

数字逻辑与处理器基础实验

32 位 MIPS 处理器设计实验报告

王晗

(2013011076)

July 19, 2015

Date Performed: July 15, 2015

Partners: 耿天毅 (2012011119)

陈志杰 withdrawn

1 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。

2 设计方案

2.1 总体结构

由于这次实验涉及的功能较多，我们将完整的 CPU 分成多个模块。指令存储器、寄存器堆、控制器、ALU 控制器、ALU、数据存储器、UART 等功能单元均在单独的 Module 中实现。其中指令存储器、寄存器堆、控制器、ALU 控制器、ALU 等单元在 Single Cycle Core 中实例化，作为单周期处理器的核心；数据存储器、UART 和定时器、LED、七段数码管、开关在 Peripheral 中实现，作为处理器的外设。处理器核心和外设在顶层模块中实例化，互相通信。

单周期 CPU 模块的结构关系如下图所示：

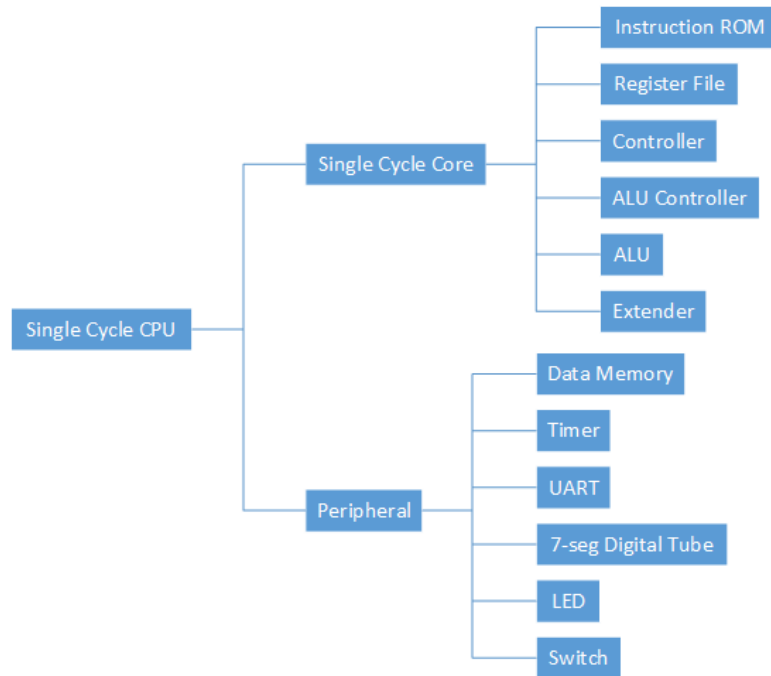


Figure 1: 单周期处理器结构

对于流水线 CPU，我们还在 Pipeline Core 中加入了流水线寄存器、冒险检测单元、数据转发单元：

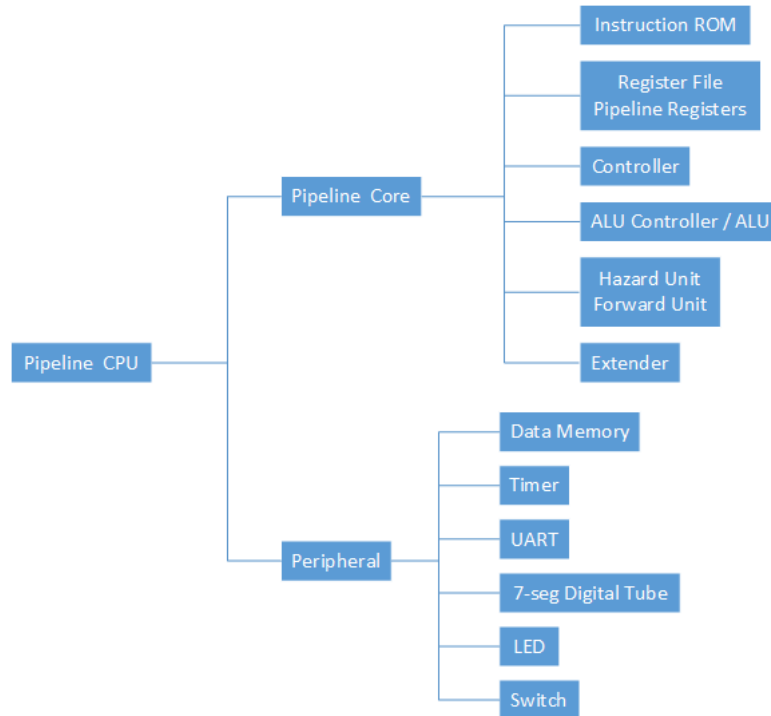


Figure 2: 流水线处理器结构

2.2 ALU ¹

ALU 模块的结构如图所示，输入两个操作数 A、B 和控制信号 ALUFun、Signed，在 ARITH 子模块中做加减法运算，CMP 子模块根据 ARITH 模块的输出进行比较判断，LOGIC 和 SHIFT 模块分别进行逻辑运算和移位运算，ALUFun 的最高两位用于控制多路选择器的输出。

