**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，熟练数字电路芯片的使用，提高数字电路的逻辑设计能力，为计算机硬件设计打下基础。

计算机组织与结构的课程实验共包含4个实验，分别是寄存器组设计、ALU设计、存储器设计及总线互连、数据通路组织。

所有实验都要求基于Quartus II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，采用功能（或时序）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，实验四需要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）温习基于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。

（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，实现所设计电路（编辑电路）；通过主菜单File→Save可以保存为原理图文件（如test\_and.bdf）。注意，电路应包括输入引脚、输出引脚，电路编辑包括元器件选择、引脚连接等环节，本例电路包含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。

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，点击该文件右键菜单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test\_and.bdf。

4）编译顶层文件：通过主菜单Processing→Compiler Tool进入编译界面，点击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test\_and.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，进行当前顶层文件的仿真波形文件编辑；选择所需的I/O引脚，设置各输入引脚在不同时刻的信号波形后，通过主菜单File→Save保存为波形文件（如test\_and.vwf）。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（Functional）、时序（Timing）两种方式，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。

3）开始仿真：点击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上升沿时发生变化（触发）、在其余时间保持不变；锁存器的状态在控制信号E为高电平时跟随输入端发生变化、为低电平时保持不变（锁存）。

Quartus II提供了多种类型的触发器，如D触发器、T触发器等。每种类型的触发器有引脚固定、引脚可变（又称参数化）两类，如74173、74273、lpm\_ff、lpm\_dff、lpm\_tff等，参数化触发器的参数不同、型号不同，如lpm\_dff0、lpm\_dff1等，其中序号是自动生成的。

（1）D触发器的功能与引脚



图1为8位D触发器lpm\_dff的I/O引脚示例。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（高电平有效），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（即clock使能）引脚（高电平有效）。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。写操作（enable＝1、aclr＝0、aset＝0）时，data上信号在clock上升沿时写入触发器。

实际应用中，通常会省略不使用的引脚，如aset、sclr、sset等引脚，缺省引脚的信号在芯片内部都处于无效状态。不建议省略enable引脚，以避免q端产生毛刺。

（2）D触发器功能的测试

触发器功能的测试包括电路实现、电路仿真及结果分析几个步骤，仿真时先采用功能仿真方式得到结果并分析，再采用时序仿真方式查看器件操作的时延特征。

电路实现需要编辑原理图文件（如test\_dff.bdf），电路由一个lpm\_dff、若干输入引脚及输出引脚组成，如图2所示。注意，信号线的连接有连线、信号线同名等多种实现方法；信号线经过器件后即可重新命名，器件wire不进行任何操作（可用于重命名）；总线信号线中部分信号线的使用方法与verilog语言相同，如DIin[5]、DIn[2..0]。

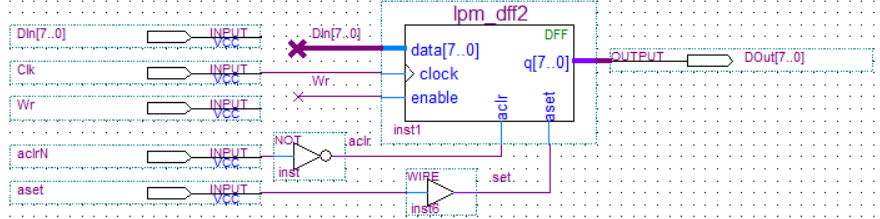


图2 触发器测试电路的组成示例

电路仿真需要先建立仿真波形文件（如test\_dff.vwf），再设置各输入信号在不同时刻的取值，输入/输出信号应包含电路的所有引脚，输入信号的组合应能够反映电路功能表的所有功能。注意，时序逻辑电路中，输入信号的时长应是时钟周期的倍数，初始化信号除外；时钟周期的宽度应大于所有操作的时延，如lpm\_dff时延约5ns，时钟周期可设置为20ns。

实际应用中，时钟周期的开始都用时钟脉冲信号的上升沿来标志，输入信号的长度为时钟周期的倍数、状态变化滞后于时钟脉冲信号，因为许多信号通过时钟脉冲信号产生。为了使功能仿真方式、时序仿真方式的结果相同，要求时钟脉冲信号的上升沿略早于输入信号的改变，如图3所示。实现方法是，设置时钟脉冲信号时，修改Time period参数的offset值。

