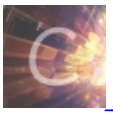
山东大学计算机组成原理整机实验

原创

PancrasBohemian 于 2017-05-11 16:49:27 发布 6179 收藏 18

分类专栏： 助教笔记

本文链接：https://blog.csdn.net/PancrasBohemian/article/details/71641686



更新了关于扩展寻址方式的思路

首先要了解基本的ALU的使用、ROM内容的读取、RAM内容的读写。完成这些之后才能开始整机实验。本实验报告分为两部分，一为初级整机的设计，二为扩展功能设计。如果照ppt上面的电路图画的话，后期的扩展将会受限，因此，如果想做比较炫酷的扩展，要从一开始就打算好。本文只介绍基础版的整机试验，能够帮助大家拿到及格分，若是还想做乘除法、中断、增加寻址方式的话，自己加油咯。

整机设计

一、硬件的选择

对于寄存器组的设置，我猜用了4个八位寄存器，分别用作通用寄存器R0，R1，指令寄存器IR，地址寄存器MAR。对于程序计数器PC，我没有采用带清零端的寄存器，而是与uPC一样，使用了八位计数器。
加法器的选择就是实验四中使用的加法器。
选择器使用的是八位二选一，其中，左侧的选择器（A），控制着RAM读出的数据和R0寄存器的数据；右侧的寄存器（B），控制着R1寄存器的数据和程序计数器PC的数据。
这次的数据通路要求非常明确，是以总线为基础，以CPU为核心构成的，因而所有的指令需经过ALU，随后通过总线进入到相应的器件。如果你想直接使用之前学长学姐的电路图的话是没办法通过实验的。
后继微地址形成部件我没有单独做一个器件，而是用八位二选一，这样的话，在做增加寻址方式的时候，IR的低四位就能派上用场了。
除上述器件外，我还使用了二四译码器、三八译码器。
有一点需要格外注意，在ppt中提供的电路图里，RAM的读端和写端是分开的，但是在我们Quartus平台中，这两段是合并在一起的。因此需要在总线与RAM数据线连接的地方增加一个三态门。三态门的高阻态与控制RAM写信号的uIR输入端相连。

二、控制方式

当我们的电路图基本连接好之后，我们需要对其测试。测试的方法就是通过指令来看看整个电路图是否能够完成我们预想的功能。我们要想让器件正常运转，需要在微指令中设计好控制信号，如CPR0、CPPC、MA（A选择器允许RAM中数据通过）、RB（B选择器允许寄存器R1通过）。因为我们采取的是微程序方式控制，而微程序的执行方式微增量方式。

三、微指令字段定义

在基础的电路图中，微指令字长为16位。微命令的形成逻辑仰仗这些器件间的配合。首先用uIR0-uIR2，连接到一个三八译码器上控制微地址的形成方式。uIR4控制RAM的写信号，uIR5控制RAM的读信号。uIR9-uIR11，连接到一个三八译码器，配合脉冲信号控制寄存器、PC、IR等寄存器或者计数器。uIR12-uIR13用于控制选择器B，uIR14-uIR15用于控制A选择器。如果你没有设置好uIR14，可以将ROM2的内容移到ROM3。

四、后继微地址产生逻辑

在简单模式下，后继微地址的生成方式一共有三种：增量方式(uPC+1)、无条件转移方式(JP)、按操作码转移方式(QJP)。其控制方式在微指令字段定义中有详细描述。如果你想做寻址方式的扩展，需要在后继微地址形成部件处做文章。

五、指令流程

我们需要清楚指令到底是干什么的。在整机试验中，指令包含微程序的首地址、操作数、操作数的地址（扩展后，原本都是立即数寻址）等。每个控制微程序首地址的指令的前四位都是不一样的。在ppt中，这些东西都是被写死了的，MOV1就是把数移动到R0里，MOV2就是把数移动到R1里，其实这是不好的，如果做扩展的话，可以从这里出发，把MOV1和MOV2合并为LOAD指令，并用指令去控制将RAM里的数据移动相应的寄存器里。

