**计算机组织与结构**

**实验讲义**

**（2019版）**

**任国林 编写**

**东南大学计算机科学与工程学院**

计算机组织与结构的课程实验是为巩固教学效果而设置的，目的是希望学生通过实验，能够加深对计算机组成及工作原理的理解，熟练使用数字电路芯片，提高数字电路的逻辑设计能力，为设计计算机整机打下基础。

计算机组织与结构的课程实验共包含4个实验，分别是寄存器组的设计、ALU的设计、存储器扩展及总线互连、数据通路的组织。

所有实验都要求基于Quartus II（或ISE）进行电路实现和正确性验证，要求采用原理图方式实现电路，采用时序仿真方式进行仿真，条件允许时基于FPGA芯片进行电路验证。实验讲义基于Quartus II撰写。

为了减少实验所花时间，实验四可以使用前三个实验的结果，因此，四个实验应使用同一个工程文件，所有文件放在同一个文件夹下。

## 实验一 寄存器组的设计

### 一、实验目的

（1）重温基于Quartus II的数字电路设计及仿真方法。

（2）熟悉D触发器的功能及使用方法。

（3）掌握寄存器组的组成原理。

### 二、实验内容

（1）测试D触发器的功能表。

（2）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。

### 三、实验原理及方案

#### 1、基于Quartus II的数字电路设计及仿真

基于Quartus II，电路设计的过程主要包括：建立工程文件、编辑原理图文件、编译原理图文件。电路仿真的过程主要包括：编辑仿真波形文件、生成功能仿真网表（仅功能仿真方式需要）、开始仿真、分析仿真结果的正确性。

下面，以Z=X·Y为例，说明使用Quartus II、采用原理图方式进行电路设计的过程，该工程文件可以用作4个实验共用的工程文件。

1）建立工程文件：通过主菜单File→New Project Wizard，可以建立工程文件，包含5个页面的设置。

·第1页为设置工程文件信息，含工程文件名（如COExp）、工作目录名（如COExp）

·第2页为在工程中加入文件，可将已有的bdf文件添加到工程文件中，本例中无

·第3页为选择FPGA器件型号，建议采用Cyclone III系列的EP3C16Q240C8芯片

·第4页为添加准备使用的EDA工具，通常直接选择Next

·第5页为查看、确认工程文件信息，无误时选择Finish即可。

2）编辑原理图文件：通过主菜单File→New→Device Design Files→Block Diagram/schematic File，可以进入原理图编辑器，进行电路编辑（实现所设计电路）；通过主菜单File→Save可以保存为原理图文件。注意，电路应包括输入引脚、输出引脚，电路编辑包括选择元器件、引脚连接等环节，本例中电路有2个输入引脚、1个输出引脚，内部逻辑为1个2输入端的与门，假设文件名为demo1.bdf。

3）设置顶层文件：在Project Navigator窗口的File页面中，选择需要编译或仿真的原理图文件，通过该文件的右键菜单Set as Top-Level Entity，可设置该文件为顶层文件。本例中选择的文件名为demo1.bdf。

4）编译顶层文件：通过主菜单Processing→Compiler Tool进入编译界面，按Start开始编译。编译时，提示窗口中将显示编译相关信息，包括错误信息。编译有错误时，修改原理图文件，重新编译；编译成功后，可进入仿真阶段。

下面，以原理图文件demo1.bdf为例，说明使用Quartus II进行电路仿真的过程：

1）编辑仿真波形文件：通过主菜单File→New→Other Files→Vector Waveform File，可以进入波形文件编辑器，进行顶层文件的波形文件编辑；选择所需的I/O引脚，设置各输入引脚信号波形后，通过主菜单File→Save保存为波形文件。已有波形文件的编辑可以通过主菜单File→Open进入。注意，一个原理图文件可以对应有多个波形文件，仿真时可以进行选择；波形文件中的信号组合，应覆盖电路功能表的全部功能。

