

清华大学电子工程系

夏季学期实验指导书

实验版

V1.3

课程教学组

2014/4/28

目录

版本历史.....	2
夏季学期实验要求及注意事项	3
夏季学期实验教学安排表	4
夏季学期实验：32 位 MIPS 处理器设计	5

版本历史

版本	日期	说明
1.0	2012.6.16	初始版本
1.1	2012.6.28	<ol style="list-style-type: none">1. ALU 功能表中，修正移位操作符错误；2. ALU 功能示意图，修正选择器地址分配；3. 单周期处理器指令集，将 Bgez 替换为 Bltz；4. 单周期处理器功能框图中，添加 IRQ 信号；5. 修改 DataMemory 地址为字节地址；6. 调整 LED 和 Switch 地址分配，与代码保持一致；7. 附件程序更新到 V1.2。
1.2	2013.6.17	<ol style="list-style-type: none">8. 修改 ALU 输出端口名称；9. 修改外部地址空间描述，统一地址为字节地址；
1.3	2014.4.28	<ol style="list-style-type: none">10. 增加串口外设

夏季学期实验要求及注意事项

1. 实验分组进行，最多 3 人组成 1 组，实验评分分为现场验收和实验报告两部分。
2. 实验时间 2012.6.30~2012.7.17，具体实验时间安排见后面“夏季学期教学实验安排表”。
3. 现场验收由助教根据现场硬件情况核定，按照小组进行验收（要求小组所有成员必须在场），现场成绩占个人成绩的 50%。
4. 实验报告可以小组完成，各部分必须标明完成人，报告成绩占个人总成绩的 50%。
5. 实验报告内容包括：实验目的；设计方案（原理说明及框图）；关键代码及文件清单；仿真结果及分析；综合情况（面积和时序性能）；硬件调试情况；思想体会。
6. 实验报告提交方式：实验报告(word 或者 pdf)和设计代码打包后提交到网络学堂，提交打包文件名按照“学号_姓名_实验编号”的规则命名。
7. 硬件实验板将在实验验收结束时上缴。
8. 同学可以在主楼 9 楼机房完成实验，也可以在寝室或者其他地方自行完成实验，没有考勤限制，现场验收必须在指点时间段内的主楼 9 楼机房进行。
9. 现场验收截止点为 7 月 17 日，现场验收采取预约制。
10. 根据综合结果，在流水线设计功能正确的所有小组中，具有最高时钟频率的前 10 个小组将可获得该实验 10%的加分，申请加分的小组需要单独提交申请，并需另提交一份代码和设计说明。
11. 注意，请同学们在遇到问题时，一定要先通过仿真自行查找原因，如果没有仿真结果，助教有权不予回答，请珍惜助教和其他同学的时间。
12. 实验严禁抄袭，有抄袭嫌疑（实验报告或者设计代码出现雷同、回答问题明显非个人完成等）的现场实验或者实验报告按零分处理。

夏季学期实验教学安排表

		周一	周二	周三	周四	周五
第 1 周 (6.30~7.4)	上午		第一组	第二组	第一组	第二组
	下午	讲座	第一组 (答疑)	第二组 (答疑)	第一组 (答疑)	第二组 (答疑)
第 2 周 (7.7~7.11)	上午	第一组	第二组	第一组	第二组	
	下午	第一组 (答疑)	第二组 (答疑)	第一组 (答疑)	第二组 (答疑)	春季学期补验收
第 3 周 (7.14~7.18)	上午	第一组 (验收)	第二组 (验收)	第一组 (验收)	第二组 (验收)	无
	下午	第一组 (验收)	第二组 (验收)	第一组 (验收)	第二组 (验收)	无

- A. 第一组: S1/S2/S6、第二组: S3/S4/S5。
- B. 最多 3 人组成实验小组, 实验小组成员在每组内部 (第一组/第二组) 自行组合。
- C. 时间段安排: 上午 8: 00~11: 30, 下午 11: 30~15: 00
- D. 验收时间: 标识为“验收”的时间段, 上午: 9: 00~11: 00, 下午 13: 00~15: 00。
- E. 答疑时间: 标识为“答疑”的时间段, 下午 12: 30~15: 00, 非本时间段上机的同学也可以来答疑, 但是由于助教和教师人员有限, 优先考虑本时间段的同学。
- F. 其他时间: 没有标识的时间段, 无助教值班, 请自行上机实验。
- G. 验收采取预约制, 助教会提前一天在网上学堂发布预约贴, 请准备验收的小组在验收当天 0: 00 之前进行预约登记, 没有预约的小组不予验收, 验收按照预约顺序进行, 每个小组只有 2 次预约机会。
- H. 注意安排有一次春季学期补验收, 请春季学期实验还没有验收的同学不要错过最后一次验收机会。