六、微代码

微代码在ppt中描述的很详细，不过有一点要注意到，我们的取值指令需要更改一下。在ppt中微地址为13、23、31、42的地方，微操作都是PC->MAR，然后下一步JP跳回到00地址，我们没必要这样，同时应老师要求，应该把这些PC->MAR移动到00地址处。如果你用计数器代替寄存器表示PC的话，那么PC+1->PC的指令可以由17XX01改为06XX01（没有用uIR14所在的ROM2）。

扩展部分

乘除法是极好的，寻址方式也不错。寻址方式的话就是从行成微地址的地方入手。因为我们MOV1只用了10-13的微地址，且是立即数寻址，而剩下的都没用，因此，其它寻址方式可以写在14-1F中，如果微程序长度合理，可以写四种寻址方式（0-F一共16位，16/4=4种）。
下面我将介绍基址寻址的扩展实现
我们首先回忆一下关于指令的特征。指令可以分为好几部分，在整机试验中，前四位为操作码，后四位给我们留了一部分发挥空间，如果你觉得不够用的话，可以采取双倍字长的实现方法，即采用两个八位寄存器当作IR，这也是扩展的一个亮点。
我们可以用第三四位作为寻址特征的编码，这样我们可以设计四种不同的寻址方式（离优秀进近了一步）。
我们来看基址寻址的一条指令（MOV1的基址寻址，再强调，`区分开MOV1和MOV2不是好的选择`，但为了简便起见，我们还是这么做了）：

操作码	扩展特征	偏移量
0001	01	11

上面这条指令的意思是，MOV1采用基址寻址，将专用寄存器BR里的基地址加上偏移量11（也就是3）。
可以看一下课本第314页关于基址寻址的相关描述。我们这里采用的是专用寄存器寻址，这需要你再增加一个寄存器BR，并给其相应的控制信号。如果不想这么做的话，用R1当作专用寄存器吧（这样其实是不对的，因为R1其实是一个通用寄存器，不过这样的话，我们还需要几位来控制寄存器的选择，我们的指令太短了，没法很好的实现，还是简单起见，这样做吧）。
这个01代表基址寻址，位于指令的第三位、第四位（IR3、IR4），这两位可以与一个二四译码器相连，译码的结果可以与一个自制的简易选择器相连，输出的结果与微地址形成部件（八位二选一中的IR的低四位，之前为0000或者空接）相连，在基址寻址中值是0100，这样再配合操作码的1000，我们形成了微地址0001 0100（14），这样我们就跳到了微地址14，如此一来，我们就实现了执行微地址位14的微程序！仿照ppt上微地址10-13里的指令，写一下新的微程序吧，别忘了JP返回。

大致思路如此，如果你理解了然后去自己实现，那么或许会发现你的实现和我描述的`不一样`。
我只是`提供一种思路`和对不知道如何扩展寻址方式的同学一点提示，里面也有许多的缺陷，本人水平有限，也请各位同学带着批判的态度来阅读。

FAQ

Q:为什么我不能使用uIR14（PIN_108）？

A:不是不能使用，是需要进行一定的设置。首先打开选择208C8的页面，点击页面中部右侧的Device and Pin Options按钮。随后在上方的卡片中选择Dual-Purpose Pins，最后将nCEO改为如图所示的Use as regular I/O即可。

Q:

为什么我读出来的数老是比预期多一？

A:

这个问题很明显是ALU的锅，在我们实验四的ALU中，C0端若为低电平，则运算后加一，若为高电平，则不加一。我们在设计指令的时候，无论是低电平有效的器件还是高电平有效的器件，都会设计成在微指令中为1时有效，这个时候就可能发生错误加一的情况。所以，如果你用微指令控制C0，记得先通过一个非门，再连接到C0端。

Q:

为什么我的指令总是在取值周期无限循环？

A:

如果真的是这样，那很可能每次都是跳到微地址00，检查后继微地址形成部件。

Q:

我怀疑遇到了玄学问题！

A:

先不要怀疑是玄学，还是检查自己的电路图，一点一点检查，可以在必要之处连接灯，看取出来的数据到底对不对。如果怀疑某一个器件有问题，可以把它单独拿出来放在一个工程文件里，用开关控制数据进行检验。

剩下的加油啦，祝大家计组实验顺利。