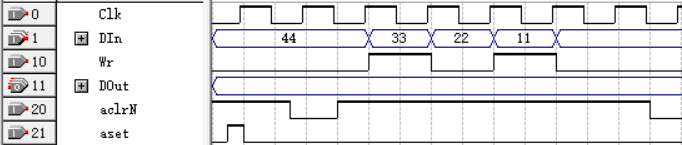


图3 触发器测试电路的信号波形示例

分析仿真结果时，应以时钟周期为单位进行分析，查看每个输出信号的变化与输入信号的关联，分析是否满足电路功能表的要求。

#### 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，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。

（1）寄存器组的设计

由于寄存器组包含4个8位寄存器，故操作时的地址引脚为log24＝2位、数据引脚为8位。由于要求读/写端口分离，寄存器所存信息的输出无需信号控制，故寄存器组可同时进行读、写操作，读操作相关引脚为地址引脚raddr[1..0]、数据输出引脚q[7..0]，写操作相关引脚为地址引脚waddr[1..0]、数据输入引脚data[7..0]、写使能引脚wen、时钟脉冲引脚Clk，清零操作相关引脚为清零引脚Clr。

寄存器组中，写操作通过waddr[1..0]指定目标寄存器，通过wen、Clk实现写入控制（边沿触发），寄存器的选择可使用译码器来实现；读操作通过raddr[1..0]选择目标寄存器，寄存器输出无需信号控制，数据输出的选择可使用选择器来实现；清零操作通过Clr来控制。寄存器组的内部组成如图4所示，假设wen、Clr都是高电平有效。



图4 寄存器组的组成

（2）寄存器组的实现与验证

寄存器组的实现需要编辑原理图文件（如GPRs.bdf），电路有多种实现方法，译码器、选择器可使用Quartus II提供的参数化模块lpm\_mux、lpm\_decode。注意，由字符及数字组成的多个相似信号线命名时，数字位数应相同（如wren01与wren21）；添加lpm\_器件时，应从本工程文件的工具条symbol Tool中选择，或.bdf文件中拷贝，不能从其他工程文件的.bdf文件中拷贝，否则编译时会产生错误（lpm\_器件未定义或定义冲突），错误修改方法是用手工选择的lpm\_器件替换.bdf中所有的lpm\_器件。

寄存器组的验证包括电路仿真、结果分析两个步骤，仿真采用功能仿真方式进行，可以再采用时序仿真查看器件操作的时延特征。

电路仿真需要先建立仿真波形文件（如GPRs.vwf），再设置waddr、raddr、data及各控制信号在不同时刻的取值，输入信号的组合需能够反映电路的所有功能特性。注意，读、写操作的测试次数应≥2，以避免错误现象被隐藏；所有输入信号的时长都应以时钟周期为单位，输入信号的改变应滞后于时钟脉冲信号Clk的上升沿。

分析仿真结果时，应以时钟周期为单位进行分析，查看每个输出信号的状态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）测试加/减法器的功能。

（2）设计具有加法、减法、逻辑与、逻辑非功能的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＝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可以选择加/减运算是无符号运算还是有符号运算，选择无符号运算时overflow结果不正确。

加/减法器功能的测试包括电路实现、电路仿真及结果分析几个步骤，仿真时先采用功能仿真方式得到结果并分析，再采用时序仿真方式查看器件操作的时延特征。

功能测试时，电路实现、电路仿真、结果分析的方法同触发器的功能测试。

注意，lpm\_add\_sub中cin引脚的功能是实现带进位加法或带借位减法，本实验不需要设置该引脚；仿真波形中数据信号的选择应能够产生各种输出结果，如加法运算测试应包含5组数据（2组++、2组--、1组+-），枚举dataA最高位为0及1时overflow、cout的可能组合，减法运算测试同样需要5组数据。

结果分析时，应以操作为单位进行分析，查看每个输出信号是否正确。最后，还应分析出无符号加/减运算的结果溢出条件，即CF的有效逻辑。

#### 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。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，可产生结果状态标志ZF、CF、OF、SF。ALU的功能如表1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

由表1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的内部组成如图5所示。



图5 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是由输出数据F形成的。

（2）ALU的实现与验证

ALU实现需要编辑原理图文件（如ALU.bdf），电路有多种实现方法，加/减法器、与门、非门可分别使用Quartus II提供的参数化模块lpm\_add\_sub、lpm\_and、lpm\_inv。

ALU的验证包括电路仿真、结果分析两个步骤，仿真采用功能仿真方式进行。

电路仿真需要先建立仿真波形文件（如ALU.vwf），再设置所有输入信号在不同时刻的取值，输入信号的组合需能够反映电路的所有功能特性。例如，加法、减法功能验证时，CF测试需包含2×2组数据，OF无需测试（加/减法器测试中已进行过），SF、ZF还需若干数据（可利用已有数据）。

