**计算机组织与结构**

**实验讲义**

**（2018版）**

**任国林 编写**

**东南大学计算机科学与工程学院**

计算机组织与结构课程实验是为巩固教学效果而设置的，学生可以通过这些实验掌握计算机部件的逻辑组成及其工作原理，熟悉数字电路芯片的使用方法，提高逻辑设计能力，为设计计算机模型机打下基础。

计算机组织与结构课程实验共有4个，分别是寄存器组成实验、运算器组成实验、存储器组成实验、CPU数据通路实验。

所有实验均基于EDA工具Quartus II（或ISE）进行设计和功能仿真，条件允许时基于FPGA芯片进行功能验证。实验讲义基于Quartus II撰写。

**实验一 寄存器组成实验**

**一、实验目的**

（1）熟悉D触发器的功能及使用方法。

（2）掌握寄存器组的逻辑组成及使用方法。

**二、实验内容**

（1）掌握Quartus II的使用方法，能够进行数字电路的设计及仿真。

（2）验证Quartus II所提供D触发器的功能及使用方法。

（3）设计具有1个读端口、1个写端口的寄存器组，并进行读写操作仿真/验证。

**三、实验原理及方案**

**1、基于Quartus II的电路设计及仿真方法**

基于Quartus II，电路设计的主要过程包括：建立工程文件、编辑原理图文件、编译原理图文件，电路功能仿真。电路功能仿真的主要过程包括：编辑仿真波形文件、生成功能仿真网表、验证仿真波形正确性。

以功能为Z=X·Y的电路为例，使用Quartus II、采用原理图方法进行电路设计的过程如下：

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

·第1页为设置工程文件信息，包括工程文件名、工作目录名

·第2页为在工程中加入文件，可将已有的GDF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，本例中采用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可进入原理图编辑器，进行电路原理图编辑，通过主菜单File→Save保存为原理图文件。注意，原理图应包括输入引脚、输出引脚，电路编辑包括选择元器件、引脚连接，本例中电原理图有2个输入引脚、1个输出引脚，电路逻辑为1个2输入端的与门，假设文件名为C\_AND.bdf。

3）设置顶层文件：在Project Navigator窗口的File页面中，选择需编译或仿真的电原理图文件，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C\_AND.bdf。

4）编译工程文件：即编译顶层文件，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。编译完成后，即可进入功能仿真阶段。

以原理图C\_AND.bdf为例，使用Quartus II进行电路功能仿真的过程如下：

1）编辑仿真波形文件：通过主菜单File→New→Other Files→Vector Waveform File可进入波形文件编辑器，进行当前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，一个电原理图文件可以有多个波形文件，仿真时允许进行选择。

2）生成功能仿真网表：仿真有功能仿真、时序仿真两种，该步骤仅在功能仿真时使用。注意，原理图修改后，仿真前需重新进行编译、生成功能网表。

3）验证仿真波形正确性：按Start开始仿真后，通过Report显示仿真波形，对照电路功能真值表，查看所设计电路功能是否正确。

注意，原理图文件被修改后，必须重新编译，且重新生成功能仿真网表。

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触发器的功能及操作**

触发器对时钟脉冲边沿（上升沿或下降沿）敏感，状态在边沿到来后变化；锁存器对时钟脉冲电平（持续时间）敏感，状态在电平持续期间都变化。

Quartus II提供了多种类型的触发器模块，如D触发器、T触发器等。固定特性的触发器模块有不同的型号，参数化的触发器模块有lpm\_ff、lpm\_dff、lpm\_tff等。D触发器常用于构建寄存器。

以8位D触发器lpm\_dff（或lpm\_ff）为例，其I/O引脚如图1所示，部分引脚可以缺省。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（复位）、置位引脚，同步的清零、置位引脚为sclr、sset，clock为时钟脉冲信号引脚，enable为写使能引脚。



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[7..0]立即变为全0；异步置位（aclr=0、aset=1）时，q[7..0]立即变为全1；同步清零、同步置位时，q[7..0]在clock上升沿时变为全0、全1。写操作（enable=1）时，data[7..0]上数据在clock上升沿时写入触发器、经一定延迟后输出到q[7..0]引脚。注意，写操作时应有aclr=aset=0，同步方式、异步方式的清零/置位操作的状态变化时机不同。

