**2022《数字逻辑与处理器基础》处理器大作业**

第一部分

2022/05/01

1. 实验目的
2. 掌握基于组合逻辑和时序逻辑的多输入加法器RTL实现方式；
3. 深入理解组合逻辑和时序逻辑在资源/面积和时序性能上的设计折中；
4. 掌握MIPS单周期处理器中32位ALU的控制逻辑与计算逻辑的实现方式。
5. MIPS指令集
6. MIPS指令集子集：

lw, sw, lui,

add, addu, sub, subu, addi, addiu,

and, or, xor, nor, andi, sll, srl, sra, slt, sltu, slti, sltiu,

beq, j, jal, jr, jalr

1. MIPS指令格式：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Instruction | OpCode[5:0] | Rs[4:0] | Rt[4:0] | Rd[4:0] | | Shamt[4:0] | Funct[5:0] |
| lw rt, offset (rs) | 0x23 | rs | rt | offset | | | |
| sw rt, offset (rs) | 0x2b | rs | rt | offset | | | |
| lui rt, imm | 0x0f | 0 | rt | imm | | | |
| add rd, rs, rt | 0 | rs | rt | rd | | 0 | 0x20 |
| addu rd, rs, rt | 0 | rs | rt | rd | | 0 | 0x21 |
| sub rd, rs, rt | 0 | rs | rt | rd | | 0 | 0x22 |
| subu rd, rs, rt | 0 | rs | rt | rd | | 0 | 0x23 |
| addi rt, rs, imm | 0x08 | rs | rt | imm | | | |
| addiu rt, rs, imm | 0x09 | rs | rt | imm | | | |
| and rd, rs, rt | 0 | rs | rt | rd | 0 | | 0x24 |
| or rd, rs, rt | 0 | rs | rt | rd | 0 | | 0x25 |
| xor rd, rs, rt | 0 | rs | rt | rd | 0 | | 0x26 |
| nor rd, rs, rt | 0 | rs | rt | rd | 0 | | 0x27 |
| andi rt, rs, imm | 0x0c | rs | rt | imm | | | |
| sll rd, rt, shamt | 0 | 0 | rt | rd | shamt | | 0 |
| srl rd, rt, shamt | 0 | 0 | rt | rd | shamt | | 0x02 |
| sra rd, rt, shamt | 0 | 0 | rt | rd | shamt | | 0x03 |
| slt rd, rs, rt | 0 | rs | rt | rd | 0 | | 0x2a |
| sltu rd, rs, rt | 0 | rs | rt | rd | 0 | | 0x2b |
| slti rt, rs, imm | 0x0a | rs | rt | imm | | | |
| sltiu rt, rs, imm | 0x0b | rs | rt | imm | | | |
| beq rs, rt, label | 0x04 | rs | rt | offset | | | |
| j target | 0x02 | target | | | | | |
| jal target | 0x03 | target | | | | | |
| jr rs | 0 | rs | 0 | | | | 0x08 |
| jalr rd, rs | 0 | rs | 0 | rd | 0 | | 0x09 |

1. 实验内容
2. （期中考试附加题，2分）请基于**组合逻辑**的设计思路，完成特殊的八输入4-bit加法器电路实现，该加法器输入为~（均为4bit），输出为（8bit）。该加法器包含6位控制信号以及，要求功能如下：
3. 当时，；
4. 当时，；
5. 当时，；