2）生成功能仿真网表：常用的仿真方式有功能仿真（Functional）、时序仿真（Timing）两种，该步骤仅用于功能仿真方式。注意，原理图修改后，开始仿真前需要重新进行编译、生成功能网表。本课程的四个实验都要求采用时序仿真方式，故该步骤可以省略。

3）开始仿真：按Start开始进行仿真，通过Report或VCD文件可以查看仿真波形。

4）分析仿真结果的正确性：对照所仿真电路的功能表，分析电路的功能是否正确。

上述介绍仅为总体过程，具体操作步骤参见所发的Quartus II使用指南。

#### 2、D触发器功能的测试

触发器有边沿触发、电位触发两种方式，通常，前者称为触发器，后者称为锁存器。触发器的状态在时钟脉冲信号CP的上升沿到来时发生变化、在其余时间保持不变；锁存器的状态在控制信号E为高电平时随输入端变化而变化、为低电平时保持不变。

Quartus II提供了多种类型的触发器，如D触发器、T触发器等。参数固定的触发器有固定的型号，如74173、74273等；参数可变（又称参数化）的触发器有lpm\_ff、lpm\_dff、lpm\_tff等，型号如lpm\_dff0、lpm\_dff1等。寄存器常由D触发器组成。

图1为8位D触发器lpm\_dff的I/O引脚示例，可以缺省部分引脚。其中，data[7..0]、q[7..0]分别为数据输入、数据输出引脚，aclr、aset分别为异步的清零、置位引脚，同步的清零、置位引脚为sclr、sset，clock为时钟脉冲引脚，enable为写使能引脚。



lpm\_dff的状态（所存储信息）由q[7..0]直接输出，无需信号控制。对lpm\_dff的操作有清零、置位、写入三种。异步清零（aclr＝1、aset＝0）时，q立即变为全0；异步置位（aclr＝0、aset＝1）时，q立即变为全1；同步清零、同步置位时，q在clock上升沿后变为全0、全1。写操作（enable＝1、aclr＝0、aset＝0）时，data上信息在clock上升沿时写入触发器。

Quartus II可以省略芯片不使用的引脚，如sclr、sset等引脚，缺省引脚的信号都为无效状态。不建议省略enable引脚，以避免毛刺的产生。

触发器功能的测试可以放在寄存器组功能的验证中进行，实现方法是：寄存器组的所有输入信号中，与触发器无关的信号保持不变，改变与触发器相关的信号进行测试。

#### 3、寄存器组的设计、实现及验证

寄存器组将多个寄存器组织在一起，按地址进行操作（读/写/清零），以简化控制。

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。因此，寄存器组的地址引脚为log24＝2位，数据引脚为8位。由于要求读/写端口分离，因此，读操作相关的引脚有地址raddr[1..0]、数据输出q[7..0]，写操作相关的引脚有地址waddr[1..0]、数据输入data[7..0]、写使能wen、时钟脉冲Clk，清零操作的引脚为Clr（高电平有效）。注意，读操作无需控制信号，即寄存器的数据输出不受限制。

（1）寄存器组的设计

寄存器组中，写操作通过waddr[1..0]指定寄存器，通过wen、Clk实现写入控制，指定寄存器可使用译码器来实现；读操作通过raddr[1..0]选择寄存器，读出无需信号控制，选择可使用选择器来实现。寄存器组的组成如图2所示。



图2 寄存器组的组成

（2）寄存器组的实现

使用原理图方式实现寄存器组时，具体过程包括：创建/打开工程文件、编辑原理图文件（如GPRs.bdf）、编译原理图文件。其中，编辑原理图文件就是实现所设计的电路。

编辑原理图文件时，所设计电路有多种实现方法，译码器、选择器可以使用Quartus II提供的参数化模块lpm\_mux、lpm\_decode。注意，多个相似的、由字符及数字组成的信号线命名时，数字的位数应相同；信号线的连接有连线、信号线同名等多种方法；信号线的换名可使用器件wire来实现；添加lpm\_器件时，应从symbol Tool中选择，不能从其他bdf文件中拷贝，否则编译时通常会产生错误（缺少定义或定义冲突）。

