**计算机组成原理实验讲义**

**（2020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、存储器设计及总线互连、数据通路组织。

所有实验都基于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，采用功能仿真（或时序仿真）方式进行电路仿真，条件允许时基于FPGA芯片进行电路验证。本实验讲义基于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所设计的电路，因此，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）、开始仿真、分析仿真结果的正确性。

下面，以测试D触发器功能为例，说明使用Quartus II、采用原理图方式进行电路设计的过程：

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。注意，该工程文件可用作4个实验共用的工程文件。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）。注意，电路须包含输入引脚、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5个输入引脚、1个输出引脚，内部逻辑为1个D触发器。

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。

注意，Quartus II可以将所设计电路封装为器件（如abc.bsf），以后即可通过选择器件(abc.bsf)来使用所设计电路了，封装方法是用原理图文件abc.bdf生成符号文件abc.bsf。生成符号文件的方法是，打开原理图文件（原理图编辑器窗口显示的文件），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，可以编辑符号文件名（建议同名），点击保存即可。注意，只要.bdf文件的输入/输出引脚有改变，就要重新生成符号文件。

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，以提高测试效率；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，修改原理图后，开始仿真前需要重新进行编译、生成功能网表。

3）开始仿真：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。为了实现精准控制，通常使用触发器，而不使用锁存器。

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，触发器的功能、引脚可以选配（参数化）。

（1）D触发器的功能与引脚

8位D触发器lpm\_dff的I/O引脚示例如图1所示。其中，data[7..0]、q[7..0]分别为数据输入、数据输出引脚，aclr、aset分别为异步清零、异步置位引脚（高电平有效），同步清零、同步置位引脚为sclr、sset（图中未选配），clock为时钟脉冲引脚，enable为clock使能引脚（控制clock信号能否进入芯片内部、高电平有效）。

该8位D触发器的功能表如表1所示，操作类型有清零、置位、写入、读出4种，其中，只有写操作与时钟脉冲信号clock有关。

表1 D触发器功能表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| aclr | Asset | 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* | 读出（*y*为所存储信息） |
| 0 | 0 | 0 | × | × | *Y* |

实际应用中，通常会省略暂不使用的引脚，如aset、sclr、sset等引脚，缺省引脚的信号默认处于无效状态。不建议省略enable引脚，以避免操作时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。

电路实现时，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）；总线信号线中一部分信号线的使用方法与verilog语言相同，如DIin[5]、DIn[2..0]。

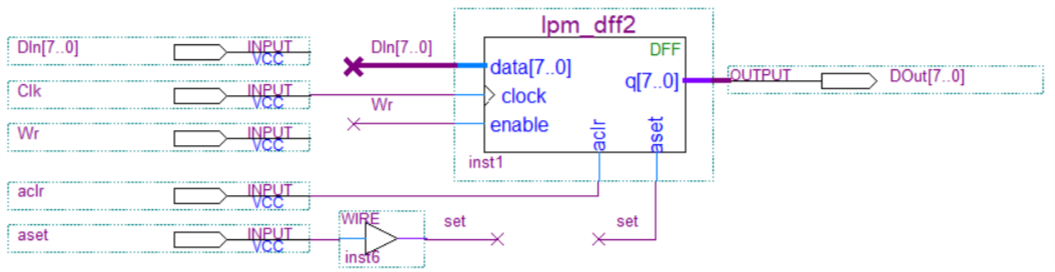


图2 触发器测试电路的组成示例

电路仿真时，需要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，再采用功能仿真方式进行电路功能仿真。注意，输入/输出信号线包含电路的所有引脚；输入信号的组合须能够反映电路功能表的所有功能，及Wr、Din、Clk信号的时序对写操作结果的影响（如图3的15ns～55ns间信号所示）；时钟周期的开始都用时钟脉冲信号的上升沿来标志。

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（含时钟脉冲信号）的变化对输出信号的影响，分析是否与电路功能表一致。注意，当Wr在Clk上升沿时开始有效（如图3的40ns处），功能仿真方式在时钟周期开始时（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）。

本课程所有实验的要求：电路仿真采用功能仿真方式，寄存器写操作在时钟周期结束时完成。因为，实际应用中的控制信号通常在时钟周期开始时由电路产生（有一定时延），因此，控制信号的状态变化会滞后于时钟脉冲信号，操作效果应与时序仿真方式相同。同理，数据信号也会滞后于时钟脉冲信号。

为了实现上述要求，编辑.vwf文件时，要么将时钟脉冲信号上升沿稍微提前，要么将其他输入信号的状态改变稍微滞后，很显然，将时钟脉冲信号上升沿提前是理想的选择（只需改变1个信号），如图3的70ns～100ns间信号所示，Din、Wr等信号的改变都可以与标尺同步。具体方法是，在设置时钟脉冲信号参数时，修改Time period参数的offset值，例如周期为10ns时，offset选择4ns。

