

（深圳）

实验作业

开课学期： 2022春季

课程名称：计算机组成原理（实验）

实验名称： 直接映射Cache设计

实验性质： 综合设计型

实验学时： 4 地点： T2615

学生班级： 5班

学生学号： 200110513

学生姓名： 宗晴

作业成绩：

实验与创新实践教育中心制

2022年5月

|  |
| --- |
| 1. Cache模块设计 |
| （画出读、写的状态转移图，并描述状态之间的转移关系和转移条件、以及每个状态需要完成什么操作。）  （1）读状态转移图如下    READY表示chache处在就绪状态。TAG\_CHECK表示检查cache是否命中。REFILL表示将取回的字块，加上相应的标签，存入自己的存储体内。  状态之间的转移关系和转移条件：若接受到reset信号，则恢复为READY状态。当前处于READY状态时，若接受到来自CPU的读请求req，则进入TAG\_CHECK状态，否则保持在READY状态不变。当前处于TAG\_CHECK状态时，cache检查读操作是否命中，若命中则转移到READY状态，否则转移到REFILL状态。命中的条件为：当前状态是TAG\_CHECK状态，并且主存字块标记与来自cache的tag相同，同时数据有效位为1。当前处于REFILL状态时，若成功从主存中取得数据，则转入TAG\_CHECK状态，否则保持为REFILL状态。  每个状态需完成的操作：READY状态时，将输出给下层内存模块的读请求以及首地址置零，读出cache中相应地址处的数据。TAG\_CHECK状态时，将输出给下层内存模块的读请求以及首地址置零，同时判断读cache操作是否命中（命中条件如上所述），当命中时需要将对应数据输出给CPU。REFILL状态时，将输出给下层内存模块的读请求置1，同时将从CPU传来的地址输出给下层内存模块，读取相应位置处的数据，将读到的数据转换为cache行数据写入cache中，完成cache重填。  （2）写状态转移图如下    READY表示chache处在就绪状态。TAG\_CHECK表示检查cache是否命中。WR\_DATA表示写命中状态，修改cache及内存。（在实际代码中，写时序的WR\_DATA与读时序共用同一状态转移名称REFILL）  状态之间的转移关系和转移条件：若接受到reset信号，则恢复为READY状态。当前处于READY状态时，若接受到来自CPU的写请求req，则进入TAG\_CHECK状态，否则保持在READY状态不变。当前处于TAG\_CHECK状态时，cache检查写操作是否命中，若命中则转移到WR\_DATA状态，否则转移到READY状态。命中的条件为：当前状态是TAG\_CHECK状态，并且主存字块标记与来自cache的tag相同，同时数据有效位为1。当前处于WR\_DATA状态时，若获取数据，则转移到TAG\_CHECK状态，否则保持。（但实际过程中，由于写cache操作时，读内存信号保持为0，所以数据有效信号保持为0，所以该转移不可能实现）  每个状态需完成的操作：READY状态时，将输出给下层内存模块的写请求、待写数据以及首地址置零，读出cache中相应地址处的数据。TAG\_CHECK状态时，将输出给下层内存模块的写请求、待写数据以及首地址置零，同时判断写cache操作是否命中（命中条件如上所述），当命中时需要将cache写使能拉高，将CPU来的数据写入cache行中的对应位置；若未命中，则不修改cache的内容。WR\_DATA状态时，将输出给下层内存模块的写请求置1，并将相应地址与数据输出给下层内存模块。 |
| 1. **调试报告** |
| （仿真截图及时序分析，要求包含读命中、读缺失、写命中及写缺失共四种情况的分析，且每种情况需列举2个测试用例进行分析。）   1. 对于读操作进行分析（关注与读操作有关的亮绿色波形部分即可）   （1）130ns-530ns，cache第一次读缺失（此操作为第一次对cache进行操作，cache中未存储数据，因此读缺失，符合预期）  1653109640(1)  整体波形如上图。  1653109678(1)  如上图所示，105ns时，reset信号拉低，表示cache可以开始正常工作。130ns时，rreq\_from\_cpu拉高，从CPU传来读请求。同时，addr\_from\_cpu为0000，表示从CPU传来的地址为0000，此时cache判断是否命中。虽然主存字块标记与来自cache的tag相同，但是valid\_bit为0，表示数据无效，所以未命中。一周期后，hit信号仍为0，表示未命中，符合预期。一周期后，即170ns时，rreq\_to\_mem拉高，表示cache向下层内存模块发出读请求，进行cache的重填阶段。符合预期。  