

# 2022《数字逻辑与处理器基础》处理器大作业

## 第一部分

2022/05/01

### 一、实验目的

1. 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；
2. 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；
3. 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。

### 二、MIPS 指令集

#### 一、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

#### 二、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

### 三、实验内容

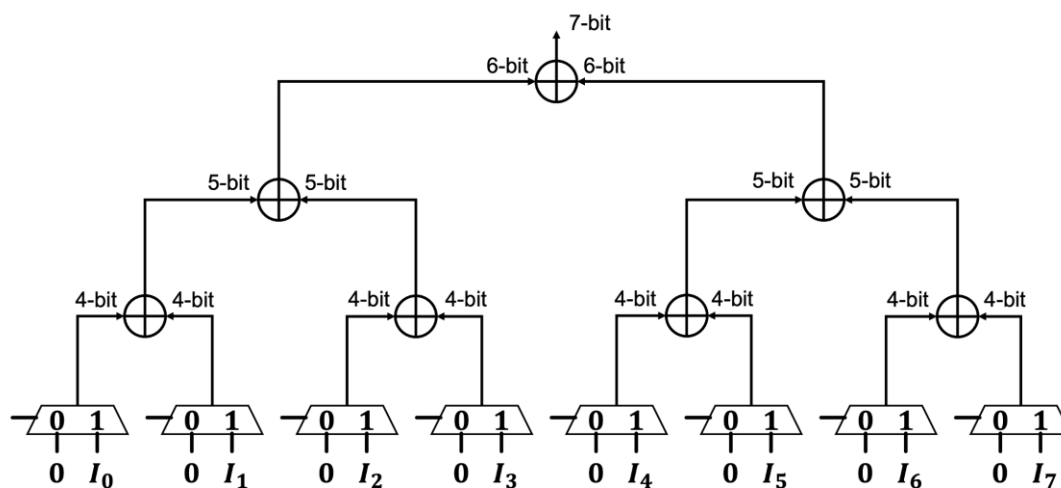
- （期中考试附加题，2分）请基于组合逻辑的设计思路，完成特殊的八输入 4-bit 加法器电路实现，该加法器输入为  $I_7 \sim I_0$  ( $I_i$  均为 4bit)，输出为  $Y$  (8bit)。该加法器包含 6 位控制信号  $(M_2 M_1 M_0)_2$  以及  $(m_2 m_1 m_0)_2$ ，要求功能如下：

1) 当  $(M_2 M_1 M_0)_2 = (m_2 m_1 m_0)_2 = n (n = 0, 1, 2, \dots, 7)$  时， $Y = I_n$ ；

2) 当  $(M_2 M_1 M_0)_2 > (m_2 m_1 m_0)_2$  时， $Y = \sum_{n=(m_2 m_1 m_0)_2}^{(M_2 M_1 M_0)_2} I_n$ ；

3) 当  $(M_2 M_1 M_0)_2 < (m_2 m_1 m_0)_2$  时， $Y = \sum_{n=(M_2 M_1 M_0)_2}^{(m_2 m_1 m_0)_2} I_n$ ；

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。



关键：如何确定每个输入MUX的控制信号

图-1 特殊的八输入 4-bit 加法器的加法树实现形式（不唯一）

- （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。
- （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：

#### ⑩ ALUController.v 模块

⑩ 输入信号：[5:0] OpCode, [5:0] Funct

⑩ 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)

#### ⑩ ALU.v 模块

⑩ 输入信号：[4:0] ALUCtrl, Sign, [31:0] in1, [31:0] in2

⑩ 输出信号：[31:0] out, zero (1-输出 out 为 0，0-输出 out 不为 0)

#### 四、实验结果与提交材料

1. 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  $3 \times 3 = 9$  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；
2. 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 的资源与时序分析报告，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）
3. 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；
4. 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；
5. 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。

#### 五、其他说明

1. 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）。
2. 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。
3. 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。
4. 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。
5. 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（如下清华网盘链接所示），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/>