Quartus II可以省略芯片不使用的引脚，如sclr、sset等引脚。但不建议省略enable引脚，以避免毛刺的产生。

触发器操作的设计与验证过程与功能为Z=X·Y的电路实验基本相同。需要注意的是，应进行触发器功能表的所有功能的仿真。

**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，以简化控制复杂程度。

本实验的寄存器组要求包含8个8位寄存器，具有1个读端口、1个写端口。因此，操作地址均为3位，数据均为8位。由于要求读写端口分离，因此，读操作的相关引脚有地址raddr[2..0]、数据输出q[7..0]，写操作的相关引脚有地址waddr[2..0]、数据输入data[7..0]、写使能wen。其中，省略读使能信号可以简化控制，即数据输出不受限制。

寄存器组通过写地址waddr[2..0]、写使能wen信号来实现触发器的写入控制，通过读地址raddr[2..0]信号来控制触发器的数据输出选择。其组成原理如图2所示。



图2 寄存器组的组成

本实验中，寄存器组的实现要求通过Quartus II的原理图方法进行，设计过程包括：建立/打开工程文件、编辑原理图文件、编译原理图文件。注意，原理图的功能实现有多种方法，原理图文件编译前需先设置顶层文件；多个信号线的名称由字符及数字组成、仅数字不同时，数字部分的长度应相同。

本实验中，寄存器组的验证可以通过Quartus II仿真实现，或通过FPGA电路实现。基于Quartus II，仿真过程包括：编辑仿真波形文件、生成功能仿真网表、验证仿真波形正确性。通过FPGA电路实现仿真时，验证过程包括：分配电路的I/O管脚、下载电路到FPGA中、连接FPGA引脚到开关/指示灯、根据输入信号观察输出结果。注意，仿真/验证时，输入信号波形应能够反映电路功能的所有特征，如清零、写入等特征。

基于Quartus II的实现及仿真的具体操作步骤请参见相关指南。

**四、实验要求**

（1）做好实验预习。了解寄存器组的逻辑组成原理，基于Quartus II提供的元器件，画出具体的电路图。

（2）完成实验内容。实现所设计电路，并进行各种操作的仿真，保存好仿真波形图。

（3）撰写实验报告。按“实验报告规范”书写实验报告，需对结果进行分析，总结心得体会与收获，提出改进建议。

**实验二 运算器组成实验**

**一、实验目的**

（1）熟悉加/减法器的功能及使用方法。

（2）掌握算术逻辑部件(ALU)的功能及其逻辑组成。

（3）加深对运算器工作原理的理解。

**二、实验内容**

（1）验证Quartus II所提供加/减法器的功能及使用方法。

（2）设计具有加法、减法、逻辑与、逻辑非4种功能的ALU，并进行功能仿真/验证。

**三、实验原理及方案**

运算器是计算机硬件进行数据加工的重要部件，包括核心部件算术逻辑部件ALU，以及存放操作数和运算结果的寄存器/锁存器、存放运算结果状态的触发器等器件。根据给定控制信号的不同，运算器可以实现不同的运算功能。

**1、ALU的逻辑组成**

ALU能够实现多种算术运算、逻辑运算功能，其功能种类由指令系统决定。

ALU的外部引脚通常由两组数据入端A和B、一组数据出端F、一组功能选择信号SEL、几个结果状态信号组成，数据宽度通常有8位、16位等。由于F＝function(A, B)，故ALU又常称为函数功能发生器，其功能选择信号SEL决定了所实现的功能。

本实验所要求的ALU数据宽度为8位、具有4种算术及逻辑运算功能，其功能选择引脚记为SEL[1..0]。该ALU实现的具体功能如表1所示，其中，算术运算可以实现有符号数、无符号数的运算功能。

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 |
| 0 | 0 | 加法 | ADD | F＝A＋B |
| 0 | 1 | 减法 | SUB | F＝A－B |
| 1 | 0 | 逻辑与 | AND | F＝A·B |
| 1 | 1 | 逻辑非 | NOT | F＝ |

ALU的加法、减法功能可由Quartus II提供的lpm\_add\_sub模块实现，逻辑与、逻辑非功能可由Quartus II提供的lpm\_and、lpm\_inv模块实现。Quartus II中，lpm\_开头的模块均为参数化模块，使用时可以自定义其功能参数，如lpm\_add\_sub是一种参数化加/减法器，可以定义其实现功能、数据宽度、结果状态等。

ALU引脚与内部模块引脚之间的关联，可以通过门电路、数据选择器、编码/译码器等实现。如各功能模块的数据输出，与ALU数据出端F的连接，可通过多路数据选择器实现。相应地，Quartus II提供了参数化的数据选择器lpm\_mux、译码器lpm\_decode。

