**计算机组成原理实验讲义**

**（2023版）**

**任国林 编写**

**东南大学计算机科学与工程学院**

计算机组成原理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，增强数字电路芯片的使用能力，提高数字逻辑电路的设计能力，为计算机硬件的设计打下基础。

计算机组成原理的课程实验共包含4个实验，分别是寄存器组设计、ALU设计、存储器及总线互连设计、数据通路组织。

所有实验都基于Quartus II进行电路实现和验证，要求采用原理图方式实现电路，采用功能仿真方式进行电路仿真。

实验四需要使用实验一、实验二所设计的电路，四个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。

## 实验一 寄存器组设计

### 一、实验目的

（1）温习基于Quartus II的数字电路实现及仿真方法。

（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）设计并实现具有1个读端口、1个写端口的4×8位寄存器组，验证其正确性。

### 三、实验原理及方案

#### 1、基于Quartus II的电路实现及仿真过程

基于Quartus II，电路实现过程主要包括：建立工程文件、编辑原理图文件、编译原理图文件，一个工程文件中可以包含若干个原理图文件，每个原理图文件都可以单独使用。电路仿真过程主要包括：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）、电路仿真、分析仿真结果。

下面，以测试D触发器功能为例，说明基于Quartus II、采用原理图方式进行电路实现的过程：

1）建立/打开工程文件：通过主菜单File→New Project Wizard或File→Open Project，可以建立/打开工程文件。建立工程文件包含5个页面的设置，本例的工程文件可用作4个实验共用的工程文件。

·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等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或File→Open，可以创建/打开原理图文件，并进入原理图编辑器，编辑所设计的电路；通过主菜单File→Save，可以保存当前原理图文件（如test\_dff.bdf）。本例电路包含5个输入引脚、1个输出引脚，内部逻辑为1个D触发器，如图1所示。

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，即可设置该文件为顶层文件。本例选择的文件名为test\_dff.bdf。

4）编译顶层文件：通过主菜单Processing→Compiler Tool，可进入编译界面，点击Start开始编译。编译时，提示窗口中将显示编译相关信息，包括警告及错误信息。编译出错时，需修改原理图文件，并重新编译；编译成功后，可进入仿真阶段。

下面，以原理图文件test\_dff.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或File→Open，可以建立/打开仿真波形文件，并进入仿真波形文件编辑器，编辑当前顶层文件所对应的仿真波形文件；通过主菜单File→Save，保存为仿真波形文件（如test\_dff.vwf）。注意，一个原理图文件可以对应多个波形文件，仿真时进行选择即可，这样做提高测试效率；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，原理图一旦修改，都需要重新进行编译、生成功能网表。

3）电路仿真：选择所选的仿真波形文件，点击Start开始进行仿真，通过Report或VCD文件可以查看仿真结果波形。

4）分析仿真结果：对照所仿真电路的功能表，分析电路的功能是否正确。

Quartus II可以将电路封装为器件（符号文件），以方便该电路的使用（选择器件）。封装方法是用原理图文件（如abc.bdf）生成符号文件（如abc.bsf）具体方法是，打开原理图文件，通过主菜单File→Create/Update→Create Symbol Files for Current File，可以生成该原理图文件中电路所对应的符号文件（建议与原理图文件同名）。

注意，原理图文件的输入/输出引脚有改变时，需要重新生成符号文件。

上述介绍仅为总体过程，具体操作步骤参见Quartus II的使用指南。

#### 2、D触发器的功能

触发器的触发方式有边沿触发、电位触发两种，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信号CP上升沿时发生改变（触发）、在其余时间保持不变；锁存器的状态在控制信号E为低电平时保持不变（锁存）、为高电平时随输入端信号而变化。

实际应用中，为了实现精准控制，通常使用的是触发器，而不是锁存器。

图1为8位D触发器的I/O引脚示例。其中，data[7..0]、q[7..0]分别为数据输入、输出引脚，clock为时钟引脚，enable为clock使能引脚（控制clock信号能/否到达器件内部、高电平有效），aclr、aset分别为异步清零、异步置位引脚（高电平有效）。

