

（深圳）

实验报告

开课学期： 2024春季

课程名称：计算机组成原理（实验）

实验名称： 高速缓存器设计

实验性质： 设计型

实验学时： 4 地点： 宿舍

学生班级： 计算机10班

学生学号： 220111016

学生姓名： 杨绵绵

作业成绩：

实验与创新实践教育中心制

2024年5月

|  |
| --- |
| 1、Cache详细设计 |
| 要求：绘制ICache的状态转换图，并详细描述状态转移关系、转移条件、各状态的输入输出信号以及需要完成的操作。*\*若完成了附加题，则分别绘制DCache的读、写状态转换图，并配以文字详细描述相应的内容。* |
| fig:ICache的状态转换图  IDLE：为初始状态；当接收到inst\_rreq时，跳转到TAG\_CHECK状态；   * mem\_ren：设置为4'b0000 * mem\_raddr：若inst\_rreq为高，设置为inst\_addr；否则为32'h0 * inst\_valid:设置为1'b0   TAG\_CHECK：当接收到hit信号时，跳转到WAIT状态等待读取数据输出；若hit信号为低电平且mem\_rrdy信号为高电平，则跳转到REFILL状态；否则一直停留在TAG\_CHECK状态；   * 若hit为高电平   + mem\_ren：设置为4'b0000   + mem\_raddr：设置为32'h0   + inst\_valid:设置为1'b0 * 若hit为低电平，mem\_rrdy为高电平，说明发生读缺失，需要访存   + mem\_ren：设置为4'b1111   + mem\_raddr：设置为访存地址{tag\_from\_cpu, cache\_index, 4'b0000}   + inst\_valid:设置为1'b0   REFILL：当接收到mem\_rvalid信号，跳转到WAIT状态；否则停留在REFILL状态。   * 若mem\_rvalid为高电平：   + cache\_tags[cache+index]设置为{1'b1,tag\_from\_cpu}，即有效位设置为1，并更新tag位   + inst\_valid:设置为1'b0 * mem\_ren：设置为4‘b0000   WAIT:等待一个周期，跳转到IDLE状态。   * inst\_valid：设置为1'b1；   DCache的状态转换图： fig:  读状态机  IDLE\_R：为初始状态； 首先判断是否为uncached访问； 当接收到uncached且data\_ren非0时，跳转到UNCACHED\_READ状态； 当未接收到uncached且data\_ren非0时，跳转到TAG\_CHECK\_R状态； 否则，停留在IDLE\_R状态。   * dev\_ren：设置为4'b0000 * dev\_raddr：若dev\_ren不为0，设置为data\_addr；否则为32'h0 * data\_valid:设置为1'b0 * ren\_reg:用于记录读取使能信号，设置为data\_ren   TAGCHECKR：当接收到hit\_r信号时，跳转到WAIT\_R状态等待读取数据输出；若hit信号为低电平且dev\_rrdy信号为高电平，则跳转到REFILL\_R状态；否则一直停留在TAG\_CHECK\_R状态；   * 若hit为高电平   + dev\_ren：设置为4'b0000   + dev\_raddr：设置为32'h0   + data\_valid:设置为1'b0 * 若hit为低电平，dev\_rrdy为高电平，说明发生读缺失，需要访存   + dev\_ren：设置为4'b1111   + dev\_raddr：设置为访存地址{tag\_from\_cpu, cache\_index, 4'b0000}   + data\_valid:设置为1'b0   REFILL\_R：当接收到dev\_rvalid信号，跳转到WAIT\_R状态；否则停留在REFILL\_R状态。   * 若dev\_rvalid为高电平：   + cache\_tags[cache+index]设置为{1'b1,tag\_from\_cpu}，即有效位设置为1，并更新tag位   + data\_valid:设置为1'b0 * dev\_ren：设置为4‘b0000   WAIT\_R:等待一个周期，跳转到IDLE状态。