为了降低硬件实现成本，比较操作通常通过减法操作、逻辑操作的组合实现，例如，判断X＜Y可通过X－Y、判断结果是否有借位或是否为负数实现，判断X＝Y可通过X－Y、判断结果是否为零实现。由于减法与逻辑判断是2个操作，因此，运算器应提供算术运算结果的状态，如零标志、进位/借位标志、溢出标志、结果符号标志。

本实验所要求ALU的逻辑结构如图3所示，控制形成电路负责产生各功能模块的控制信号，本实验中与门、非门无需控制是特例；状态形成电路负责产生4个状态标志，分别是零标志ZF、进位/借位标志CF、溢出标志OF、结果符号标志SF。



图3 ALU逻辑结构

**2、运算器的设计、实现及验证**

本实验的运算器仅包含支持加法、减法、逻辑与、逻辑非4种功能的8位ALU，能够产生ZF、CF、OF、SF标志位，无需包含暂存结果的锁存器或寄存器组。

本实验中，运算器的实现要求通过Quartus II的原理图方法进行，实现过程包括：建立/打开工程文件、编辑原理图文件、编译原理图文件。注意，原理图的功能实现有多种方法，原理图文件编译前需先设置顶层文件。

本实验中，运算器的验证可以通过Quartus II仿真实现，或通过FPGA电路实现。基于Quartus II，仿真过程包括：编辑仿真波形文件、生成功能仿真网表、验证仿真波形正确性。通过FPGA电路实现仿真时，验证过程包括：分配电路的I/O管脚、下载电路到FPGA中、连接FPGA引脚到开关/指示灯、根据输入信号观察输出结果。注意，仿真/验证时，输入信号波形应能够反映电路功能的所有特征，如运算溢出、结果符号等特征。

基于Quartus II的设计及仿真的具体操作步骤请参见相关指南。

**四、实验要求**

（1）做好实验预习。了解ALU的逻辑组成原理，基于Quartus II提供的元器件，画出具体的电路图。

（2）完成实验内容。实现所设计电路，并进行ALU各种功能仿真，保存仿真波形。

（3）撰写实验报告。按“实验报告规范”书写实验报告，需对结果进行分析，总结心得体会与收获，提出改进建议。

**实验三 存储器组成实验**

**一、实验目的**

（1）熟悉半导体存储器的存取方法。

（2）掌握存储器的扩展方法。

（3）掌握存储器与总线的连接方法。

**二、实验内容**

（1）验证Quartus II所提供半导体存储器的功能及使用方法。

（2）设计一个读/写端口分离的256×8bit的存储器，地址空间中前一半为只读、后一半为可读可写，并进行存取操作仿真/验证。

（3）将上述存储器连接到地址/数据复用的总线上，并进行存取操作仿真/验证。

**三、实验原理及方案**

**1、半导体存储器的功能及操作**

Quartus II提供了多种类型的半导体存储器模块，如lpm\_dq、lpm\_dp、lpm\_rom等，这些模块均均为同步存储器，可以自定义存储字长、存储字数。本实验中要求存储字长为8位即可。

以8位字长、256个存储单元、读/写端口分离的存储器lpm\_dq为例，其I/O引脚如图4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为同步操作的时钟脉冲信号。



图4 lpm\_dq引脚示例

对lpm\_dq的操作有读、写两种。写操作（wren=1）时，数据在clock上升沿锁存并写入到存储单元中，地址也由clock上升沿锁存。读操作（wren=0）时，地址由clock上升沿锁存，数据在clock上升沿后一定时间输出到引脚（功能仿真时延迟可忽略、实时仿真时延迟为可见）。

存储器操作的设计与验证过程与上述实验基本相同。

**2、存储器扩展的设计、实现及验证**

本实验的存储器扩展要求ROM与RAM一起进行字扩展，两者地址空间相同，均为128个单元，地址引脚为address[6..0]，数据引脚为q[7..0]。

存储器扩展的原理图设计很简单，设扩展后存储器的地址引脚为A[7..0]、数据输出引脚为Q[7..0]，则A[7]用于选中ROM或RAM，即控制ROM或RAM的clock时钟信号产生、哪个q[7..0]连接到Q[7..0]。

存储器扩展的验证可以通过Quartus II仿真实现，或通过FPGA电路实现。不同功能电路的Quartus II仿真过程基本相同，具体操作步骤请参见Quartus II相关指南。