图1所示8位D触发器的功能如表1所示，操作有清零、置位、写入3种类型，×表示信号取值不影响功能实现。可见，读出操作可通过保持功能实现，仅写入操作与时钟信号clock有关。

表1 D触发器功能表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| aclr | aset | enable | clock | data[7..0] | q[7..0] | 功能说明 |
| 1 | 0 | × | × | × | 0…0 | 异步清零（与clock无关） |
| 0 | 1 | × | × | × | 1…1 | 异步置位（与clock无关） |
| 0 | 0 | 1 |  | *x* | *x* | 写入（与clock有关） |
| 0 | 0 | 1 |  | × | *y* | 保持（*y*为所存储信息） |
| 0 | 0 | 0 | × | × | *y* |

#### 3、寄存器组的组成

寄存器组由多个寄存器组成，按地址进行操作，操作类型有读出、写入两种。读出、写入操作是否可以同时进行，取决于读、写端口的地址引脚是否独立。

本实验要求设计4×8位寄存器组，其内部组成如图2所示。其中，写入控制由译码器实现，所写寄存器由waddr（2位）指定，是/否写由wen控制（是/否译码）；所读寄存器由raddr（2位）指定。



图2 寄存器组的组成

可见，寄存器组的功能表如下：写入功能为REG[waddr]＝data，在wen＝1、Clk上升沿时实现；读出功能为q＝REG[raddr]，可随时实现；清零功能为REG[\*]＝0，\*表示所有寄存器，可随时实现。

### 四、实验步骤及要求

#### 1、D触发器的功能测试

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多种型号，如74173、74273、lpm\_ff、lpm\_dff、lpm\_tff等，其中lpm\_器件的参数可以设置。lpm\_触发器的型号随参数不同而不同，如lpm\_dff0、lpm\_dff1等，其序号是自动生成的。

D触发器器件要求选用由lpm\_dff模块实现的8位D触发器，以学习lpm\_器件的使用方法，其I/O引脚的设置可如图1所示。

注意，lpm\_器件可缺省部分引脚（如图1中缺省sclr和sset），其信号处于无效状态。强烈反对省略enable引脚、用Wr·clock（Wr为写控制信号）连接到clock引脚的做法，这样会导致q端产生毛刺。

触发器的功能测试包括电路实现、电路仿真及结果分析三个步骤。

电路实现时，先编辑原理图文件（如test\_dff.bdf），如图3所示；再编译原理图文件。

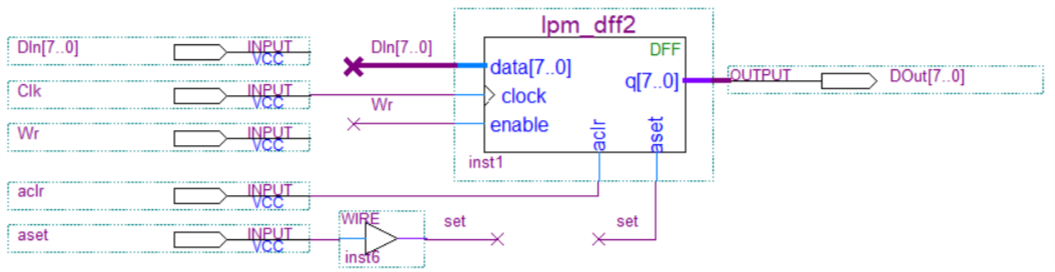


图3 触发器测试电路的组成示例

注意，信号线的连接方法有连线、信号线同名等方法；信号线经过器件后即可重新命名，器件WIRE只用于重命名（不进行任何操作）；总线中部分信号线的使用方法与verilog语言相同，如DIin[5]、DIn[2..0]，单根信号线不要写成DIin[5..5]。