夏季学期实验：32 位 MIPS 处理器设计

实验名称：32 位 MIPS 处理器设计

实验目的：熟悉现代处理器的基本工作原理；掌握单周期和流水线处理器的设计方法。

实验原理：

参考教材第五章和第六章相关内容。

实验内容：

对电路进行优化通常有三种途径，分别是：

- A. 对电路进行优化，达到最大性能（最小延时）
- B. 对电路进行优化，达到最小成本（最小面积）
- C. 对电路进行优化，达到最佳的性价比（最小的面积×延时）

实验分为三个阶段，按照次序依次进行。

1. 设计一个 32 位 ALU（30%）。

实现基本的算术、逻辑、关系、位与移位运算，尽量优化以达到最小的面积延时积。ALU 功能表如下所示：

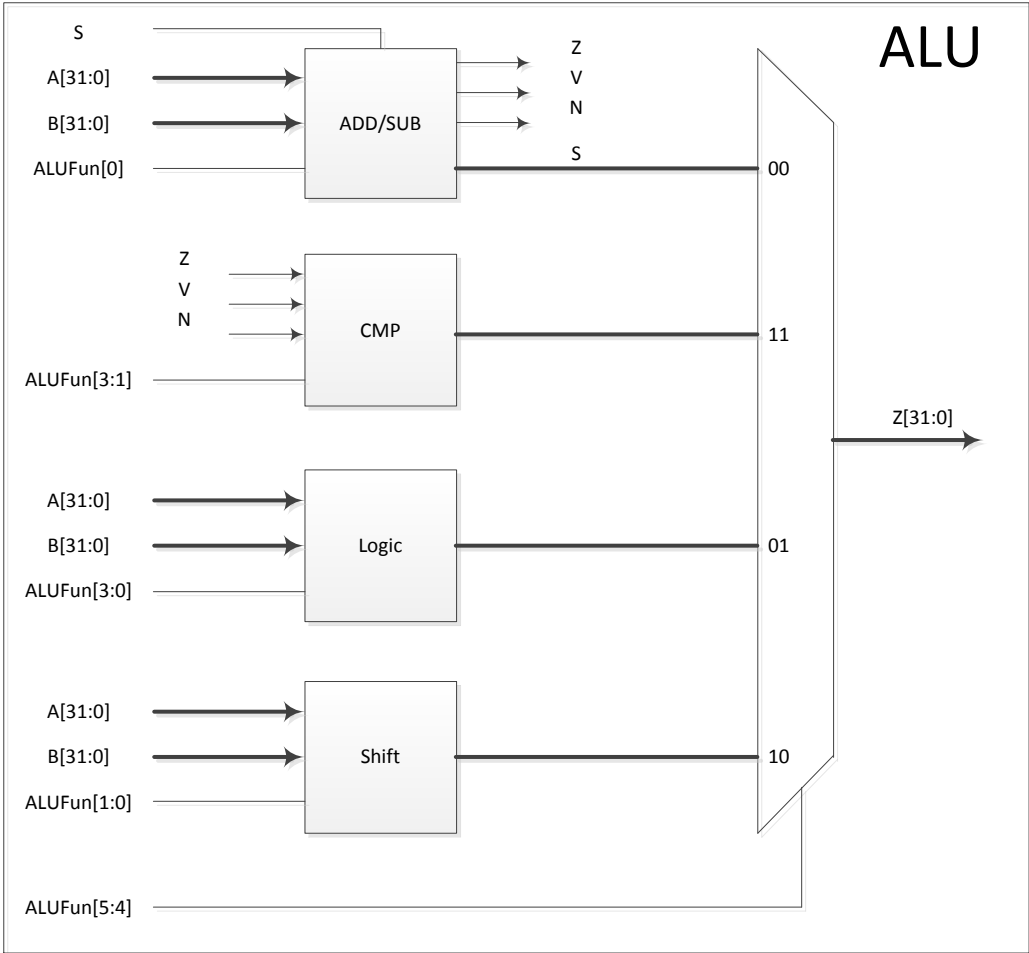
类型	功能	ALUFun	描述
算术	ADD	000000	S=A+B
	SUB	000001	S=A-B
位运算	AND	011000	S=A&B
	OR	011110	S=A B
	XOR	010110	S=A^B
	NOR	010001	S=~(A B)
	“A”	011010	S=A
移位运算	SLL	100000	S=B<<A[4:0]
	SRL	100001	S=B>>A[4:0]
	SRA	100011	S=B>>a[4:0] 算术移位
关系运算	EQ	110011	If(A==B) S=1 else S=0
	NEQ	110001	If(A!=B) S=1 else S=0
	LT	110101	If(A<B) S=1 else S=0
	LEZ	111101	If(A<=0) S=1 else S=0
	GEZ	111001	If(A>=0) S=1 else S=0
	GTZ	111111	If(A>0) S=1 else S=0