提示：组合逻辑的实现可以采用加法树的实现方式，如图-2所示。八输入加法树需要**4+2+1=7个加法器**，请自行设计所需要的二选一多路选择器与控制逻辑信号。

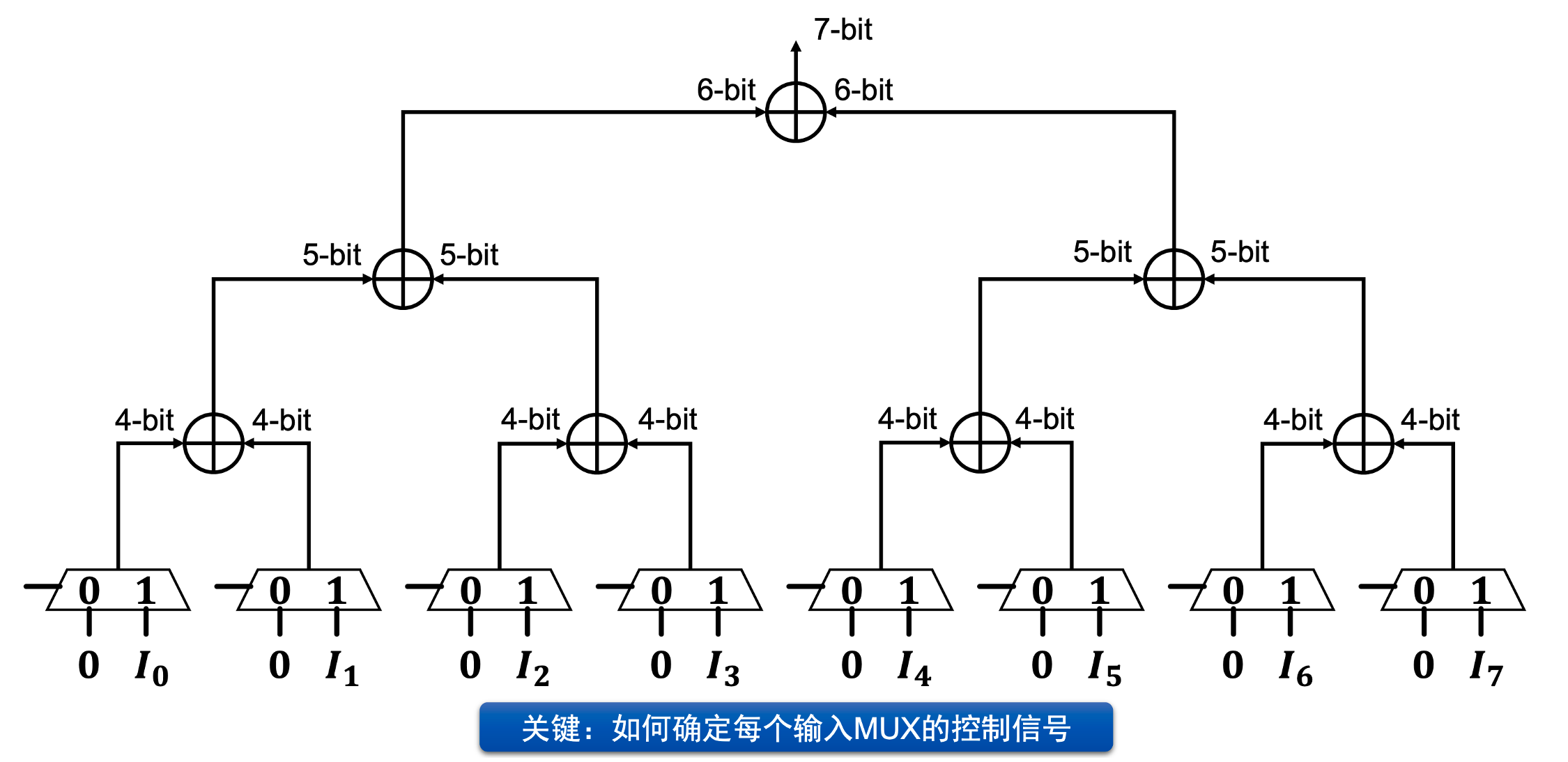


图-1 特殊的八输入4-bit加法器的加法树实现形式（不唯一）

1. （期中考试附加题，2分）请基于**时序逻辑**设计思路，完成该特殊的八输入4-bit加法器电路实现。时序逻辑实现的计算部分要求只使用**1个加法器**（即每个周期/状态只能进行一次加法），请自行设计相应的有限状态机和数据通路。
2. （单周期CPU的ALU设计，3分）根据精简MIPS指令集的OpCode和Funct字段，完成MIPS**单周期**处理器的ALUController和32位ALU的行为级RTL实现，模块接口要求如下：

* ALUController.v模块
  + - * 输入信号：[5:0] OpCode，[5:0] Funct
      * 输出信号：[4:0] ALUCtrl（ALU控制信号），Sign（1-有符号计算，0-无符号计算）
* ALU.v模块
  + - * 输入信号：[4:0] ALUCtrl，Sign，[31:0] in1, [31:0] in2
      * 输出信号：[31:0] out, zero（1-输出out为0，0-输出out不为0）

1. 实验结果与提交材料
2. 针对实验内容1与2，请自行设计测试样例（每种情形至少包含3种测试样例，即种），并进行功能性仿真与验证。提交实验报告并说明你的设计思路、代码逻辑和验证结果，并附上仿真波形图；
3. 针对实验内容1与2，基于Vivado工具进行综合(synthesis)并开展静态时序分析（STA），根据Vivado的资源和时序分析报告，对比组合逻辑与时序逻辑两种实现方式的资源开销与时序性能。提交实验报告并分析说明两种实现方式所可能达到的最高时钟频率、单次计算所需要的最低延时、和所使用的硬件资源开销，并附上Vivado的综合分析报告截图；（提示：最高时钟频率和最低延时可以根据STA方法进行估算得到，合理即可）
4. 提交实验内容1与2中特殊八输入4-bit加法器的Verilog源代码文件、testbench测试代码文件和xdc约束文件，分别存放在***code\_comb***（实验内容1-组合逻辑）和***code\_temp***（实验内容2-时序逻辑）文件夹下；
5. 针对实验内容3，请自行设计测试样例，对精简MIPS指令集中的每一条指令（共计26条MIPS指令）进行功能性仿真与验证。提交实验报告并请说明你的设计思路、代码逻辑和验证结果，并附上仿真波形图；
6. 提交实验内容3中的ALUController和32位ALU的Verilog源代码文件与testbench测试代码文件，统一存放在***code\_ALU***（实验内容3-ALU）文件夹下。
7. 其他说明
8. 本学期处理器大作业分为两部分，第一部分占比7分，第二部分占比8分（附加题3分），总分为15分（超过15分按15分计）。
9. 处理器大作业第一部分的提交时间为5月2日至**5月22日23点59分**，第二部分的提交时间为5月16日至**6月5日23点59分**，晚交将直接影响最终成绩（直接在总分上扣除），最终扣分规则以网络学堂公告为准。
10. 我们鼓励讨论，但是要求所有代码与实验报告均独立完成，严禁抄袭！如发现抄袭现象，将上报学校教务处进行处理。
11. 如对本次处理器大作业有任何问题或建议，请发送邮件至曾书霖助教邮箱（[zengsl18@mails.tsinghua.edu.cn](mailto:zengsl18@mails.tsinghua.edu.cn)），或到罗姆楼4-205房间进行线下咨询。
12. 针对部分没有选修“数逻实验课”的同学，我们提供了数逻实验课的相关资料（如下清华云盘链接所示），供各位同学**自行学习**Verilog语法和相关工具的使用。

链接：<https://cloud.tsinghua.edu.cn/f/cec824dad54b4dcf9c2a/>