编译原理图文件时，需要先将原理图文件设置为顶层文件。

（3）寄存器组的验证

触发器功能没有单独测试时，寄存器组的验证有测试寄存器功能、验证寄存器组功能两个环节。测试/验证都通过仿真＋分析方法来实现，仿真都采用时序仿真方式进行。

基于Quartus II进行测试/验证的过程包括：编辑仿真波形文件、开始仿真、分析仿真结果的正确性（或功能特性）。其中，编辑仿真波形文件就是设计电路测试方案。为了便于从不同侧面进行测试，一个电路可以对应有多个仿真波形文件。

编辑仿真波形文件时，可以使用同一个仿真波形文件（如GPRs1.vwf）进行寄存器及寄存器组的功能测试/验证。对于寄存器功能测试，使waddr、raddr保持不变（如＝0），改变各控制信号（Clr、wen）的值，以输出寄存器的所有功能特性。对于寄存器组功能验证，改变waddr、raddr及各控制信号的值，以输出所设计电路的所有功能特性。

注意，输入信号的组合需能够反映电路的所有功能，如Clr与wen有4种组合；时钟脉冲信号Clk的周期应大于每种操作的时延，Clk的上升沿应放在时钟周期的中部（电位-脉冲制）；所有输入信号都应以时钟周期为单位，即值的改变与Clk对齐。

分析仿真结果的正确性时，应以时钟周期为单位进行分析，查看每个输出信号的变化及时序，是否满足电路功能表的要求。

### 四、实验要求

（1）做好实验预习。了解触发器的功能特性、寄存器组的组成方法，基于Quartus II提供的元器件，画出电路图，标明引脚名，写出仿真时的操作序列（最好含信号取值）。

（2）完成实验内容。实现所设计电路，验证电路正确性，保存仿真波形文件。

（3）撰写实验报告。按所给模板撰写，等四个实验做完后一起交上来。

（4）保存实验电路图。由于第四个实验需要使用前三个实验所设计的电路，因此，每个实验的电路图都必须保存起来。由于每个工程文件的同类lpm\_器件，都从0开始编号，因此，为了实现已有电路的重用，所有实验应该使用同一个工程文件，所有文件放在同一个文件夹下；否则，编译时lpm\_器件易产生错误，需要重新修改电路。切记！切记！

## 实验二 ALU的设计

### 一、实验目的

（1）熟悉加/减法器的功能及使用方法。

（2）掌握ALU的组成原理。

### 二、实验内容

（1）测试Quartus II所提供加/减法器的功能特性。

（2）设计具有加法、减法、逻辑与、逻辑非4种功能的8位ALU，ALU可产生结果状态标志ZF/CF/OF/SF，并验证设计正确性。

### 三、实验原理及方案

ALU的核心是加/减法器，加/减法器的基础为加法器，产生的结果状态标志可以用来实现关系运算功能。

#### 1、加/减法器功能的测试

Quartus II提供的lpm\_add\_sub模块可实现加法、减法运算，可输出溢出标志overflow、最高位进位cout。注意，控制引脚add\_sub为1、0时分别实现加法、减法（与常见约定相反）；cout是加法器最高位进位，不是CF；有些Quartus版本中，lpm\_add\_sub可以选择运算是无符号运算还是有符号运算，选择无符号运算时overflow永远为0。

加/减法器功能的测试可以放在ALU功能的验证中进行，实现方法是：ALU能够输出加/减法器的所有信号（需要时可增加输出引脚），改变与加/减法器有关的信号进行测试。

#### 2、ALU的设计、实现及验证

ALU能够实现多种算术运算、逻辑运算功能，其功能由指令系统决定。ALU的引脚由数据入端A和B、数据出端F、功能选择信号SEL、结果状态标志信号组成，F＝function(A, B)，当前所实现的功能由SEL决定。

本实验要求的ALU数据宽度为8位，具有4种算术及逻辑运算功能，可产生的结果状态标志为：零标志ZF、进位/借位标志CF、溢出标志OF、符号标志SF。因此，ALU的功能选择引脚为2位（记为SEL[1..0] ），其功能表如表1所示。