ALU 端口如下表所示

名称	类型	描述
A[31:0]	输入	操作数 1

B[31:0]	输入	操作数 2
ALUFun[5:0]	输入	ALU 功能
Sign	输入	运算符号，1: 有符号；0: 无符号
Z[31:0]	输出	结果输出

- A. 加法运算实现可以采用逐次进位、超前进位等结构，减法可以通过加法实现（参见见面理论课讲义或者前面实验）；同时输出 Z（结果为零）、V（结果溢出）、N（结果为负）等标志位，注意有符号数和无符号数标志产生的不同。
- B. 比较运算根据减法运算的结果（Z/V/N）产生，自行分析比较操作与算术运算之间的关系。
- C. 移位运算可以考虑将移位操作拆分为 16 位移位、8 位移位、4 位移位、2 位移位、1 位移位等几个子运算的组合，然后级联形成最后的运算结果。
- D. 逻辑运算可以根据要求直接产生。



ALU 实现示意图（实际可根据自己设计实现）

2. 设计一个单周期 MIPS 处理器（40%）。

- A. 本实验完成一个 32 位的单周期 MIPS 架构处理器，实现核心 MIPS 指令集系统的一个子集，如下：
- i. 空指令：nop
 - ii. 存储访问指令：lw、sw、lui；
 - iii. 算术逻辑指令：add、addu、sub、subu、addi、addiu、and、or、xor、nor、andi、sll、srl、sra、slt、slti、sltiu；
 - iv. 分支和跳转指令 branch(beq、bne、blez、bgtz、bltz)和 jump(j、jal、jr、jalr)；
 - v. 其他指令可以根据情况自行添加。
- B. 该处理器支持异常（为简单起见，可以只支持未定义指令异常）和中断（定时器中断）的处理。
- C. Register File 模块提供 32 路 32bits 的寄存器资源，注意：其中\$0 的取值永远为 0，见附件。
- D. 指令存储器地址空间和数据存储器地址空间是分离的，指令存储器实现采用常量数组（ROM）的方式，根据自己设计的程序设定合理的大小，样例见附件。
- E. 数据存储的地址空间被划分为 2 部分：0x00000000~0x3FFFFFFF（字节地址）为数据 RAM，可以提供数据存储功能；0x40000000~0x7FFFFFFF（字节地址）为外设地址空间，对其地址的读写对应到相应的外设资源（LEDs、SWITCH...），见附件。具体地址划分如下：

地址范围（字节地址）	功能	描述
0x00000000~0x000003FF	数据存储器	256×32bits（可以根据需要自行扩展大小）
0x40000000~0x4000000B	定时器	定时器外设地址 Timer
0x4000000C	外部 LEDs	0bit: LED 0 1bit: LED 1 7bit: LED 7
0x40000010	外部 SWITCH	0bit: Switch 0 1bit: Switch1 7bit: Switch7
0x40000014	七段数码管	0bit: CA 1bit: CB 7bit: DP 8bit: AN0 9bit: AN1 10bit: AN2 11bit: AN3
0x40000018~0x40000023	UART	UART 外设地址

注：对于 Altera DE2 平台，需额外使用示例代码 `digitube_scan.v` 将硬件上的非扫描模式的七段数码管转换为扫描模式。

- F. 提供一个定时器外设，可以根据设定周期产生外部中断，通过该定时器触发 7 段数码管的扫描显示。

地址范围	功能	备注
0x40000000	定时器 TH	每当 TL 计数到全 1 时，自动加载 TH 值到 TL
0x40000004	定时器 TL	定时器计数器，TL 值随时钟递增
0x40000008	定时器控制 TCON	0bit: 定时器使能控制, 1-enable, 0-disable 1bit: 定时器中断控制, 1-enable, 0-disable 2bit: 定时器中断状态

定时器软件操作流程:

- i. 关闭定时器, TCON 写入 0;
- ii. 设置定时器周期, TH 取值决定定时器的计数周期;
- iii. 设置定时器 TL 为 0xFFFFFFFF;
- iv. 启动定时器, TCON 写入 3.