**3、存储器与总线连接的实现与验证**

本实验中存储器lpm\_dq连接的总线要求是地址/数据复用总线，即存储器的地址引脚、数据引脚连接到同一组总线上。而存储器lpm\_dq是读/写端口分离的，即数据输入引脚、数据输出引脚是不同的引脚。因此，本实验中存储器lpm\_dq的address[7..0]、data[7..0]、q[7..0]引脚同时连接到同一组总线bus[7..0]上。

总线操作的基本要求是，同时只允许一个设备向总线发送信息，允许多个设备从总线接收信息。因此，本实验中，存储器lpm\_dq的地址信号、数据输入信号、数据输出信号须分时与总线交互。

存储器lpm\_dq与地址/数据复用总线连接时，地址信号、数据输入信号的分时写入可通过增设地址锁存器实现，输入信号、输出信号的分时传送可通过增设三态门实现；同样地，输入部件、存储器的分时传送可通过增设三态门实现。其连接电路原理如图5所示。



图5 lpm\_dq与地址/数据复用总线的连接

存储器与地址/数据复用总线连接的原理图设计中，输入部件可用输入引脚实现，锁存器可采用lpm\_latch模块、三态门可采用lpm\_bustri模块。

存储器与地址/数据复用总线连接的验证过程与上述实验基本相同。

**四、实验要求**

（1）做好实验预习。了解RAM的逻辑组成原理，基于Quartus II提供的元器件，画出具体的电路图。

（2）完成实验内容。实现所设计电路，进行RAM的读、写操作仿真，保存波形文件。

（3）撰写实验报告。按“实验报告规范”书写实验报告，需对结果进行分析，总结心得体会与收获，提出改进建议。

**实验四 CPU数据通路实验**

**一、实验目的**

（1）掌握CPU数据通路的逻辑组成。

（2）了解指令功能的实现过程及其控制方法。

**二、实验内容**

（1）设计一个单总线结构的CPU数据通路，部件包括4种功能的8位ALU、4×8位的寄存器组、256×8位的RAM、8位计数器各一个。

（2）给出相关部件控制信号，分别实现取数、加法、条件转移指令的功能。

**三、实验原理及方案**

1、CPU数据通路的设计

CPU数据通路指指令执行过程中数据所经过的部件。数据通路的结构有总线通路、专用通路两种，总线通路指所有部件间有一条或几条公共的路径来传递数据，故又可分为单总线通路、双总线通路等，专用通路指相关部件间均有一条独立的通道来传递数据，是总线通路的极端情况。

单总线结构的数据通路中，所有部件通过同一个总线传递数据，即所有部件的数据入端（简称入端）、数据出端（简称出端）均连接在同一个总线上。为保证数据传输的正确性，单总线通路要求同时只能有一个部件输出数据、可以有多个部件接收数据，多个数据传递操作必须串行实现。

单总线结构中，要求所有部件的数据入端、数据出端均连接到同一总线；总线操作时，要求同时只能有一个部件输出数据。因此，通路设计时，应该可以控制所有部件出端的数据是否（或者只有一个）送上总线，即控制所有部件出端与总线间的通断（或者选择部件出端），实现方法通常有三态门、多路选择器两种。

本实验要求的是设计单总线结构数据通路，数据宽度为8位，包括ALU、寄存器组、RAM、计数器4个功能部件。各部件的功能分别是，ALU支持加法、减法、逻辑与、逻辑非4种操作（同实验一），寄存器组用作通用寄存器组（记为REGs），由4个具有独立读端口、写端口的寄存器组成（同实验三）；RAM用作Cache，地址空间为8位、读/写端口分离（同实验二）；计数器用作程序计数器（记为PC），具有置数、计数功能。

本实验的数据通路可以采用如图6所示的方案，其中，三态门（记为TSL）是依据总线操作特性（同时只有一个部件能发送数据）而设置的；锁存器是为解决部件的多个端口在单总线上数据接收冲突而设置的；REG\_S为状态寄存器，存放关系运算所需的标志位（如ZF）；其它部件与总线的数据输入、数据输出接口记为BUS\_I、BUS\_O，输入、输出分开便于观察实验结果。



图6 数据通路的逻辑组成

由于数据通路是单总线结构，而ALU为组合逻辑部件，因此，需增设锁存器Y，解决ALU 的两个入端间的输入冲突；需增设锁存器Z，解决ALU的入端-出端间的环路冲突。同理，需增设锁存器A，解决RAM的地址-数据引脚间的输入冲突。

图6中，功能部件的控制信号名称已标出，辅助部件的控制信号名称未标出。其中，ALU的操作控制信号线(2根)为op[2..0]，REGs的写地址信号线(2根)、读地址信号线(2根)分别为WA[2..0]、RA[2..0]，REGs、RAM的写操作控制信号线分别为r\_wr、m\_wr，计数器的置数控制信号线、时钟信号线分别为ldn、cp。