1653109708(1)  如上图所示，读取内存阶段持续到490ns，此时rdata\_from\_mem显示为0e0c0b1a，表示主存返回的数据为0e0c0b1a。cache\_line\_r显示为200e0c0b1a，表示待写入cache的行数据为200e0c0b1a（十六进制表示），其中最高位的2由数据有效位为1得到，中间5位0表示存储地址，最后的0e0c0b1a表示存储的数据。同时，rvalid\_from\_mem信号拉高，表示数据已准备好，可以取走，符合预期。此时wea信号拉高，即cache写使能信号有效，将上述cache\_line\_r写入cache中，符合预期。一周期后，即510ns时，hit\_to\_cpu信号拉高，表示cache读命中，从cache中读出数据cache\_line为200e0c0b1a，符合预期。此时rdata\_to\_cpu为1a，表示输出到CPU的数据为1a，符合预期。hit信号和读出的数据均只持续一个周期，同时有效，符合预期。一周期后，即530ns时，rreq\_from\_cpu拉低，这一轮的读操作结束。  （2）570ns-610ns，cache第一次读命中（此操作紧跟上一轮读缺失，且读取地址相邻，所读内容0b为上一周期写入cache中的数据0e0c0b1a的一部分，符合预期）  1653110274(1)  570ns时，rreq\_from\_cpu拉高，从CPU传来读请求。同时，addr\_from\_cpu为0001，表示从CPU传来的地址为0001，此时cache判断是否命中。此时主存字块标记与来自cache的tag相同，并且valid\_bit为1，表示数据有效，所以命中。一周期后，即590ns时，hit\_to\_cpu信号拉高，表示cache读命中，从cache中读出数据，符合预期。此时rdata\_to\_cpu为0b，表示读出的数据为0b，符合预期。hit信号和读出的数据均只持续一个周期，同时有效，符合预期。一周期后，即610ns时，rreq\_from\_cpu拉低，这一轮的读操作结束。  （3）650ns-690ns，cache第二次读命中（此操作紧跟上一轮读命中，且读取地址相邻，所读内容0c仍为第一次读缺失时写入cache中的数据0e0c0b1a的一部分，符合预期）  1653111076(1)  650ns时，rreq\_from\_cpu拉高，从CPU传来读请求。同时，addr\_from\_cpu为0002，表示从CPU传来的地址为0002，此时cache判断是否命中。此时主存字块标记与来自cache的tag相同，并且valid\_bit为1，表示数据有效，所以命中。一周期后，即670ns时，hit\_to\_cpu信号拉高，表示cache读命中，从cache中读出数据，符合预期。此时rdata\_to\_cpu为0c，表示读出的数据为0c，符合预期。hit信号和读出的数据均只持续一个周期，同时有效，符合预期。一周期后，即690ns时，rreq\_from\_cpu拉低，这一轮的读操作结束。  （4）810ns-1210ns，cache第二次读缺失（此操作为第五次对cache进行读操作，且前五次读取地址均相邻。第一次读缺失时写入cache中的数据已经于前四次读完，cache的第5个地址中未存储数据，因此读缺失，符合预期）  1653111477(1)  整体波形如上图。    如上图所示，810ns时，rreq\_from\_cpu拉高，从CPU传来读请求。同时，addr\_from\_cpu为0004，表示从CPU传来的地址为0004，此时cache判断是否命中。虽然主存字块标记与来自cache的tag相同，但是valid\_bit为0，表示数据无效，所以未命中。一周期后，hit信号仍为0，表示未命中，符合预期。一周期后，即850ns时，rreq\_to\_mem拉高，表示cache向下层内存模块发出读请求，进行cache的重填阶段。符合预期。  1653112156(1)  如上图所示，读取内存阶段持续到1170ns，此时rdata\_from\_mem显示为2c21ff80，表示主存返回的数据为2c21ff80。cache\_line\_r显示为202c21ff80，表示待写入cache的行数据为202c21ff80（十六进制表示），其中最高位的2由数据有效位为1得到，中间5位表示存储地址，最后的2c21ff80表示存储的数据。同时，rvalid\_from\_mem信号拉高，表示数据已准备好，可以取走，符合预期。此时wea信号拉高，即cache写使能信号有效，将上述cache\_line\_r写入cache中，符合预期。一周期后，即1190ns时，hit\_to\_cpu信号拉高，表示cache读命中，从cache中读出数据cache\_line为202c21ff80，符合预期。