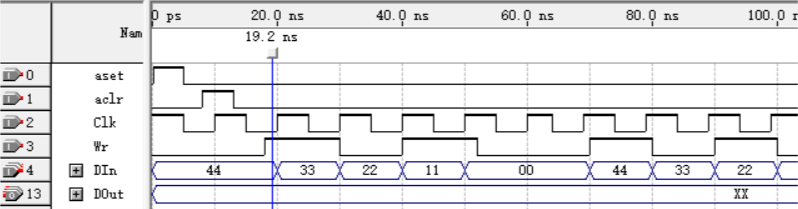


图3 触发器测试电路的信号波形示例

#### 3、寄存器组的设计、实现及验证

寄存器组由多个寄存器组成，按地址进行操作，以简化控制复杂度。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，故地址引脚为log24＝2位、数据引脚为8位。由于读/写端口分离，故寄存器组可以同时进行读、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。

设计寄存器组的内部逻辑之前，应先约定好寄存器组的功能表。例如，写操作通过waddr[1..0]指定目标寄存器，通过data[7..0]指定所写数据，通过wen、Clk实现写入控制（边沿触发）；读操作通过raddr[1..0]选择目标寄存器，通过wen无效实现读出控制；清零操作通过Clr来控制。

寄存器组的内部组成如图4所示，其中，译码器用于所写寄存器的选择，wen控制译码器是否译码（不译码时输出全为0），选择器用于所读寄存器的选择。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，添加lpm\_器件时，应从本工程文件的工具条symbol Tool中选择，或从已有.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），所有.bdf文件中的同名lpm\_器件都会出错，错误修改方法是从工具条symbol Tool中建立/选择lpm\_器件，来替换出错的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，要求达到时序仿真方式的操作结果（在时钟周期结束时写入）。

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，读操作、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（以备重新进行仿真）。

（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，节省实验所花时间，所有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。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需产生结果状态标志ZF、CF、OF、SF，并验证设计正确性。

### 三、实验原理及方案

ALU的加减运算由加减法器实现，加减法器的核心为加法器，产生的结果状态标志可以用于实现关系运算。

#### 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为内部加法器的最高位进位，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，选择为无符号运算时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。电路实现、电路仿真、结果分析的方法同D触发器功能的测试。

电路实现时，应缺省lpm\_add\_sub的cin引脚，该引脚可用于实现带进位加法（cin=1时）和带借位减法（cin=0时），该引脚缺省时，加法时默认cin=0（不带进位），减法时默认cin=1（不带借位）。

电路仿真时，数据信号的组织应能够产生各种输出结果，如加法运算至少测试5组数据（2组++、2组--、1组+-），以枚举dataA最高位为0及1时的overflow、cout的可能组合，减法运算同样至少测试5组数据（2组+-、2组-+、1组--或++）。

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。本实验要求的ALU，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的功能表如表2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，不受影响的结果标志输出均为0。可见，SEL为2位（记为SEL[1..0]），F的位数与A及B相同。

表2 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 | 影响的结果标志 |
| 0 | 0 | 加法 | ADD | F＝A＋B | ZF、SF、CF、OF |
| 0 | 1 | 减法 | SUB | F＝A－B | ZF、SF、CF、OF |
| 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2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的内部组成如图5所示。



图5 ALU的组成

其中，控制信号形成电路负责产生各模块所需的操作控制信号，状态形成电路负责产生4个结果状态标志，本实验中只有加减法器、状态形成电路需要使用控制信号。由于逻辑运算的结果影响ZF的取值，故ZF由输出数据F形成。

（2）ALU的实现与验证

ALU的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。

电路仿真时，需要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，加法、减法功能验证时，CF测试需包含2×2组数据，OF测试需包含2×2组数据，SF测试可包含在CF、OF测试中，ZF测试还需若干数据（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）设计一个电路，将上述存储模块连接到地址线/数据线复用的8位总线上，并能够通过总线对该存储器进行操作、查看操作结果。

### 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使用信号控制；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）。lpm\_ram\_dq是同步RAM，clken用于控制clock信号能否进入芯片内部，因此，clken可以用作lpm\_ram\_dq芯片的片选信号。

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立即被送到引脚q（输出缓冲器无锁存功能时），或下个时钟周期上升沿被输出到引脚q（输出缓冲器带锁存功能时）。注意，当设置读使能引脚rden时，读、写操作的有效逻辑都与rden有关；读周期默认为2个clock（即输出缓冲器带锁存功能），若希望读周期为1个clock时，应取消默认的输出锁存功能。

存储器功能的测试放在存储模块设计的验证中进行（存储模块的设计太简单）。

#### 2、存储模块的设计、实现及验证

存储模块可通过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（记为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，可尝试用64×4位ROM或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），则连接内部芯片时，ROM及RAM的地址、clock引脚分别连接MEM的A[5..0]、clock引脚，ROM及RAM的数据输出引脚通过选择器MUX连接到MEM的Q引脚（MUX用A[6]控制），RAM的数据输入、wren引脚直接连接MEM的D、wren引脚，ROM的clken＝CS·A[6]、RAM的clken＝CS·A[6]。

（2）存储模块的实现及验证

存储模块实现时，需编辑原理图文件（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实现，其中lpm\_rom需预先写入各存储单元内容（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进入后实现。