电路仿真时，先建立仿真波形文件（如test\_dff.vwf），设置各输入信号线在不同时刻的信号取值，如图4所示；再采用功能仿真方式进行电路功能仿真。

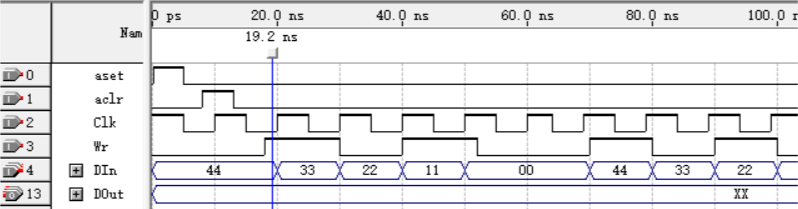


图4 触发器测试电路的信号波形示例

注意，仿真波形文件中的输入/输出信号线应包含电路的所有引脚；输入信号的取值组合须能够反映电路功能表的所有功能，如图4中15ns～55ns间的Wr、Din、Clk信号组合对写入结果的影响；时钟周期的开始都用时钟信号的上升沿来标志。

注意，当Wr在Clk上升沿时开始有效（如图4的40ns处），功能仿真方式是在时钟周期开始时（当前Clk上升沿）完成写操作（40ns时），而时序仿真方式是在时钟周期结束时（下个Clk上升沿）完成写操作（50ns时）。

本课程所有实验对寄存器写操作的要求是：寄存器写操作在时钟周期结束时完成。如图4中68ns～78ns为一个时钟周期，写操作在第78ns完成。为此，编辑.vwf文件时，要么将时钟信号（如Clk）的上升沿稍微提前，要么将其他输入信号（如Din、Wr）的状态改变稍微滞后。很显然，将时钟信号上升沿提前、其余信号的改变与标尺同步是理想的选择（只调整1个信号）。改变时钟信号的具体方法是，修改其Time period参数的offset值，例如周期为10ns时，offset选择4ns。

分析仿真结果时，应以时钟周期为单位进行分析，查看每个输入信号（含时钟信号）的变化对输出信号的影响，分析其是否与电路功能表一致。

#### 2、寄存器组的实现及验证

（1）寄存器组的实现

本实验要求的寄存器组为4×8位寄存器，具有1个读端口、1个写端口。有实验原理可见，所设计电路的结构与图2基本相同。

电路实现时，先编辑原理图文件（如GPRs.bdf），再编译原理图文件。电路文件中，译码器、选择器可选用Quartus II提供的参数化模块lpm\_mux、lpm\_decode。

注意，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），错误修改方法是重新建立/选择lpm\_器件，来替换出错的lpm\_器件。

（2）寄存器组的验证

寄存器组的验证包括电路仿真、结果分析两个步骤，要求寄存器写入在时钟周期结束时完成。

电路仿真时，需要先建立仿真波形文件（如GPRs.vwf），再进行电路仿真。波形文件中，所有输入信号的取值组合需能够反映电路的所有功能特性。

注意，读操作、写操作的测试次数应≥2次，以避免错误现象被隐藏；所有输入信号的时长应以时钟周期为单位，时钟信号Clk的上升沿应略早于输入信号的改变。

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。

#### 3、其他要求

（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）保存实验电路。由于实验四可以使用前三个实验所设计的电路，为了避免重新设计并实现电路，所有实验应共用同一个工程文件，所有文件放在同一个文件夹下。

（3）撰写实验报告。按所给模板及时撰写，等四个实验做完后一起上交（电子稿），4个实验报告放在同一个文件中。

## 实验二 ALU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）设计并实现具有加法、减法、逻辑与、逻辑非功能的8位ALU，算术运算产生标志ZF、CF、OF、SF，逻辑运算仅产生ZF标志，验证其正确性。

