山东大学 计算机科学与技术 学院

计算机组成与设计 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号： | 姓名： | | 班级： |
| 实验题目：补码加法器 | | | |
| 实验学时：2 | | 实验日期：2025/3/11 | |
| 实验目的：  本实验要求采用一位全加器和异或门设计一个四位二进制补码加法器，操作数和运算结果均采用补码表示，该加法器要求可做补码加法，也可做补码减法运算。能够通过功能测试验证补码加法器的正确性。 | | | |
| 实验软件和硬件环境：  软件环境：  Vivado软件、FPGA实验平台  硬件环境：  1.实验室台式机  2.FPGA服务器，PYNQ-Z2开发板 | | | |
| 实验原理和方法：  实验原理：  在补码运算中：    （X)补 十（Y)补 ＝（X十Y)补     （X)补 -（Y)补 ＝（X)补 十（-Y)补  下面是是八位补码加法器的原理图，图中SUM7-SUM0表示四位和数，C4为向高位的进位，A7-A0为A操作数，B7-B0为B操作数，C0为低位来的进位， K为控制端，当K为高电平时，执行减法运算，K为低电平时，执行加法运算。  然后根据K的正负来进行输入数据的取反，具体实现方法就将K与输入的数据进行异或，减法K就是1，然后进行异或，就把原来的数据取反了，然后进行后续的步骤。  后面就是m7482加法器，将第一个低位的加法器的进位端设置为K，因为前面只是取反，并没有末尾加一，这里K=1就是减法，然后就加一，K=0就是加法，然后就是加零。  然后把每一位的向高位进位的输出端，连接到高位相加的m7482的低位进位输入端，这样然后把对应的输出连接到adpt\_out的输入接口，最后，adpt\_out的输出端连接到LED灯的位置。    实验方法：  硬件搭建与连接  使用FPGA开发板（PYNQ-Z2）和Vivado 2022.2软件搭建实验环境。  将FPGA开发板与实验室台式机连接，确保硬件环境正常运行。  根据原理图连接输入模块（A、B操作数、K控制端）和输出模块（LED灯），并通过FPGA服务器分配远程开发板资源。  软件设计与验证  在Vivado中创建工程，选择PYNQ-Z2器件，添加实验源代码和IP核。  使用Block Design工具搭建四位补码加法器的顶层设计，将输入模块与cg\_fpga模块连接，并设置C0与K连接以实现减法操作。  生成HDL Wrapper并综合生成bit文件，确保顶层模块设置正确。  将bit文件烧写至FPGA开发板，通过输入开关键改变操作数的值，观察LED灯显示的输出结果。  功能测试与验证  测试加法和减法功能：  加法测试：设置K=0，输入操作数A和B，观察LED灯显示的输出结果是否与理论值一致。  减法测试：设置K=1，输入操作数A和B，观察LED灯显示的输出结果是否与理论值一致。  测试溢出检测：通过观察溢出端LED灯的状态，验证溢出检测功能是否正确。  结果记录与分析  记录不同输入情况下的理论输出和实际输出，对比分析结果是否一致。  验证高位相加和溢出情况下的输出是否符合预期。 | | | |
| 实验步骤：  （1）创建工程：打开本地安装的Vivado 2022.2，新建项目，选择pynq-z2器件。  （2）. 添加实验环境：进入FPGA在线实验环境，点击右上角项目材料下载实验源代码和希冀ip核到本地并解压。  （3）3. 在Vivado项目中，点击 Settings → IP → Repository ，将上一步解压后的ip\_repo文件夹的位置  添加进IP搜索目录。  （4）点击 Sources 窗口中的 + ，选择 Add or create design sources → Next → Add File , 添加  实验源代码文件。  （5）点击 Create Block Design 创建一个新的顶层设计，随后点击添加IP核按钮，添加cg\_fpga IP.  （6）在 Sources 窗口下的 Design sources 中，根据上面的电路图拖拽相应模块，完成原理图的输入。  （7）将输入的模块与cg\_fpga如下图连接，并选择cg\_fpga模块上的 FIXED\_IO 和 DDR ，点击右键→ Make  External . 这里为了保证补码减法器运算正确，可直接将 C0 与输入 K 连接。  （8）：右击 Sources 下顶层设计图标→ Create HDL Wrapper ，待Wrapper正确生成后，点击左下方 Generate Bitstream ，开始综合并生成bit文件。注意：综合前 wrapper 模块应被设置为顶层 （加粗表示），若自动设置错误，需右击wrapper图标点击 Set as Top 手动设置。  （9）：通过 FPGA 云实验平台，可在线分配远程 FPGA 硬件开发板。首先点击 connect 按钮，然后在下  拉菜单中选择任意空闲的开发板，并点击 Choose File 中选择上一步生成的 \*.bit 文件，后点击 send ，即可将本地bit文件烧写至希冀远程FPGA.  利用输入开关键改变输入操作数的值，看LED指示灯显示的结果是否正确并记录结果。下面是具体的验证过程。  我在这里已经将4位扩展成8位加法器了，这里测试的样例就是8位加法器。  结果验证：  首先！在K = 0，a = 00000101，b = 00010001输入情况下分析其正确性。  理论情况下，输出应该是00010110:  下面是K = 1，a = 00000011，b = 00000001输入情况下分析其正确性。  理论情况下输出就是：00000010，通过观察我们发现实际输出和理论输出相同，然后这里的溢出端亮灯更是因为判断的K是相反的。    还有就是高位相加的情况下：K = 0，a = 00110011，b = 00110011，理想情况下输出应该是01100110，实际输出是01100110，这和理论输出是相同的。 | | | |
| 结论分析与体会：  这次的实验让我对于补码加法器有了一个深入的理解。  实验结果分析：  加法验证：在K=0的情况下，输入操作数A=00000101和B=00010001，理论输出为00010110，实际输出与理论值一致，验证了加法功能的正确性。  减法验证：在K=1的情况下，输入操作数A=00000011和B=00000001，理论输出为00000010，实际输出与理论值一致，验证了减法功能的正确性。  高位相加验证：在K=0的情况下，输入操作数A=00110011和B=00110011，理论输出为01100110，实际输出与理论值一致，验证了高位相加功能的正确性。  溢出检测：通过观察溢出端LED灯的状态，验证了溢出检测功能的正确性。  实验结论：  本次实验成功实现了四位二进制补码加法器的设计与验证。通过控制端K的设置，可以实现加法和减法运算，且溢出检测功能正常工作。  实验结果表明，补码加法器的设计符合预期，能够正确处理加法、减法以及溢出情况。  实验体会  通过本次实验，我对补码加法器的原理和实现有了更深入的理解。补码运算的灵活性（通过取反加一实现减法）让我认识到计算机硬件设计的巧妙之处。  本次实验让我更加理解了理论与实践的结合，通过实际操作加深了对计算机组成原理的理解，为后续学习和研究打下了坚实的基础。 | | | |
|  | | | |