·   * inst\_valid：设置为1'b1；   UNCACHED\_READ:若接收到dev\_rvalid信号，跳转到 WAIT\_R状态；否则保持在UNCACAHED\_READ状态；   * dev\_ren:设置为data\_ren * dev\_raddr:设置为data\_addr * 若dev\_rvalid为高电平：   + dev\_rdata：设置为data\_rdata   + dev\_rvalid：设置为1   + dev\_ren：4‘b0   写状态机  IDLE\_W：为初始状态； 首先判断是否为uncached访问； 当接收到uncached且data\_wen非0时，跳转到UNCACHED\_WRITE状态； 当未接收到uncached且data\_wen非0时，跳转到TAG\_CHECK\_W状态； 否则，停留在IDLE\_W状态。   * dev\_wen：   + 若uncached为1,设置为data\_wen，否则为4’b0； * dev\_waddr：设置为data\_addr * data\_wresp:设置为1'b0 * dev\_data：   + 若uncached为1,设置为data\_wdata，否则为0；   TAGCHECKW： 当接收到hit\_w信号时，跳转到HIT\_W状态； 若hit\_w信号为低电平且dirty[cache\_index]为1，并且dev\_rrdy信号为高电平，即则跳转到WRITE\_BACK状态； 若hit\_w信号为低电平且dirty[cache\_index]为0，并且dev\_rrdy信号为高电平，即则跳转到ALLOCATE状态； 否则一直停留在TAG\_CHECK\_W状态；   * 若hit\_w为高电平   + dev\_ren：设置为4'b0000   + dev\_raddr：设置为32'h0 * 若hit\_w为低电平，dev\_rrdy为高电平，dirty[cache\_index]为高电平：   + 此处的操作为写缺失且cache块被标记为脏，需要写回，故操作等待到WB阶段进行。 * 若hit\_w为低电平，dev\_rrdy为高电平，dirty[cache\_index]为低电平：   + dev   HIT\_W：命中状态。跳转到WAIT\_W状态  ALLOACTE：分配阶段。 若dev\_rvalid为高电平，即已经完成读取主存中的数据，则跳转到TAF\_CHECK\_W阶段，否则仍保持在ALLOCATE阶段   * 若dev\_rvalid为高电平   + dirty[cache\_index]设置为0   + cache\_tags[cache\_index]设置为{1'b1, tag\_from\_cpu}；   WRITE\_BACK：写回阶段。 此处因为cacheline大小为4个32位字，而写主存一次只能写1个字，故需要进行4次主存写操作才能将一个cacheline的内容写回主存；此处选择添加了3个状态WB1-3，加上WB共4个状态，用于写回4个字。 若dev\_wrdy，即本次写操作完成，则跳转到下一阶段；否则停留在本阶段。  对于WB3：若dev\_wrdy，即本次写操作完成，则跳转到ALLOCATE状态。   * 若dev\_wrdy:   + dev\_wen <= 4'hf;   + dev\_wdata <= cache\_line\_r[95:64];   + dev\_waddr <= {tag\_from\_cache,cache\_index, 4'b1000};   HIT\_W：命中。修改脏标记位和有效标记位   * dirty[cache\_index] <= 1'b1;*//修改脏位* * cache\_tags[cache\_index] <= {1'b1,tag\_from\_cpu};   WAIT\_W:等待一个周期，跳转到IDLE状态。   * 若dev\_wrdy：   + data\_wresp：设置为1   + dev\_wen：设置为0   UNCACHED\_READ:若接收到dev\_rvalid信号，跳转到 WAIT\_W状态；否则保持在UNCACAHED\_WRITE状态；   * dev\_wen:设置为0 |
| 2、调试报告 |
| 要求：结合仿真波形截图对ICache作详细的时序分析，要求包含读命中、读缺失2种情形，且每种情形列举2个测试用例。*\*若完成了附加题，则需额外给出DCache的仿真波形截图及其详细文字分析，要求包含写命中、写缺失和Uncached访问3种情形。* |