### 三、实验原理及方案

#### 1、加减法器的功能

加减法器可以实现有/无符号定点数的加法、减法运算，能够产生运算结果、结果标志，结果标志用来表明运算结果是否正确。

有/无符号加减法的运算方法完全相同，但结果溢出判断方法不同，因此，加减法器只有一组数据输出引脚，分别设置结果标志（或其组成参数）引脚。图5是8位加减法器的I/O引脚示例。其中，dataa及datab、out为数据输入、输出引脚，add\_sub为操作控制引脚（高电平表示加法），cin为内部加法器的最低位进位引脚，cout为内部加法器的最高位进位引脚（高电平表示有进位），overflow为有符号运算结果溢出引脚（高电平表示溢出）。

注意，cout不是CF，CF的形成逻辑可放在加减法器外部完成，CF＝cout⊕(~add\_sub)。

图5所示加减法器的功能为：加法out＝dataa＋datab＋cin，减法out＝dataa－datab－cin。

#### 2、ALU的组成

ALU能够实现多种算术运算、逻辑运算功能，能够产生运算结果标志，结果标志包括ZF、CF、OF、SF。

本实验要求的ALU具有加法、减法、逻辑与、逻辑非运算功能，算术运算产生结果标志ZF、CF、OF、SF，逻辑运算仅产生ZF标志。ALU的功能表约定可如表2所示，其中，A、B为数据入端，F为数据出端（8位），SEL为功能选择（操作控制）信号（2位），不产生的结果标志均应输出0。

表2 ALU功能表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能选择 | 实现功能 | | | |
| SEL | 操作 | 助记符 | 功能函数 | 产生的结果标志 |
| 0 | 加法 | ADD | F＝A＋B | ZF、SF、CF、OF |
| 1 | 减法 | SUB | F＝A－B | ZF、SF、CF、OF |
| 2 | 逻辑与 | AND | F＝A·B | ZF |
| 3 | 逻辑非 | NOT | F＝ | ZF |

基于表2的功能约定，ALU的内部组成可如图6所示。其中，加法、减法运算用加减法器来实现，逻辑与、逻辑非运算可用与门、非门来实现，当前操作结果的输出可用选择器来实现，当前操作结果的标志用门电路产生。



图6 ALU的组成

### 四、实验步骤及要求

#### 1、加减法器的功能测试

加减法器器件建议选用Quartus II提供的lpm\_add\_sub模块，其I/O引脚的设置可如图5所示，建议缺省cin引脚。

加减法器的功能测试包括电路实现、电路仿真及结果分析三个步骤，具体方法同D触发器的功能测试。

电路实现时，建议缺省lpm\_add\_sub的cin引脚，感兴趣的同学可以试试该引脚的功能。

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不是CF；有些Quartus II版本的lpm\_add\_sub可以选择加减运算是无符号/有符号运算，应选择有符号运算（否则overflow结果会出错）。

电路仿真时，输入数据的组织应能够产生各种输出结果（含标志），如加法运算至少有5组数据（2组正正、2组负负、1组正负），以枚举overflow、cout的组合，减法运算同样至少有5组数据（2组正负、2组负正、1组负负或正正。

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。ZF、CF的有效逻辑分析可以放在ALU中进行。

#### 2、ALU的实现及验证

（1）ALU的实现

本实验要求的ALU具有加法、减法、逻辑与、逻辑非运算功能，算术运算产生结果标志ZF、CF、OF、SF，逻辑运算仅产生ZF标志。由实验原理可见，所设计电路的结构与图6基本相同。

电路实现时，先编辑原理图文件（如ALU.bdf），再编译原理图文件。电路文件中，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。

（2）ALU的验证

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。

电路仿真时，先建立仿真波形文件（如ALU.vwf），再进行电路仿真。波形文件中，所有输入信号的取值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，ZF测试需增加一组数据，OF测试能够产生就行（器件已测试）。