定时器中断软件服务程序流程(此时处理器处于内核态,监督位为1'):

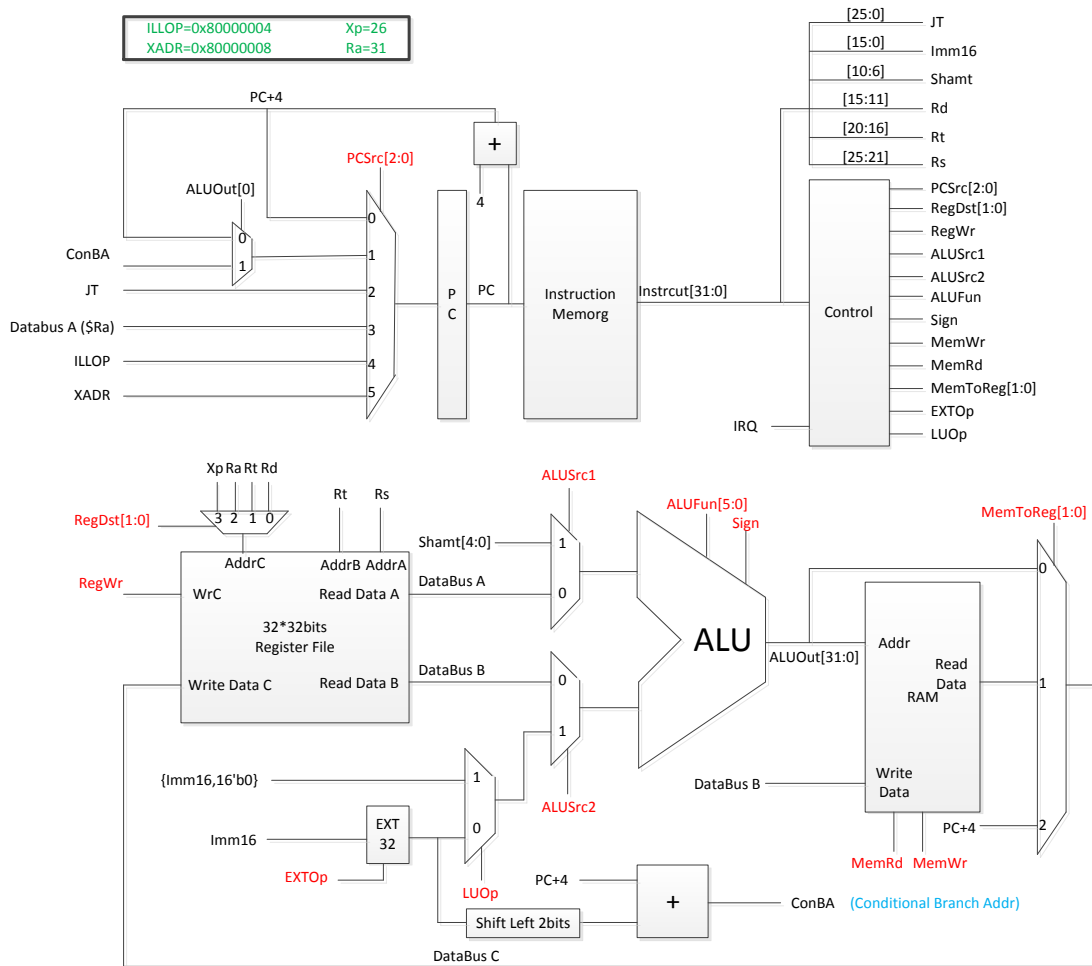
- i. 定时器中断禁止,同时中断状态清零,TCON 的 1-2bit 清零, $TCON \&= 0xfffffff9$;
- ii. 保护现场;
- iii. 中断处理代码;
- iv. 恢复现场;
- v. 使能中断, TCON 的 1bit 置 1, $TCON |= 0x00000002$;
- vi. 退出中断服务程序, 跳转到中断发生时保存的断点地址处继续执行 (\$26) .

- G. 根据春季学期设计的 UART 实现一个 MIPS 的 UART 外设，波特率固定在 9600，起始位 1 位，停止位 1 位，无校验位。访问方式可以采用**中断方式**也可以采用**轮询方式**，建议采用轮询方式会比较简单。**寄存器根据设计自行决定**，参考实现的寄存器映射如下：