补充说明一下，计数器的数据引脚有入端D、出端Q，控制引脚有置数LDN（低电平有效)、时钟CP、同步/异步清零CLRN（低电平有效)等。计数器的真值表如下：CLRN=0时，Q=0（清零）；CLRN=1、LDN=0、CP上升沿时，Q=D（置数）；CLRN=1、LDN=1、CP上升沿时，Q=Q＋1（计数）；其余情况Q保持不变。

对于部件→总线的数据输出控制，可以通过三态门实现（如图6），无部件输出时总线信号为高阻状态，多个三态门可通过译码器来控制；也可以通过多路选择器实现，总线信号为某部件输出端状态，可自行设计相关电路代替图6中的三态门。

2、指令功能的实现

本实验要求的是实现取数、加法、条件转移指令的功能，各指令功能如下：

取数LD -- RD←M[(RS)]

加法ADD -- RD←(RD)＋(RS)及RD←(RD)＋M[(RS)]

条件转移JZ -- ZF=1时，PC←(PC)＋disp；ZF=0时，PC←(PC)＋1

其中，RD、RS表示寄存器编号，(Rx)、[(Rx)]表示寄存器寻址、寄存器间接寻址方式，(PC)＋disp表示相对寻址方式（disp为偏移量）。可见，ADD指令的源操作数支持两种寻址方式，JZ指令的PC值依据状态寄存器中ZF位的状态形成。

（1）指令格式参数的产生

我们知道，CPU中指令执行过程由取指令、执行指令两个阶段组成，取指令阶段的任务是实现IR←M[(PC)]、PC←(PC)＋1，执行指令阶段的任务是实现指令约定的操作功能。

理论上，指令格式的参数OPER(操作类型)、RS、RD、disp均来自IR，ZF来自状态寄存器；本实验中，为简化实验难度，指令格式的参数通过引脚输入产生，如参数OPER用引脚表示（如op[2..0]、m\_wr、r\_wr、ldn等），RS、RD用引脚RS[2..0]、RD[2..0]产生，disp用引脚BUS\_I[8..0]产生，ZF状态用引脚ZF产生。

需要注意的是，指令操作的参数与部件的引脚不一定都是一一对应的，如RA[2..0]可能来自RS或RD，WA[2..0]只能来自RD。

（2）指令功能的实现

基于数据通路，每个指令的功能可表示为一个uOP序列，序列中每一步的uOP可同时实现。只要按序实现某uOP序列中各uOP的功能，即可实现该指令的功能。

uOP的功能可通过控制相关部件（给出控制信号）来实现，如锁存器Z→R1的数据传送uOP控制信号如下：三态门2控制信号=1(有效)、WA[2..0]=01、r\_wr=1(有效)。因此，根据指令功能（uOP序列）的需要，按序控制相关部件的操作，即可实现指令功能。

指令功能→uOP序列的分解，需要基于数据通路，使用CPU的4种基本操作实现。CPU的4种基本操作分别为寄存器间数据传送、ALU运算、MEM读、MEM写，基于单总线通路，第1种基本操作可用一个uOP实现，后3种均需用二个uOP实现。

至此，根据取数LD、加法ADD、条件转移JZ指令的功能约定，可以分别写出指令执行时的uOP序列。

指令功能的实现可以通过Quartus II仿真完成，即使用所设计的单总线数据通路电路，为每条指令建立一个波形文件，在波形文件中，给出与该指令uOP序列相对应的部件操作控制信号序列，并观察仿真结果是否与该指令约定功能一致。

若仿真结果与指令功能不一致，说明数据通路电路设计、或部件操作控制信号序列有错误，须反复检查、修改相关电路或控制信号，直到结果相一致。注意，应及时为部件准备好原始数据，如向REGs写入数据、向RAM写入数据（可通过初始化文件实现）。

不同波形文件的Quartus II仿真过程基本相同，具体步骤请参见Quartus II使用指南。

**四、实验要求**

（1）做好实验预习。了解CPU数据通路的逻辑组成原理，基于Quartus II提供的元器件，画出具体的电路图。基于所画电路图，分析三种指令功能的实现过程，写出具体的uOP命令序列。

（2）完成实验内容。实现所设计电路，仿真三种指令的功能，保存仿真波形。

（3）撰写实验报告。按“实验报告规范”书写实验报告，需对结果进行分析，总结心得体会与收获，提出改进建议。