注意，同一组数据可同时测试CF、OF、SF、ZF标志，以减少测试数据集大小。

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。

#### 3、其他要求

（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）保存实验电路。原因同实验一。

（3）撰写实验报告。按所给模板及时撰写，等四个实验做完后一起上交。

## 实验三 存储器及总线互连设计

### 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）掌握部件与总线连接的方法。

### 二、实验内容

（1）测试RAM的功能。

（2）设计并实现单向数据引脚的128×8位存储模块，存储模块的前64B为只读空间，验证其正确性。

（3）将上述存储模块连接到地址线/数据线复用的8位总线上，通过总线对该存储模块进行操作。

### 三、实验原理及方案

#### 1、RAM的功能

RAM有SRAM、DRAM两种类型，其工作方式有异步、同步两种方式。异步方式的RAM需要设置状态引脚（如mfc）来表示当前操作是否完成，同步方式的RAM需要设置时钟引脚（如clock）来约定下一步操作何时开始。

图7为单向数据引脚的64×8位同步SRAM的I/O引脚示例。其中，data、q为数据输入引脚、输出引脚，wren为写使能引脚（高电平有效），clock为时钟引脚，clken为clock使能引脚（高电平有效）。可见，clken可以用作SRAM的片选引脚。

图7所示SRAM的功能表如表3所示，操作有写入、读出两种类型，×表示信号取值不影响功能实现。

表3 同步SRAM功能表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| clken | wren | clock | address | data | q | 功能说明 |
| 1 | 1 |  | *a* | *x* | Mem[*a*] | 写入（Mem[*a*]＝*x*） |
| 1 | 0 |  | *a* | × | Mem[*a*] | 读出 |
| 1 | × |  | *a* | × | *y* | 保持（*y*为上次操作的输出） |
| 0 | × | × | × | × | *y* |

由表3可以看出，读出、写入操作均与时钟信号有关。在clken＝1、clock上升沿到来时，address、data引脚上的信息被锁存（记为*a*和*x*）；然后，若wren＝1则信息*x*被写入到存储单元*a*中，若wren＝0则存储单元*a*中内容被送到q引脚。

#### 2、存储器的逻辑设计

存储器设计可以通过对RAM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。

本实验要求设计单向数据引脚的128×8位存储模块，存储模块的前64B为只读空间。因此，存储模块可使用1片64×8位ROM、1片64×8位SRAM，通过字扩展来实现。

存储模块内部的芯片连接很简单，进行字扩展时，ROM及SRAM的地址引脚、数据输入引脚、控制引脚连线相同，仅片选引脚、数据输出引脚的连线有所不同。其中，片选引脚通过地址高位进行判断，数据输出引脚通过选择器进行选择（如图6所示）。

#### 3、存储器与总线的连接

总线由地址线、数据线、控制线组成，地址线/数据线可以复用，控制线可以缺省。地址线/数据线复用时，地址、数据需分时传送；控制线缺省时，各部件的操作控制信号需单独给出。部件与总线连接时，其出端需通过三态门连接到总线，其入端及出端只有一个可以直接连接总线（其余通过锁存器/触发器连接）。

本实验要求将存储模块（记为MEM）连接到地址线/数据线复用的8位总线上，通过总线对该存储模块进行操作。为了避免涉及总线传输协议，假设总线不包含控制信号线，各部件的操作控制信号单独给出。

由于MEM的数据引脚是单向的，故MEM的数据输出引脚Q需通过三态门连接到总线。由于总线的地址线/数据线是复用的，故MEM的数据输入引脚D、地址引脚A只有一组可以直接连接到总线，假设D直接连接总线，A通过锁存器连接到总线，Q与总线间无需设置锁存器（Q与D非组合逻辑关系）。

可见，存储器与总线的连接方法如图8所示。其中，LA用于锁存地址，KEY、CRT用于信息输入、结果查看，实验中可用输入、输出引脚代替；MA和MDO为调试引脚，用于查看MEM的操作特性。

由于8位总线可寻址空间为256个，而MEM地址引脚为7位（128个存储单元），假设MEM占用总线地址空间的低端（0～127），因此，MEM连接总线时，片选信号CS的有效逻辑需通过电路（如图8中电路C）实现，即CS＝~LA[7]。