分析仿真结果时，应以操作为单位进行分析，查看每个输入信号对应的输出信号是否正确。最后，还应分析出有/无符号关系运算的结果表示方法。

### 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 存储器设计及总线互连

### 一、实验目的

（1）熟悉RAM的功能及使用方法。

（2）掌握存储器的设计方法。

（3）掌握基于总线的部件互连及操作方法。

### 二、实验内容

（1）测试RAM的功能。

（2）设计一个读/写端口分离的128×8位存储模块，前64B为只读空间，并验证设计正确性。

（3）将所设计存储模块、输入部件、输出部件连接到地址/数据复用的8位总线上，通过输入部件对存储器进行读/写操作、通过输出部件查看结果，并验证设计正确性。

### 三、实验原理及方案

#### 1、RAM功能的测试

Quartus II提供了多种类型的RAM模块，如lpm\_ram\_dq、lpm\_ram\_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64×8位的读/写端口分离的lpm\_ram\_dq模块为例，其I/O引脚示例如图6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（高电平有效）。clken用于允许、禁止clock信号进入芯片内部，因此，clken可以用作芯片的片选信号。

对lpm\_ram\_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存储单元中。读操作时（wren＝0、clken＝1），地址在clock上升沿被锁存，然后指定存储单元的数据经过内部端口q，立即被送到引脚q（端口q不带锁存功能时），或下个时钟周期上升沿被输出到引脚q（端口q带锁存功能时）。注意，设置有读使能引脚rden时，读操作的有效逻辑是rden＝1、clken＝1。

存储器功能的测试放在存储模块设计的验证中进行（存储模块设计太简单）。

#### 2、存储模块的设计、实现及验证

存储模块可对存储器芯片进行容量扩展来实现，容量扩展的方法有位扩展、字扩展、字位扩展3种。本实验要求设计一个读/写端口分离的128×8位存储模块MEM，前64B为只读空间。

（1）存储模块的设计

由设计要求可知，存储模块MEM可对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，可用位扩展方法实现64×8位ROM或64×8位RAM。

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[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ROM、RAM连接时，地址引脚连接A[5..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MUX连接到Q、用A[6]进行选择，ROM的clken＝CS·A[6]、RAM的clken＝CS·A[6]。

（2）存储模块的实现及验证

存储模块的实现需要编辑原理图文件（如Mem.bdf），电路有多种实现方法，ROM、RAM、MUX可使用Quartus II提供的参数化模块lpm\_rom、lpm\_ram\_dq、lpm\_mux，lpm\_rom需要预先写入各存储单元内容（可通过设置其初始化文件[如rom.mif]来实现）。lpm\_rom初始化文件的建立，可通过主菜单File→New→Other Files→Memory Initialization File进入。注意，存储模块MEM应设置片选引脚CS；应使lpm\_rom及lpm\_ram\_dp都不带输出锁存功能，以便于立即输出读操作结果；应使lpm\_rom中拟测试的存储单元的内容各不相同，以防止错误现象被隐藏。

存储模块的验证包括存储器功能测试、存储模块功能验证两个环节，可通过分别访问不同存储器芯片法方法来实现。验证有电路仿真、结果分析两个步骤，仿真时先采用功能仿真方式得到结果，再采用时序仿真方式查看器件操作的时延特征。

电路仿真需要先建立仿真波形文件（如MEM.vwf），在设置所有输入信号在不同时刻的取值，输入信号的组合需能够反映电路的所有功能特性。波形文件中，先使0≤A[6..0]＜63，通过2组数据测试ROM的功能；再使64≤A[6..0]＜128，通过3组数据（写单元a、写单元b、读单元a）测试RAM的功能。注意，所有输入信号的时长都应以时钟周期为单位，输入信号的改变应滞后于时钟脉冲信号clock的上升沿。

分析仿真结果时，应以时钟周期为单位进行分析，查看每个输出信号的状态及时序是否正确。

#### 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，任何部件只有一个输入端或输出端可以直接连接到总线上。

本实验要求将所设计存储模块MEM（128×8位）、输入部件、输出部件连接到地址/数据复用的8位总线上，通过输入部件对MEM进行读/写操作、通过输出部件查看结果。

（1）总线互连的设计

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；由于总线是地址/数据复用总线，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；由于MEM是时序逻辑部件，数据输出信号仅与内部保存的状态有关、与数据输入信号无关，MEM的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，3个部件通过地址/数据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7 部件通过总线互连的连接电路

（2）总线互连的实现与验证

总线互连的实现需要编辑原理图文件（如Bus.bdf），电路有多种实现方法，存储模块MEM需使用实验三设计的电路（MEM.bsf），地址锁存器LA、三态门可使用Quartus II提供的参数化模块lpm\_dff、lpm\_bustri，KEY、CRT可直接使用引脚input、output。注意，LA应该用触发器实现，以便于基于clock进行操作控制（同MEM控制方式）。

总线互连的验证包括电路仿真、结果分析两个步骤，仿真采用功能仿真方式进行。

电路仿真需要先建立仿真波形文件（如Bus.vwf），再在文件中进行写RAM单元b1、读ROM单元c、写RAM单元b2、读RAM单元b1操作即可。注意，每个操作需要2个时钟周期实现（写LA、访存）；所有输入信号的时长都应以时钟周期为单位，输入信号的改变应滞后于时钟脉冲信号clock的上升沿。