表1 ALU功能表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 |

（1）ALU的设计

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的组成如图3所示。



图3 ALU的组成

其中，状态形成电路负责产生4个结果状态标志，控制信号形成电路负责产生各模块所需的操作控制信号，本实验中只有加/减法器、状态形成电路需要使用控制信号。注意，结果状态标志ZF是由输出的数据形成的。

（2）ALU的实现

使用原理图方式实现ALU时，具体过程包括：打开工程文件、编辑原理图文件（如ALU.bdf）、编译原理图文件。其中，编辑原理图文件就是实现所设计的电路，打开工程文件可以实现不同实验共用同一个工程文件。

编辑原理图文件时，所设计电路有多种实现方法，加/减法器、与门、非门可以使用Quartus II提供的lpm\_add\_sub、lpm\_and、lpm\_inv模块。注意，为了能够用ALU测试加/减法器的功能特性，需要在lpm\_add\_sub的cout引脚处增加一个输出信号cout，lpm\_add\_sub的其余引脚都已经可以直接/间接输出了。

编译原理图文件时，需要先将原理图文件设置为顶层文件。

（3）ALU的验证

加/减法器功能没有单独测试时，ALU的验证有测试加/减法器功能、验证ALU功能两个环节。测试/验证都通过仿真＋分析方法来实现，仿真都采用时序仿真方式进行。

基于Quartus II进行测试/验证的过程包括：编辑仿真波形文件、开始仿真、分析仿真结果的正确性（或功能特性）。

编辑仿真波形文件时，可以只使用一个仿真波形文件（如ALU1.vwf）。对于加/减法器功能测试，分别使SEL为00及01，改变A及B的值，以输出F、OF及cout的所有特性。对于ALU功能验证，改变SEL、A及B的值，以输出所设计电路的所有功能特性。

注意，数据及SEL的取值应能够反映状态标志的有效逻辑，如测试加/减法器功能时，SEL＝00及01，应该有5×2＝10组输入数据，其中3×2组数据（++/+-/--）测试运算功能、2×2组数据测试overflow（正溢出及负溢出）；又如，验证ALU功能时，应有4组数据验证CF的有效逻辑（可放在加/减法器测试中进行），OF的有效逻辑则无需验证（加/减法器中已测试过）。

分析仿真结果的正确性时，应以操作为单位进行分析，查看每个SEL及输入数据对应的输出信号是否正确，同时注意输出信号的时延特性。

### 四、实验要求

（1）做好实验预习。了解加/减法器的功能特性、ALU的组成原理，基于Quartus II提供的元器件，画出电路图，标明引脚名，写出仿真时的操作序列及输出结果。

（2）完成实验内容。实现所设计电路，验证电路正确性，保存仿真波形文件。

（3）撰写实验报告。按所给模板撰写，等四个实验做完后一起交上来。

（4）保存实验电路图。原因同实验一。

## 实验三 存储器扩展及总线互连

### 一、实验目的

（1）熟悉RAM的功能及使用方法。

（2）掌握存储器的容量扩展方法。

（3）掌握基于总线的部件互连方法及操作方法。

### 二、实验内容

（1）测试Quartus II所提供的RAM的功能特性。

（2）设计一个读/写端口分离、容量为256×8位的存储模块，前128B为只读空间、后128B为读写空间，并验证设计正确性。

（3）将存储器、输入部件、输出部件连接到地址/数据复用的总线上，能够通过输入部件对存储器进行读/写操作，并验证设计正确性。

### 三、实验原理及方案

#### 1、半导体存储器功能的测试

Quartus II提供了多种类型的RAM模块，如lpm\_dq、lpm\_dp、lpm\_rom等，这些模块都为同步存储器，即读/写操作都在时钟脉冲信号的上升沿开始；读操作都可以选择是否需要信号控制；读操作的数据输出都可以选择是否带输出锁存功能。