图8 部件与总线的连接电路

### 四、实验步骤及要求

实验内容中要求先测试RAM的功能。由于存储模块的组成很简单，故RAM的功能测试可以放在存储模块的验证中一起进行。

#### 1、存储模块的实现及验证

本实验要求实现单向数据引脚的128×8位存储模块，存储模块的前64B为只读空间。因此，存储模块可由64×8位ROM、64×8位SRAM进行字扩展而构成。

（1）SRAM/ROM的器件选择

Quartus II提供了多种类型的RAM模块，如lpm\_ram\_dq、lpm\_ram\_dp、lpm\_rom等，这些模块都为同步方式的SRAM或ROM，即读/写操作在时钟信号的上升沿开始；读操作可以选择是否需要使用信号控制；读操作的数据输出可以选择是/否带锁存功能。

SRAM、ROM建议选用Quartus II提供的lpm\_ram\_dq、lpm\_rom模块，其I/O引脚的设置可如图7所示，ROM没有data、wren引脚。

注意，lpm\_ram\_dq、lpm\_rom模块中，存储单元与输出引脚q之间设置有输出缓冲器，输出缓冲器可以设定为无锁存或带锁存功能，默认带锁存功能。输出缓冲器无锁存功能时，所读数据在时钟周期开始后被送到引脚q（读周期＜1 clock），当前时钟周期结束时可将数据写入寄存器；输出缓冲器带锁存功能时，所读数据在时钟周期结束时被送到引脚q（读周期＝1 clock），下个时钟周期结束时才可将数据写入寄存器。

为了减少错误概率，要求一个时钟周期内读存储器并写入寄存器，因此，应取消lpm\_ram\_dq默认的输出锁存功能。

（2）存储模块的实现

存储模块实现时，先编辑原理图文件（如Mem.bdf），再编译原理图文件。电路文件中，ROM、SRAM、MUX可使用Quartus II提供的参数化模块lpm\_rom、lpm\_ram\_dq、lpm\_mux，ROM中相关存储单元需预先存放好数据。

lpm\_ram\_dq、lpm\_rom模块的I/O引脚的设置如图7所示，lpm\_rom没有data、wren引脚。lpm\_rom预先存放数据可以通过设置其初始化文件[如rom.mif]来实现，lpm\_rom初始化文件的创建，则可通过主菜单File→New→Other Files→Memory Initialization File进入后实现。

为了便于测试SRAM的功能，建议在Mem.bdf中增设调试引脚，直接输出lpm\_ram\_dq的数据输出引脚q。

注意，存储模块应设置片选引脚CS，以便被用作器件；lpm\_rom及lpm\_ram\_dp都应不带输出锁存功能，以实现读周期＜1个时钟周期；lpm\_rom中拟测试单元的内容应各不相同，以防止错误现象被隐藏。

（3）存储模块的验证

存储模块的验证需要完成RAM功能测试、存储模块功能验证，RAM功能测试可以通过对存储模块中的lpm\_ram\_dp多次操作来实现，存储模块功能验证则是对不同模块操作来实现。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。

电路仿真时，先建立仿真波形文件（如MEM.vwf），再进行电路仿真。波形文件中，所有输入信号在不同时刻的取值组合需能够反映电路的所有功能特性。例如，先使0≤地址＜63，通过2组数据测试ROM的功能；再使64≤地址＜128，通过3组数据（写单元a、写单元b、读单元a）测试SRAM的功能。

注意，所有输入信号的时长都应以时钟周期为单位，时钟信号clock的上升沿应先于其他输入信号的状态改变。

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。通过增设的lpm\_ram\_dq调试引脚，查看RAM的功能测试是否正常。

#### 2、存储器与总线连接的实现与验证

（1）存储器与总线连接的实现

本实验要求将存储模块（记为MEM）连接到地址线/数据线复用的8位总线上，通过总线对该存储模块进行操作。由实验原理可见，所设计电路的结构与图8基本相同；电路的I/O引脚中，KEY、CRT、MA、MDO等均为8位，MEM的地址引脚为7位；电路C的逻辑为CS＝~LA[7]。