注意，存储模块MEM应设置片选引脚CS，以便于被用作器件；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。

存储模块验证时，需实现存储器功能测试、存储模块功能验证，存储模块功能验证可通过分别访问不同存储器芯片来实现，存储器功能测试可通过测试lpm\_ram\_dp的功能表来实现。

电路仿真时，需先建立仿真波形文件（如MEM.vwf），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。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，时钟脉冲信号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、部件基于总线互连的设计、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。为了防止部件的不同输入端之间、输入端与输出端之间的信号干扰，输入端及输出端只有一个可以直接连接到总线，其余连接到总线之前需要设置锁存器。

本实验要求将所设计存储模块MEM（128×8位）连接到地址线/数据线复用的8位总线上，通过总线对MEM进行操作、查看操作结果。

可见，8位总线可寻址空间为256个，而MEM只有128个存储单元，假设MEM安排在总线地址空间的低端（0～127），总线不包含操作控制信号。

（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；由于总线是地址线/数据线复用总线，MEM的数据输入引脚D、地址引脚A只有一组可直接连接到总线，假设D直接连接总线，则A需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无关）。

为了能够通过总线进行MEM操作、查看操作结果，电路需要设置输入引脚、输出引脚可以理解为电路连接了输入部件、输出部件。输入部件也需要通过三态门连接到总线，输出部件可直接连接总线。

MEM通过总线互连的原理图如图7所示，其中，电路C为MEM片选信号CS的有效逻辑电路（操作地址为8位、MEM地址仅7位），所需的操作控制信号也已标出（时钟脉冲信号除外）。



图7 部件通过总线互连的连接电路

为了进一步查看存储器MEM的操作特性，可以在电路中增加调试引脚MA和MDO，如图7所示。

（2）总线互连的实现与验证

电路实现时，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，以便于基于clock进行操作控制（与MEM的操作控制方式相同）。

电路仿真时，需要先建立仿真波形文件（如Bus.vwf），实现写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，时钟脉冲信号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。

### 四、实验要求

（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）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实现原理。

### 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）编写测试程序并存入存储器，给出程序执行过程的μOPCmd序列，来验证数据通路设计的正确性。

### 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，数据地址无需计算；

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、专用结构（有称点点结构）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中，取指部件应包括程序计数器PC、指令寄存器IR，执行部件应包括ALU、寄存器组GPRs、状态寄存器PSR、存储器MEM，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，可使用实验二所设计的ALU来实现；GPRs应包含4个8位寄存器，总线互连时只需具有1个读端口和1个写端口，可使用实验一所设计的寄存器组来实现；MEM的容量应≤256×8位，可使用lpm\_rom芯片来实现。由于机器字长等于存储字长（均为8位），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。IR、MAR、MDR可用Quartus II提供的lpm\_dff模块来实现。

要在电路中使用实验一至实验二的设计结果，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，数据通路的组成如图8所示，与教材中的图5.7基本相同。图中，Y、Z为ALU连接总线所需的锁存器，TS0~TS3为信号输出到总线所需的三态门，MUX为SUB指令中2次读GPRs所需的地址选择器，当前读地址（RS或RD）来自于IR。



图8 单总线结构的数据通路组成

由于MEM的读/写端口分离，故MDR由MDRI及MDRO实现；由于MEM只需要支持读出功能，故MDRI、控制信号MDRin及MemWr都可以省略（用虚线表示）；由于MEM可在一个时钟周期内完成读操作，故可用控制信号MemRd来实现MDRO的写入控制（同步控制方式）。

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也可由人工完成，PSR可以省略（用虚线表示）。同理，ID也可省略。

为了便于调试，要求至少需输出PC、IR、RS、RD及总线信号CRT的内容。

（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。

数据通路中，数据传送μOP应采用电位-脉冲制来实现，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（时延为图9中①），控制Rx输出端对应的三态门打开（时延为图9中②），数据传送到Ry后（时延≤图9中③），写入通常在时钟周期结束时（图9中④[即下个clock上升沿]）进行，有时也可在时钟周期中部（clock下降沿）进行。

因此，时序逻辑部件连接时，PC、IR、GPRs、Y、Z、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，操作需求包括GPRs←MEM，故MEM的读操作应安排在时钟周期中部（clock下降沿）开始。

#### 3、数据通路的验证

本实验要求编写测试程序并存入存储器，组织程序执行过程的μOPCmd序列，来验证数据通路及指令执行过程的正确性。

测试程序只需包含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，本实验通过将测试程序预先存入MEM的初始化文件、硬件初始化（PC＝测试程序首地址）来实现。

MEM的初始化文件（如rom.mif），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0#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)，以便于测试JNZ指令。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，需要根据指令功能约定、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。

电路仿真时，需要先建立仿真波形文件（如DataPath.vwf），首先实现硬件初始化（Reset有效后无效），然后给出与测试程序相对应的每一个指令周期所需的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，否则分析时根本不知道对与错。

### 四、实验要求

（1）做好实验预习。了解数据通路的组成原理、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，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。将4个实验所共用的工程文件、实验报告（包含4个实验）的电子稿交上来。每个组只交一份，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。