此时rdata\_to\_cpu为80，表示输出到CPU的数据为80，符合预期。hit信号和读出的数据均只持续一个周期，同时有效，符合预期。一周期后，即1210ns时，rreq\_from\_cpu拉低，这一轮的读操作结束。   1. 对于写操作进行分析（关注与写操作有关的亮绿色波形部分即可）   （1）1392690ns-1392730ns，cache第一次写缺失  1653115080(1)  如上图所示，1392690ns时，wreq\_from\_cpu拉高，从CPU传来写请求。同时，addr\_from\_cpu为0000，表示从CPU传来的地址为0000，wdata\_from\_cpu为ff，表示待写入CPU的数据为ff。此时cache判断是否命中，虽然valid\_bit为1，表示数据有效，但主存字块标记为00，而来自cache的tag为1f，二者不同，所以未命中。一周期后，hit信号仍为0，表示未命中，符合预期。此时cache\_line\_r为200e0c0bff，表示待写入cache的行数据为200e0c0bff，但由于未命中，所以cache的写使能wea为0，并未向cache写入数据。且wreq\_to\_mem为0，也并未向内存中写入数据，符合预期。一周期后，即1392730ns时，wreq\_from\_cpu拉低，这一轮的写操作结束。  （2）1393210ns-1393250ns，cache第二次写缺失  1653115640(1)  如上图所示，1393210ns时，wreq\_from\_cpu拉高，从CPU传来写请求。同时，addr\_from\_cpu为1f00，表示从CPU传来的地址为1f00，wdata\_from\_cpu为ff，表示待写入CPU的数据为ff。此时cache判断是否命中，虽然valid\_bit为1，表示数据有效，但主存字块标记为1f，而来自cache的tag为00，二者不同，所以未命中。一周期后，hit信号仍为0，表示未命中，符合预期。此时cache\_line\_r为3f0e0c0bff，表示待写入cache的行数据为3f0e0c0bff，但由于未命中，所以cache的写使能wea为0，并未向cache写入数据。且wreq\_to\_mem为0，也并未向内存中写入数据，符合预期。一周期后，即1393250ns时，wreq\_from\_cpu拉低，这一轮的写操作结束。  （3）1394090ns-1394150ns，cache第一次写命中  1653115891(1)  如上图所示，1394090ns时，wreq\_from\_cpu拉高，从CPU传来写请求。同时，addr\_from\_cpu为1f04，表示从CPU传来的地址为1f04，wdata\_from\_cpu为ff，表示待写入CPU的数据为ff。此时cache判断是否命中，valid\_bit为1，表示数据有效，且主存字块标记与来自cache的tag均为1f，二者相同，所以命中。一周期后，即1394110ns时，hit信号为1，表示命中，符合预期。此时cache\_line\_r为3f2c21ffff，表示待写入cache的行数据为3f2c21ffff，由于命中，所以cache的写使能wea为1，向cache写入数据3f2c21ffff。且一周期后，即1394130ns时，wreq\_to\_mem为1，同时waddr\_to\_mem为1f04，wdata\_to\_mem为ff，向内存地址1f04中写入数据ff，符合预期。同时，wreq\_from\_cpu拉低，这一轮的写操作结束。  （4）1394970ns-1395030ns，cache第二次写命中  1653116921(1)  如上图所示，1394970ns时，wreq\_from\_cpu拉高，从CPU传来写请求。同时，addr\_from\_cpu为1f08，表示从CPU传来的地址为1f08，wdata\_from\_cpu为ff，表示待写入CPU的数据为ff。此时cache判断是否命中，valid\_bit为1，表示数据有效，且主存字块标记与来自cache的tag均为1f，二者相同，所以命中。一周期后，即1394990ns时，hit信号为1，表示命中，符合预期。此时cache\_line\_r为3f0a1c3aff，表示待写入cache的行数据为3f0a1c3aff，由于命中，所以cache的写使能wea为1，向cache写入数据3f0a1c3aff。且一周期后，即1395010ns时，wreq\_to\_mem为1，同时waddr\_to\_mem为1f08，wdata\_to\_mem为ff，向内存地址1f08中写入数据ff，符合预期。同时，wreq\_from\_cpu拉低，这一轮的写操作结束。 |