电路实现时，先编辑原理图文件（如Bus.bdf），再编译原理图文件。电路文件中，存储模块MEM为本实验已设计的电路，使用前需先生成其符号文件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来实现。

注意，LA应该用触发器实现，以便基于clock进行操作控制（与MEM的操作控制方式相同）。

（2）存储器与总线连接的验证

存储器连接总线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。

电路仿真时，先建立仿真波形文件（如Bus.vwf），再进行电路仿真。波形文件中，只需实现写单元b1、读单元c、写单元b2、读单元b1操作即可。为了考查MEM中ROM、SRAM连接的正确性，可使得b1、b2的值≥64#，c的值＜64#。

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，时钟信号clock的上升沿应先于其他输入信号的状态改变。

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、其他要求

（1）做好实验预习。了解RAM的组成原理、基于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）保存实验电路。原因同实验一。

（3）撰写实验报告。按所给模板及时撰写，等四个实验做完后一起上交。

## 实验四 数据通路组织

### 一、实验目的

（1）理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）设计并实现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）编写测试程序并存入存储器，根据所组织的μOPCmd序列控制程序执行过程，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

本实验要求支持教材中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←Mem[(RS)]、RD←(RD)－(RS)、ZF＝0时PC←Addr否则PC←(PC)+2。其中，RD、RS表示GPR编号，(Rx)、Mem[(Rx)]表示GPR、存储单元的内容，Addr为直接寻址方式的地址码，ZF为状态寄存器PSR中的结果标志。

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，采用定点格式表示；

2）数据操作只有8位的加法、减法两种，需产生结果标志ZF；

3）数据寻址方式有寄存器寻址、寄存器间接寻址两种，数据地址无需计算；

4）指令寻址方式有直接寻址、隐含寻址两种，地址计算方法为8位加法；

5）GPR有4个，长度为8位，每条指令最多有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、数据通路的组织

数据通路组织包含功能部件设计、部件互连设计两个环节，本实验要求采用单总线结构实现部件互连。

（1）功能部件设计

数据通路部件由取指、执行部件组成，分析部件（ID）包含在CU中。取指部件应包括PC、IR，为了满足3条目标指令的要求，执行部件应包括ALU、GPRs、PSR，以及MAR、MDR、存储器MEM。

由Demo\_IS的分析结果可见，ALU应具有加法、减法功能，需产生结果标志ZF；GPRs由4个8位寄存器组成，应具有1个读端口（总线互连时）和1个写端口；MEM容量应≤256×8位；PC可具有计数功能，便于单字长指令隐含寻址的实现。

（2）部件互连设计

部件互连采用单总线结构实现时，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。

根据功能部件设计结果，本实验的数据通路组成如图9所示。其中，Y、Z为ALU连接总线所需的锁存器，TS0~TS3为部件出端连接总线所需的三态门，MUX为SUB指令中2次读GPRs所需的地址选择器，所读地址来自IR中的RS或RD。



图9 单总线结构的数据通路组成

由于Quartus II提供的MEM都是单向数据引脚、同步工作方式的SRAM，故MDR应由MDRI及MDRO组成；由于实验无需支持存数指令，故MDRI可以省略，控制信号MDRin及MemWr也可缺省（图中用虚线表示）；MDRO的写入控制需要用控制信号实现，本实验可借用MemRd来实现（MEM读周期＜1个时钟周期）。

PSR用于存放结果标志ZF，供CU使用，由于本实验的μOPCmd产生由手工实现（仿真波形文件的输入引脚），故判断ZF＝0也可由人工完成，故PSR可以省略。

注意，硬件启动后，各GPR的值应该是随机的，使用时先写后读。为便于测试程序得到确定的结果，实验时GPRs预先清零（与PC初始化同时进行）。这仅仅是简化实验的变通方法，实际计算机中并非这样。