| 测试用例1  fig:  fig: 首先cpu发出inst\_rreq，取值地址为0x00000000，应取指令为0x0040006f；状态来到TAG\_CHECK； 由于是第一次取指，必定发生读缺失，hit信号为低电平；状态转移到REFILL 经历很长一段时间，mem\_rvalid为高电平时，cache\_we为高电平；cache\_line\_w在同一时刻变为从主存中读出的内容； 下一时钟上升沿，状态来到TAG\_CHECK；cache完成写入，并将写入cache的内容呈现在cache\_line\_r中，且更新tag和validbit。此时hit信号为高电平，读命中。进入WAIT阶段等待输出。 下一时钟上升沿，inst\_out被更新为cache命中取到的指令，inst\_valid拉高1，成功输出应取指令0x0040006f；同时状态来到IDLE  接下来3条指令都是读命中，时序类似。  测试用例2 fig:  如图所示，在连续三次命中后，取地址为0x00000010的指令时发生了读缺失；在TAG\_CHECK状态，检测到hit信号变为0，于是进入REFILL阶段。 mem\_ren变为1111，mem\_addr变为0x00000010，开始读主存。 fig:  经历很长一段时间，mem\_rvalid为高电平时，cache\_we为高电平；cache\_line\_w在同一时刻变为从主存中读出的内容；  下一时钟上升沿，状态来到TAG\_CHECK；cache完成写入，并将写入cache的内容呈现在cache\_line\_r中，且更新tag和validbit。此时hit信号为高电平，读命中。进入WAIT阶段等待输出。  下一时钟上升沿，inst\_out被更新为cache命中取到的指令，inst\_valid拉高1，成功输出应取指令0x0080a023；同时状态来到IDLE  接下来3条指令都是读命中，时序类似。  测试用例1：uncached访问（写）  fig:  如图，data\_wen为1111，uncached为1，从IDLE\_W阶段进入UNCACHED\_WRITE阶段； 一段时间后，写入完成，dev\_wrdy变为1，将wresp拉高变为1，表明完成了uncached写操作。  测试用例2：写缺失与写命中  fig:  如图，data\_wen为1111，且uncached为0，并且hit\_w为0，发生写缺失。状态进入ALLOCATE 将dev\_raddr设置为data\_addr:00004010。dev\_ren设置为1111，开始读取主存内容；  fig:  过去一段时间，dev\_rvalid为1，同时cache\_we为1; cache写入主存数据完成后，状态进入5即TAG\_CHECK，此时hit\_w为1，写命中；状态又跳转为b即HIT\_W；随后进入状态aWAIT\_W；cache\_ce再次为1，写入cpu要写入的数据，对cache修改完成后wresp拉高为1。 |
| 3、思考与讨论 |
| （1）分别给出无ICache时和有ICache时，SoC运行测试程序的总时间的截图，并谈谈你对该测试结果的理解。  random delay+ without ICache  fig:  random delay + ICache  fig:  添加了ICache之后的运行总时间为原来的27.118%。 这表面添加ICache确实大大提高了程序运行的速度，因为运行过程中取指令的过程被大大加快了，避免了cpu的空转。  （2）给出你的ICache命中率的截图，并尝试分析如何提高ICache命中率。  fig:  直接映射的ICache，对于这一测试程序，命中率为79%。 我认为可以通过改变ICache的映射方式（例如全相联或n路组相联）并更换更灵活的替换算法来提高命中率。 |
| 4、总结与反思 |
| 要求：总结完成本课程实验获得的收获，并给出合理的意见和建议。 |
| 通过计组课程，我再次熟悉并增进了verilog编程的技术，且对数字逻辑、时序分析等有了更深的理解；同时，因为准备参加龙芯杯竞赛，在备赛的同时完成计组实验也非常有助于理解cpu实现的原理，包括一些平时可能不倾向于主动去写的、有助于提高cpu性能的部件（例如ICache和DCache），在实验的督促下也算是写出来了，不过做的仓促，性能还没有优化过，稍作修改后应该能加入自己的MIPScpu中。  建议：我在完成ICache的实验之初感到一头雾水，虽然理解cache的原理和实现逻辑，但是具体的实现方式对于初次上手来说还是很抽象的，对着指导书和框架代码看了很久很久才明白自己要添加什么内容、修改什么内容。所以，我认为实验指导书可以适当给予一些提示，因为框架只给出了关键代码填空，对于cache\_tag表和有效位表是没有提及的，指导书也没有提及，对于一些脑筋转的慢（比如我）的学生来说，会需要比较长的一段时间去悟出来这些东西要自己写。 |