以256×8位的读/写端口分离的lpm\_dq模块为例，其I/O引脚示例如图4所示。其中，data[7..0]、q[7..0]分别为数据输入引脚、数据输出引脚，wren为写使能引脚（高电平有效），clock为时钟脉冲引脚，clken为时钟脉冲使能引脚。可见该lpm\_dq的读操作无需信号控制。clken＝1时clock上信号可进入芯片内部（为脉冲信号），clken＝0时clock上信号被封锁（为低电平），因此，clken可以用作片选信号。



对lpm\_dq的操作有读、写两种。写操作时（wren＝1、clken＝1），地址及数据在clock上升沿被锁存，然后数据被写入到指定的存储单元中。读操作时（clken＝1），地址在clock上升沿被锁存，然后指定存储单元的数据被输出到引脚q（lpm\_dq无输出锁存功能时）或内部锁存器（lpm\_dq带输出锁存功能时）。注意，当读操作使用信号rden控制时，读操作的有效逻辑是rden＝1、clken＝1。

存储器功能的测试可以放在总线互连的验证中进行。

#### 2、存储器扩展的设计、实现及验证

存储器的容量扩展有位扩展、字扩展、字位扩展3种方法。本实验要求设计一个读/写端口分离、容量为256×8位的存储模块MEM，前128B为只读空间、后128B为读写空间。

（1）存储器扩展的设计

由设计要求可知，存储模块MEM可以用128×8位ROM、128×8位RAM进行字扩展获得。

读/写端口分离的128×8位同步ROM的引脚为：7位地址、8位数据输出、时钟脉冲clock、片选clken，128×8位的同步RAM的引脚还有8位数据输入、写使能wren。

假设存储模块MEM的地址引脚为A[7..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，则ROM、RAM连接时，地址引脚连接A[6..0]，数据输入、clock、wren引脚直接连接对应引脚，数据输出引脚通过选择器连接到Q（用A[7]进行选择），ROM的clken＝CS·A[7]、RAM的clken＝CS·A[7]。

（2）存储器扩展的实现及验证

存储器扩展的实现及验证可以放在总线互连的验证中进行。

#### 3、部件通过总线互连的设计、实现与验证

由于总线上同时只能有一个部件发送数据，因此，每个部件的输出端必须通过三态门连接到总线上。当部件有多个输入端时，为了防止不同输入端之间的信号干扰，输入端连接到总线之前需要设置锁存器；为了防止输入端与输出端之间的信号反馈（仅针对组合逻辑部件），输出端连接到总线之前需要设置锁存器，只有一个输入端或输出端可以直接连接到总线上。

本实验要求将存储器、输入部件、输出部件连接到地址/数据复用总线上，并能够通过输入部件对存储器进行读/写操作。为了便于进行存储器扩展的验证，总线宽度设定为8位，存储器的容量为256×8位，刚好可以使用扩展的存储模块实现。

（1）部件通过总线互连的设计

由于存储器是读/写端口分离的，因此，数据输出引脚需要通过三态门连接到总线。由于实验所用的总线是地址/数据复用总线，因此，地址引脚连接总线时需要设置地址锁存器，以防止地址引脚、数据输入引脚之间存在信号干扰。由于存储器是时序逻辑部件，输出信号不完全受输入信号的影响，因此，数据输出引脚连接总线时无需设置锁存器。

同样，输入部件也需要通过三态门连接到总线，三个部件通过总线互连的原理图如图5所示。



图5 部件通过总线互连的连接电路

为了便于进行存储器扩展的测试，LA及MEM的输出端需要增加输出引脚Maddr[7..0]及Mout[7..0]，这样就可以实时观察存储器的操作结果了。

（2）部件通过总线互连的实现

使用原理图方式实现总线互连时，具体过程包括：打开工程文件、编辑原理图文件（如Bus.bdf）、编译原理图文件。其中，编辑原理图文件就是实现所设计的电路，打开工程文件可以实现不同实验共用同一个工程文件。

