ST | 0011 | rd | rs | - | M[(rs)] ← (rd) |
| ADD | 0100 | rd | rs | - | rd ← (rd) + (rs) |
| 0101 | rd | rs | - | rd ← (rd) + M[(rs)] |
| SUB | 0110 | rd | rs | - | rd ← (rd) - (rs) |
| INC | 1000 | rd | 00 | - | rd ← (rd) + 1 |
| DEC | 1001 | rd | 00 | - | rd ← (rd) - 1 |
| JNZ | 1100 | 00 | 00 | addr | if(!ZF) PC←addr |
| 1101 | disp | | - | if(!ZF) PC←(PC)+disp |

由Demo\_IS的指令格式及上述3条指令功能约定，可得到如下分析结果：

* 数据类型只有8位整数一种，采用定点格式（补码编码）表示；
* 数据操作只有8位的加法、减法两种，需产生状态标志ZF；
* 数据寻址有寄存器寻址、寄存器间接寻址两种方式，地址无需计算；
* 指令寻址有直接寻址、隐含寻址两种方式，地址计算方法为8位加法；
* 寄存器有4个，长度为8位，每条指令最多2次读、1次写操作；
* 存储器按字节编址、地址空间为8位，每条指令最多有1次读/写操作。

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。

**2、数据通路的设计与实现**

数据通路由通路部件、部件互连两部分组成，互连结构有总线结构、点点结构两种类型。本实验的数据通路要求采用单总线结构。

**（1）功能部件设计**

为了满足Demo\_IS中3条指定指令的要求，数据通路的功能部件应包括ALU、寄存器组GPRs、状态寄存器PSR、存储器MEM、程序计数器PC、指令寄存器IR，以及地址寄存器MAR、数据寄存器MDR。

由Demo\_IS的分析结果可见，ALU应具有加法、减法功能，需产生状态标志ZF，可使用实验二设计的ALU来实现；GPRs应包含4个8位寄存器，具有1个读端口和1个写端口，可使用实验一设计的寄存器组来实现；MEM的容量应≤256×8位，可使用实验三设计的存储模块（容量为128×8位）来实现。为了简化控制，假设PC具有计数功能，可使用Quartus II提供的lpm\_counter模块来实现。

**（2）部件互连设计**

单总线结构的数据通路中，所有部件的数据入端、数据出端都连接在同一个总线上。为了保证数据传送的正确性，部件的出端需通过三态门连接到总线；部件的入端及出端中，只有1个可以直接连接总线，其余都需通过锁存器连接到总线。

本实验中，数据通路的宽度为8位，数据通路的组成如下图所示，TS0~TS3为三态门，GPRs的读地址引脚前增加选择器MUX，可以满足SUB指令中2次读GPRs的需求，当前读地址（RS或RD）用控制信号GRsel来选择。



**图4.1 单总线结构的数据通路组成**

图中的MEM需要存放测试程序，故要求使用实验三设计的存储模块（前64B为ROM空间）来实现，MEM片选引脚的连接也类似于实验三。由于实验三的存储模块读/写端口分离、未设置读操作控制信号、可在一个时钟周期内完成读写操作，故MDR由MDRI及MDRO实现，控制信号MemRd可以用来实现MDRO的写入控制（MDRinB）。

PSR用于存放结果状态标志，供CU使用，由于本实验的μOPCmd由手工给出，故判断ZF＝0也由人工完成，PSR可以省略（用虚线表示）。同理，ID也可省略。

**（3）数据通路的实现**

数据通路的实现需要编辑原理图文件（如DataPath.bdf），电路有多种实现方法，ALU、GPRs、MEM、PC都按照功能部件设计的方案来实现，IR、MAR、MDR、Y、Z应使用lpm\_dff实现，所有的部件控制信号都需设置为输入引脚。