地址范围	功能	备注
0x40000018	串口发送数据 UART_TXD	串口发送数据寄存器，只有低 8bit 有效； 对该地址的写操作将触发新的 UART 发送
0x4000001C	串口接收数据 UART_RXD	串口接收数据寄存器，只有低 8bit 有效
0x40000020	串口状态、控制 UART_CON	0bit: 发送中断使能, 1-enable, 0-disable 1bit: 接收中断使能, 1-enable, 0-disable 2bit: 发送（中断）状态，每当 UART_TXD 中的数据发送完毕后该比特置‘1’，当执行对该地址的读操作后，将自动清零 3bit: 接收（中断）状态，每当 UART_RXD 中已经接收到一个完整的字节时该比特置‘1’，当执行对该地址的读操作后，将自动清零

		4bit: 发送模块状态, 0-发送模块处于空闲状态, 1-发送模块处于发送状态
--	--	--

H. 单周期处理器可以参照下面的结构实现，也可以根据讲义或者扩充后的指令集自行设计实现。



- 1) PCSrc, PC 的值可以有 6 种选择：正常执行时，PC+4；条件分支时，ConBA 或者 PC+4，根据条件分支中的条件判断结果而定（ALUOut[0]），ConBA 来自于 PC+4 与指令中 16 位立即数左移 2 位后的数值之和；直接跳转时，JT，其值来自于 J 型指令中的 26 位地址；寄存器跳转时，Databus A，其值取自 \$Ra；发生中断时，ILLOP（常量 0x80000004）；发生异常时，XADR（常量 0x80000008）。
- 2) RegDst，写入寄存器的地址可以有 4 种选择：R 型指令中，指令中的 Rd；I 型指令中，指令中的 Rt；JAL/JALR 指令中，来自 Ra（常量，31）；异常处理时，Xp（为简化处理，这里指定为常量 26，即发生中断或者异常时，返回地址将保存到 \$26 寄存器中）。
- 3) EXTOp，根据指令产生 16 位立即数的 32 位扩展，无符号扩展和有符号扩展。
- 4) LUOp，根据指令 LUI，将 16 位立即数装入 32 位数的高 16 位。

- 5) ALUSrc1、ALUSrc2，根据指令决定参与 ALU 运算的两个操作数。
- 6) ALUFun，ALU 运算功能控制，同前面的 ALU 设计中要求。
- 7) Sign，决定 ALU 运算的性质，有符号数还是无符号数。
- 8) MemWr、MemRd，数据 memory 读写控制信号。
- 9) MemToReg，根据指令决定写入寄存器的数据来源，有 3 种选择：运算指令，ALU 的输出将被写入寄存器；Load 指令，数据存储器中的数据将被写入寄存器；JAL/JALR 指令或者异常中断过程中，PC+4 将被写入寄存器中。
- 10) PC 的最高位 PC[31]为监督位。当该位为‘1’时，处理器处于内核态，此时异常和中断被禁止；当该位为‘0’时，处理器处于普通态，此时允许发生中断和异常。注意 PC[31]不能作为地址最高位去索引指令存储器，取指令时应当固定将地址最高位置零。只有 RESET、异常、中断等有可能将 PC[31]设置为‘1’，其他指令不能设置该位为‘1’，JR 和 JALR 指令可以使监督位清零。
 - 在处理器复位后，PC 中的值应该为 0x80000000；
 - 发生中断时，PC 中的值应该为 0x80000004；
 - 在发生异常时，PC 中的值应该为 0x80000008；
 - PC+4 逻辑电路实现时应该保证 PC[31]不变；
 - 分支语句和 J、JAL 语句不应该改变 PC[31]；
 - 当执行 JR、JALR 指令时，PC[31]的值由跳转地址（\$Ra）中的第 31 位（最高位）决定。
- I. 设计一个简单的编译程序，能够将简单的汇编代码编译为机器码。
- J. 设计一个计算两个整数的最大公约数的汇编程序，使用设计的编译程序得到机器码，要求通过 UART 输入两个 8bits 的操作数，通过七段数码管显示十六进制的操作数（必须通过定时器中断以扫描的形式进行数码管显示，七段译码采用软件译码，定时中断频率为扫描频率），通过 8 个 LEDs 显示计算结果并通过 UART 输出，自行设计接口逻辑。

3. 在单周期 MIPS 处理器的基础上，设计一个 5 级流水线的 MIPS 处理器（30%）。

- A. 采用如下方法解决竞争问题：
 - i. 采用完全的 forwarding 电路解决数据关联问题。
 - ii. 对于 Load-use 类竞争采取阻塞一个周期+Forwarding 的方法解决
 - iii. 对于分支指令在 EX 阶段判断（提前判断也可以），在分支发生时刻取消 ID 和 IF 阶段的两条指令。
 - iv. 对于 J 类指令在 ID 阶段判断，并取消 IF 阶段指令。
- B. 将计算最大公约数的程序在流水线 MIPS 处理器中正常运行。