编辑原理图文件时，所设计电路有多种实现方法，ROM、RAM、三态门、输入部件、输出部件可以使用Quartus II提供的lpm\_rom、lpm\_dq、lpm\_bustri、input、output模块，锁存器LA可以用触发器或锁存器实现，建议使用触发器（如lpm\_dff）实现，因为实验四需要使用触发器方法来实现，早些使用可以减少实验四所花时间。

注意，扩展后的存储模块MEM应设置片选引脚CS；lpm\_rom及lpm\_dp都应选择无输出锁存功能，以便于能够立即输出操作结果。

lpm\_rom中存储单元的内容需要预先写入，可以通过设置初始化文件（如rom.mif）来实现，设置操作在lpm\_rom的MEM Init页面中完成。注意，应使每个存储单元的内容都不同，以防止遗漏操作错误。

编译原理图文件时，需要先将原理图文件设置为顶层文件。

（3）部件通过总线互连的验证

存储器扩展功能没有单独测试时，部件通过总线互连的验证有测试存储器扩展正确性、验证部件互连正确性两个环节。测试/验证都通过仿真＋分析方法来实现，仿真都采用时序仿真方式进行。

基于Quartus II进行测试/验证的过程包括：编辑仿真波形文件、开始仿真、分析仿真结果的正确性（或功能特性）。

编辑仿真波形文件时，可以使用同一个仿真波形文件（如Bus1.vwf）。对于存储器扩展测试，ROM空间的操作至少有2个，RAM空间的每种操作至少有2个，建议操作序列为读单元a1、写单元b1、读单元a2、写单元b2、读单元b1、读单元b2，其中a1、a2为ROM中的单元。对于部件互连正确性验证，建议操作序列为写单元b1、读单元c、写单元b2、读单元b1即可。

注意，每个操作都需要2个时钟周期实现（写LA、访存）；时钟脉冲信号Clk的周期应大于操作时延，Clk的上升沿应安排在时钟周期的中部；所有输入信号都应以时钟周期为单位，即值的改变与Clk对齐。

分析仿真结果的正确性时，应以时钟周期为单位进行分析，查看每次输入信号对应的输出信号是否与预期结果一致。分析存储器扩展正确性时，关注Maddr[7..0]、wren及Mout[7..0]信号；分析总线互连正确性时，关注输入部件、输出部件信号即可。

### 四、实验要求

（1）做好实验预习。了解RAM的组成原理、单总线的部件互连方法，基于Quartus II提供的元器件，画出电路图，标明引脚名，写出仿真时的操作序列（含信号取值）。

（2）完成实验内容。实现所设计电路，验证电路正确性，保存仿真波形文件。

（3）撰写实验报告。按所给模板撰写，等四个实验做完后一起交上来。

（4）保存实验电路图。原因同实验一。

## 实验四 数据通路的组织

### 一、实验目的

（1）了解数据通路的组织方法。

（2）掌握指令执行过程的控制原理。

### 二、实验内容

（1）设计一个单总线结构的数据通路，支持教材中Demo\_IS指令系统的取数(LD)、减法(SUB)、双字长的分支(JNZ)指令。

（2）编写测试程序，给出该程序执行过程的μOPCmd序列，来验证数据通路的正确性。

### 三、实验原理及方案

#### 1、指令功能分析

本实验所指的Demo\_IS指令系统如教材中图4.16所示，其取数(LD)、减法(SUB)、双字长的分支(JNZ)指令的功能如下：

取数LD— RD←M[(RS)]

减法SUB— RD←(RD)－(RS)

分支JNZ— ZF＝0时PC←Addr，ZF＝1时PC←(PC)＋1

其中，RD、RS表示寄存器编号，(Rx)、M[(Rx)]表示寄存器寻址、寄存器间接寻址方式，Addr表示直接寻址方式，ZF为上一条指令所产生的结果状态标志。

由Demo\_IS的指令格式及指令功能约定可知，数据类型有8位整数、8位逻辑数两种，分别用定点格式（补码/无符号编码）、位向量格式表示；数据操作只有加法、减法两种，需要产生状态标志ZF；寄存器有4个，长度为8位，每条指令最多有2次读操作、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。