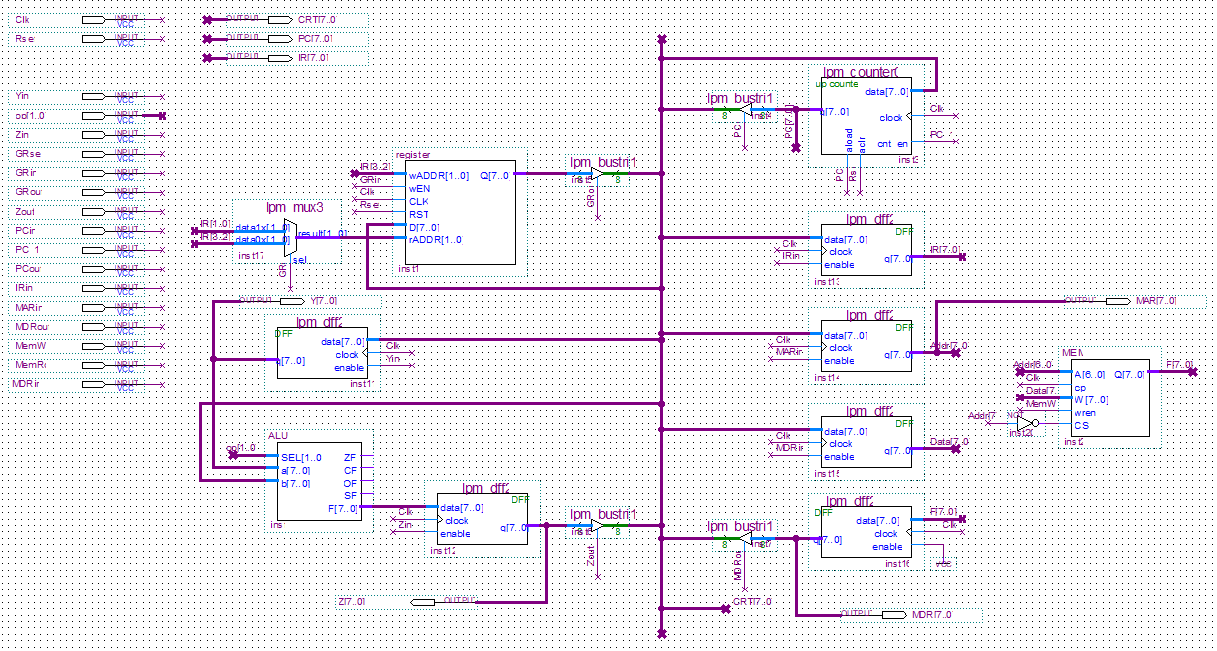
数据传送μOP应采用电位-脉冲制来实现，即发送部件在时钟周期开始时就输出数据（如打开三态门），接收部件在时钟周期结束前接收数据并写入。如图9所示，μOPCmd由CU在时钟周期开始时（clock上升沿）通过电路产生（有一定时延[图9中①]），控制相应三态门打开（有一定时延[图9中②]），数据写入只能在时钟周期结束时（即下个clock上升沿[图9中④]）进行。

注意，MAR、MDRI、MDRO应在时钟周期结束时写入，由于图8中的MEM为同步RAM，故MEM操作应安排在时钟周期中部（clock下降沿）开始。

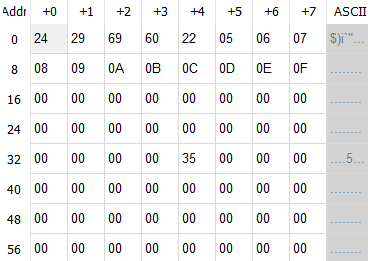
**表4.2 OPCmd**

|  |  |  |  |
| --- | --- | --- | --- |
| 微操作 | 时钟周期 | OP | OPCmd |
| R1←M[(R0)] | t1 | MAR←(PC) | PCout、MARin |
| t2 | MDR←M[(MAR)],PC←(PC)＋1 | Read、WMFC，PC+1 |
| t3 | IR←(MDR) | MDRout、IRin |
| t4 | MAR←(R0) | GRout、Rsel、MARin |
| t5 | MDR←M[(MAR)] | Read、WMFC |
| t6 | R1←(MDR)，End←1 | MDRout、GRin，End |
| R2←M[(R1)] | t1 | MAR←(PC) | PCout、MARin |
| t2 | MDR←M[(MAR)],PC←(PC)＋1 | Read、WMFC，PC+1 |
| t3 | IR←(MDR) | MDRout、IRin |
| t4 | MAR←(R1) | GRout、Rsel、MARin |
| t5 | MDR←M[(MAR)] | Read、WMFC |
| t6 | R2←(MDR)，End←1 | MDRout、GRin，End |
| R2←(R2)－(R1) | t1 | MAR←(PC) | PCout、MARin |
| t2 | MDR←M[(MAR)],PC←(PC)＋1 | Read、WMFC，PC+1 |
| t3t | IR←(MDR) | MDRout、IRin |
| t4 | Y←(R2) | GRout、Yin ;Rsel=0 |
| t5 | Z←(Y)－(R1) | GRout、Rsel、op＝01、Zin |
| t6 | (R2)←(Z)，End←1 | Zout、GRin，End |
| JNZ 22H | t1 | MAR←(PC) | PCout、MARin |
| t2 | MDR←M[(MAR)],PC←(PC)＋1 | Read、WMFC，PC+1 |
| t3 | IR←(MDR) | MDRout、IRin |
| t4 | MAR←（PC） | PCout、MARin |
| t5 | MDR←M[(MAR)];PC←PC+1 | Read、WMFC |
| t6 | PC←(MDR);End←1 | MDRout、PCin、End |