分析仿真结果时，应以时钟周期为单位进行分析，查看每个输出信号的状态及时序是否正确。

### 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)、双字长分支(JNZ)指令。

（2）将测试程序存入主存，根据程序执行过程的μOPCmd序列，控制所设计数据通路，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

本实验要求支持Demo\_IS指令系统的取数(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。

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：

1）数据类型只有8位整数一种，采用定点格式（补码编码）表示；

2）数据操作只有8位的加法、减法两种，需产生状态标志ZF；

3）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；

4）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；

5）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；

6）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、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指令系统分析结果可见，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来实现。为了简化控制，假设PC具有计数功能，可用Quartus II提供的lpm\_counter模块来实现。

想使用实验一至实验三的设计结果，就需要先根据原理图文件（.bdf）生成相应的符号文件（.bsf），然后就可以像使用lpm\_dff等器件一样，使用所设计电路了。

（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，数据通路的组成如图8所示，其中的TS0~TS3为三态门，与教材中的图5.7基本相同。为了便于调试，图中增加了输出信号PC、IR及CRT，可以根据需要，自行增加输出信号，要求至少需输出PC、IR、CRT、RS、RD。



图8 单总线结构的数据通路组成

图中的GPRs的读地址引脚前增加了选择器MUX，以满足SUB指令周期中2次读GPRs的需求，当前地址（RS或RD）用控制信号GRsel来选择。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为只读空间）来实现。由于实验三的存储模块未设置读操作控制信号，图8中的控制信号MemRd可以省略（读命令用MemWr表示）。

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，因此，判断ZF＝0也是由人工完成的，故PSR可以省略（用虚线表示）。

（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。

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结束前接收数据并写入。由于μOPCmd由CU在时钟周期开始时（clock上升沿）产生（电路会有一定延迟），因此，μOPCmd在整个时钟周期内都有效，μOPCmd有效时，clock上升沿已经错过了，μOP只能在当前时钟周期结束时（即下个clock上升沿时）写入数据，效果如图3所示。

#### 3、数据通路的验证

本实验要求将测试程序存入主存，根据程序执行过程的μOPCmd序列，控制所设计数据通路，来验证数据通路的正确性。

测试程序需要自行编写，只需包含3种指令即可。测试程序示例如下：①R1←M[(R0)]、②R2←M[(R1)]、③R2←(R2)－(R1)、④JNZ 22H，其中，前3条指令为单字长指令，第4条指令为双字长指令，程序共占5个存储单元。

计算机启动时都会进行硬件初始化，假设图8初始化后，GPRs及PC的内容均为0。计算机中执行一个程序都有准备、执行两个环节。

（1）程序执行的准备

程序执行的准备工作是将程序调入主存、将程序首地址写入PC。该工作原本由操作系统完成，本实验通过将程序预先存入MEM的初始化文件中来实现。

MEM的前64B空间由lpm\_rom实现，其初始化文件（如rom1.mif）中，可存放测试程序及预存数据。由于初始化后(PC)＝0，故测试程序从地址为0的单元开始存放。

为了有效进行测试程序中指令③的测试，应该使(R2)≠(R1)。由于初始化后GPRs的内容都为0，故(R1)＝M[0]，而M[0]的内容为测试程序中指令①的内容（值为24H）。欲使(R2)≠(R1)，在M[24H]单元中存入不同的值（如35H）即可。

（2）程序执行的实现

程序执行的任务是自动、逐条按(PC)取出指令并执行，本实验通过手工输入（代替CU自动产生）各条指令执行过程所需的μOPCmd序列，来实现程序执行的控制。

每条指令的μOPCmd序列，可以根据指令功能、图8的数据通路给出。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，每个μOP时延为1个时钟周期。注意，不同指令的μOPCmd序列之间不要留空档，尽量模仿真实的程序执行过程。

（3）数据通路的验证

数据通路的验证包括电路仿真、结果分析两个步骤，仿真采用功能仿真方式进行。

电路仿真需要先建立仿真波形文件（如DataPath.vwf），再在文件中初始化硬件（Reset有效后无效），然后给出与测试程序相对应的每一条指令周期所需的μOPCmd序列。注意，所有输入信号的时长都应以时钟周期为单位，输入信号的改变应滞后于时钟脉冲信号clock的上升沿

分析仿真结果时，应以μOP（1个时钟周期）为单位进行分析，查看每个输出信号的状态及时序是否正确（与预期结果一致）。若不一致，则数据通路或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，一定要预先写好完整的μOPCmd序列，及每个μOP的预期结果，否则根本不知道对与错。

### 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。班长收齐电子稿后，一起交给老师。