#### 2、数据通路的设计与实现

数据通路由通路部件、互连结构两部分组成，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FLAG、存储器MEM、程序计数器PC及指令寄存器IR。为了能够实现访存μOP与其他μOP的并行，功能部件还应包括地址寄存器MAR、数据寄存器MDR。

由Demo\_IS指令系统分析结果可见，ALU应具有加法、减法功能，建议使用实验二设计的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，建议使用实验一设计的寄存器组来实现；MEM的容量应为256×8位、读/写端口分离，建议使用Quartus II提供的lpm\_dq模块来实现，不建议使用实验三的设计结果（很可能有瑕疵）。为了简化控制，假设PC具有计数功能，建议用Quartus II提供的lpm\_counter模块来实现。

要使用实验一、实验二的设计结果，就需要先生成相应的符号文件（.bsf），然后就可以像使用lpm\_dq等器件一样，使用自行设计的电路。

（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，部件的出端都需要通过三态门连接到总线，部件间直接连接除外；部件有多个入端时，入端连接总线时需要设置锁存器，以防止入端之间发生信号干扰，如ALU的入端A前应增设锁存器Y；对于组合逻辑部件，还需要防止入端与出端之间存在信号反馈，处理方法也是增设锁存器，如ALU的出端后应增设锁存器Z。

本实验中，数据通路的宽度为8位，数据通路的组成如图6所示，与教材中的图5.7基本相同。其中TSL\_0~TSL\_4为三态门。图中还增加了输入部件KEY及输出信号BUS及IR，以便于调试。学生可以根据需要自行增加输出信号，要求至少输出PC、IR、RS、RD。



图6 单总线结构的数据通路组成

为了在指令执行过程中能够复用部件，部件的引脚信号应该可以改变，因此，需要增加相应电路满足指令功能的需求。例如SUB指令执行过程中，需要2次读GPRs、1次写GPRs，读操作的地址来自指令字的RS、RD，写操作的地址来自指令字的RD，因此，图6中GPRs的读地址引脚前需要增设选择器MUX1，来选择当前连接的是IR中的RD还是RS。

功能部件设计时建议MEM用lpm\_dq来实现，这是由于lpm\_dq具有输出锁存功能（可选），因此，MEM带输出锁存功能时，图6中的MDR就可以省略（用虚线表示）。FLAG存放结果状态标志，供CU使用，由于本实验的μOPCmd由手工给出，因此，判断ZF＝0也是由人工完成的，故FLAG可以省略（用虚线表示）。

（3）数据通路的实现

使用原理图方式实现数据通路时，具体过程包括：打开工程文件、编辑原理图文件（如DataPath.bdf）、编译原理图文件。其中，编辑原理图文件就是实现所设计的电路，打开工程文件可以实现不同实验共用同一个工程文件。

编辑原理图文件时，ALU、GPRs、MEM、PC按照功能部件设计的方案实现，IR、MAR用lpm\_dff实现，Y、Z也应该用lpm\_dff实现（具体原因后面会有说明）。

注意，MEM用lpm\_dq实现时，若MEM带输出锁存功能就需要省略MDR；部件控制信号的命名尽量与教材相同，以便于加深理解，图6中已列出了部分信号名。

为了正确地进行数据传送，μOP控制应采用电位-脉冲制，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的中部（上升沿）接收数据并写入。因而，图6中的锁存器Y和Z等都应该用触发器lpm\_dff实现，并且时钟脉冲信号的上升沿都应安排在时钟周期的中部，μOPCmd可以在时钟周期开始时就有效。