电路图如下：



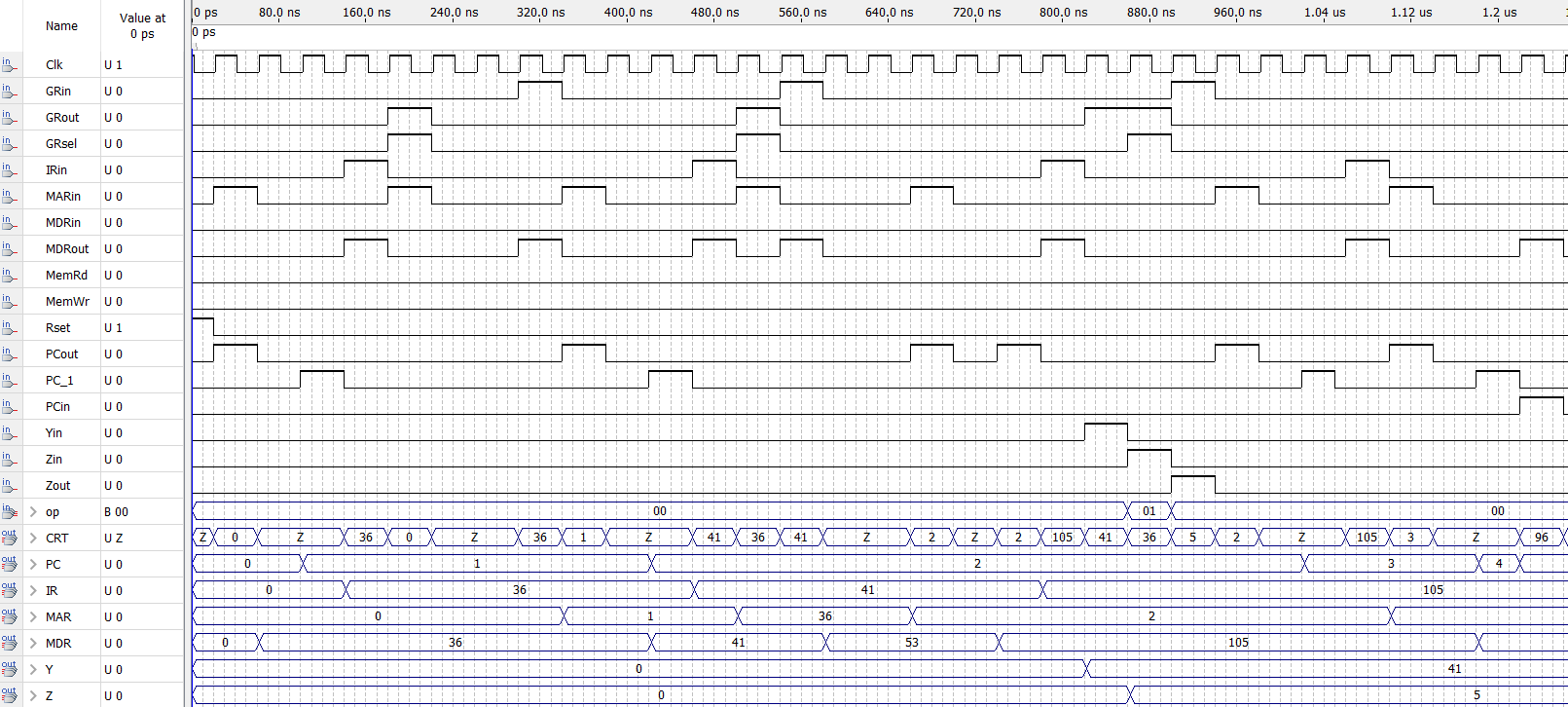
**图4.2 MEM电路图**



**图4.3 rom内容**

**三、电路正确性验证**

得到仿真结果如下所示



**图4.4 仿真波形图**

**四、实验小结**

通过这次试验，加深了对前三个实验的理解，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。

通过本次实验，小组成员巩固了所学习的计算机组成原理知识，对ALU的工作原理有了更加深刻的认识。小组成员也通过本次实验认识到了合作的重要性，锻炼了勇于面对困难与未知的精神。

**五、教师评语**

教师签字：

日期：