#### 3、数据通路的验证

本实验要求通过执行程序来验证数据通路是否正确，整个验证过程包括测试程序编写、程序执行准备、程序执行实现3个阶段。

（1）测试程序的编写

测试程序的目标为验证所支持指令的执行结果是否正确。因此，测试程序需包含3种指令，例如：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。

（2）程序执行的准备

程序执行准备的任务是将程序调入主存、将程序入口地址写入PC。该工作原本由操作系统完成，本实验通过将测试程序预先写入MEM、初始化PC来实现。

测试程序预先写入MEM，可通过设置其初始化文件（如rom.mif）来实现；初始化PC需由信号触发（如图9中的Reset），初始化结果为：PC＝常数。

注意，测试程序在MEM中的入口位置，必须与PC初始化结果相同。

假设初始化后PC＝0、各GPR＝0，则测试程序应从MEM的0#单元开始存放。可见，Mem[0]＝24H（指令①的指令字），指令②的源操作数为Mem[24H]；若想指令④的执行结果为转移，则应有(R2)≠(R1)，即Mem[24H]≠24H（如35H）。

注意，测试程序所需的MEM数据也可放在MEM的初始化文件中。

（3）程序执行的实现

程序执行实现的任务是自动、逐条按(PC)取出指令并执行。本实验通过手工给出指令执行过程所需的μOPCmd序列，来实现程序的执行控制；且假设译码不占用单独的节拍（如取指阶段最后一个μOP中实现）。

### 四、实验步骤及要求

#### 1、数据通路的实现

本实验要求实现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。由实验原理可见，所设计电路的结构与图9基本相同，可以缺省的部件应尽量缺省；电路I/O引脚中，地址、数据均为8位，应增设调试信号PC、IR、RS、RD及CRT。

电路实现时，先编辑原理图文件（如DataPath.bdf），再编译原理图文件。

电路文件中，GPRs、ALU可使用实验一、实验二的设计结果；MEM可选用Quartus II提供的lpm\_rom模块（不建议使用实验三的设计结果），容量设置为256×8位，这样既可缺省片选电路，又可避免实验三的设计错误；PC可选用Quartus II提供的lpm\_counter模块。IR、MAR、MDRO可用Quartus II提供的lpm\_dff模块来实现。

时序逻辑部件连接时，时钟引脚连接到节拍（时钟周期）的哪一个工作脉冲（中部、结束时），取决于部件是/否为μOP的目的部件。如图9中，PC、IR、GPRs、Y、Z、MAR、MDRO的写入应安排在时钟周期结束时，而MEM的读出应安排在时钟周期中部开始。

注意，所有部件的控制信号都需设置为输入引脚，避免因门电路而产生毛刺。

#### 2、数据通路的验证

本实验的数据通路验证包括测试程序准备、电路仿真、结果分析三个步骤，仿真采用功能仿真方式进行。

测试程序准备时，需要设置MEM的初始化文件[如prg.mif]。初始化文件的内容包括所编写的测试程序、测试所需的MEM数据，它们的内容、存放位置等不再重复叙述。

电路仿真时，先建立仿真波形文件（如DataPath.vwf），再进行电路仿真。波形文件中，首先进行硬件初始化（使Reset有效后无效），然后给出测试程序中每条指令执行所需的μOPCmd序列。

注意，所有输入信号的时长都应以时钟周期为单位，时钟信号的上升沿应略早于输入信号的改变，不同指令执行过程应连续（尽量模拟实际执行过程）。

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，要预先写好完整的μOPCmd序列，及每个μOP的预期结果，否则分析时根本不知道对与错。

#### 3、其他要求

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

（2）撰写实验报告。按所给模板及时撰写。

（3）上交实验结果及实验报告的电子稿。需提交的文件包括4个实验所共用的工程文件的压缩文件、4个实验所共用的实验报告文件，文件名称为COEXP-09021302XXX格式（09021302XXX为学号和姓名）。班长收齐电子稿后，一起交给老师。