编译原理图文件时，需要先将原理图文件设置为顶层文件。

#### 3、数据通路的验证

本实验要求通过执行测试程序来验证数据通路设计的正确性。测试程序需要自行编写，但必须包含3种指令。测试程序示例如下：

R1←M[(R0)] ；单字长指令

R2←M[(R1)] ；单字长指令

R2←(R2)－(R1) ；单字长指令

JNZ 22H ；双字长指令

计算机系统中执行一个程序都有执行准备、执行程序两个环节，执行准备的任务是将程序调入主存、将程序首地址写入PC，执行程序的任务是自动、逐条地取出指令并执行。

（1）执行准备环节的组织

执行准备环节的工作是由操作系统完成的，本实验通过具有相同效果的操作来实现。

对于程序调入主存而言，可以通过在lpm\_dq中设置初始化文件（如Prg1.mif）来实现。假设，测试程序存放在Prg1.mif中地址为20H~24H的存储单元。

为了更好地进行测试，应使测试程序中R0、R1、R2的初值为指定值，这可以通过修改Prg1.mif中相应存储单元的内容来实现，也可以通过从KEY输入来实现，建议采用前者。例如，机器启动时会初始化部件（如清零PC、GPRs），则有(R0)＝(R1)＝(R2)＝0，设置M[00H]＝4、M[04H]＝8时，则前2条指令执行结束时，(R0)＝0、(R1)＝4、(R2)＝8。

对于将程序首地址写入PC而言，可以通过从KEY输入来实现。即在初始化部件之后、执行程序之前，使控制信号KEYout及PCin有效、输入信号KEY＝20H即可实现。

（2）执行程序环节的组织

执行程序的任务是自动、逐条地取出指令并执行，通过执行μOPCmd序列即可实现指令的执行。本实验要求手动给出μOPCmd序列，来代替CU自动产生的μOPCmd序列。

指令执行过程由取指、译码、执行三个阶段组成，译码阶段有一定时延、但没有μOP，本实验假设，译码安排在取指阶段的最后一个μOP中实现，μOP的定时采用同步定时方式（即节拍周期等于时钟周期）。

给出μOPCmd序列时，按照测试程序的指令执行顺序逐条给出，执行每条指令的μOPCmd序列与教材例5.1较为相似。注意，不同指令的μOPCmd序列之间不要留空，尽量模仿真实的程序执行过程。

（3）数据通路的验证

数据通路的验证通过仿真＋分析方法来实现，仿真采用时序仿真方式进行。基于Quartus II进行验证的过程包括：编辑仿真波形文件、开始仿真、分析仿真结果的正确性。

编辑仿真波形文件时，为了反映计算机工作的全过程，给出的μOPCmd序列应包括系统总清（Reset有效）、将程序首地址写入PC、执行测试程序的μOPCmd序列。为了反映JNZ指令执行的两种执行结果，执行测试程序的μOPCmd序列中，可以包含执行到25H单元处指令的所有μOPCmd，即循环执行完毕。

注意，时钟脉冲信号Clk的周期应大于各μOP时延，Clk的上升沿应安排在时钟周期的中部；所有输入信号都应以时钟周期为单位，即值的改变与Clk对齐；μOPCmd序列中不需要增加查看μOP结果的μOPCmd，查看μOP结果可以通过查看BUS[7..0]来实现，或测试程序中下一条指令使用上条指令的结果来实现。

分析仿真结果的正确性时，应以μOP为单位进行分析，查看每次输入信号对应的输出信号是否与预期结果一致。若不一致，则数据通路电路或μOPCmd序列有错误，分析原因、找出错误点、修改电路或μOPCmd序列，直到结果一致为止。注意，一定要预先准备好每个μOP的预期结果，否则会不知道对与错。

### 四、实验要求

（1）做好实验预习。了解数据通路的组成原理及指令执行过程的组织方法，基于Quartus II提供的元器件，画出电路图，标明引脚名，编写测试程序，写出相应的μOPCmd序列。

（2）完成实验内容。实现所设计电路，验证电路正确性，保存仿真波形文件。

（3）撰写实验报告。按所给模板撰写。

（4）上交实验结果及实验报告。本实验做完的下一周，将4个实验所共用的工程文件内容、实验报告（包含4个实验）的电子稿交上来。每组学生交一份，文件夹名称为该组学生的学号及姓名（如301XXX302YYY），实验报告就放在这个文件夹下，工程文件的内容放在下一级文件夹中。