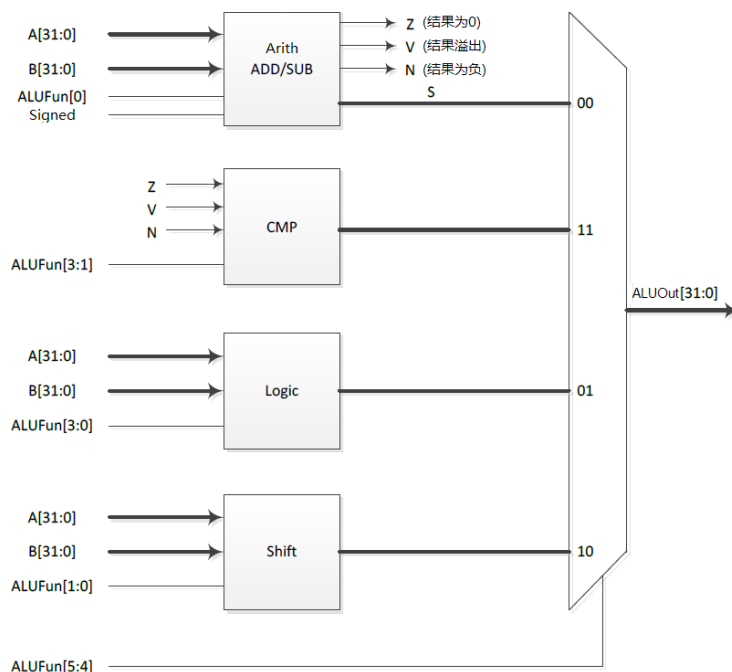


Figure 3: 流水线处理器结构

ARITH 模块 ARITH 模块中包括减法和加法两个模块，加法模块直接通过 + 号运算，减法模块先对第二个操作数取补码，再调用加法模块做加法运算。Overflow 和 Negative 信号的产生是 ALU 中的难点：

ADD		
	Overflow	Negative
Signed	pos+pos=neg neg+neg=pos	neg+neg, pos+pos: MSB of A otherwise: MSB of S
Unsigned	big+big big+small=small small+big=small	0

Figure 4: ADD 中的 Overflow 和 Negative

¹ 原作者：陈志杰；修改：王晗

其中 pos 为正数，neg 为负数，big 为 MSB=1 的无符号数，small 为 MSB=0 的无符号数。

SUB		
	Overflow	Negative
Signed	pos-neg=neg neg-pos=pos	pos-pos, neg-neg: MSB of S otherwise: MSB of A
Unsigned	small-big big-big, small-small: MSB of S	big-small: 0 small-big: 1 otherwise: MSB of S

Figure 5: SUB 中的 Overflow 和 Negative

图中的缩写含义同上。

CMP 模块 CMP 模块直接根据 ARITH 模块产生的 Zero, Overflow, Negative 进行关系判断。

LOGIC 模块 LOGIC 模块直接根据 ALUFun[3:0] 指定的逻辑运算进行运算。

SHIFT 模块 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，分别用 Shamt 的每一个 bit 位控制，组合产生最后的运算结果。

2.3 寄存器堆、指令存储器、数据存储器 and 外设²

寄存器堆 直接采用 reg [31:0] RF_DATA[31:1] 实现，注意 RF_DATA[0] 不存在，读取时直接返回 0。

指令存储器 将机器码以十六进制文本的形式存放在 .rom 文件中，使用 \$readmemh 系统任务初始化一个大小为 256words 的只读存储器。

数据存储器 由于数据存储器容量设计为 256words，因此寻址时只根据 address[9:2] 寻址。

另外，0x40000000 开始的地址用于外设编址，因此数据存储器不对 0x40000000 开始的地址进行读写操作。

其他外设 定时器、LED、Switch 参考老师提供的样例代码直接在 Peripheral.v 中实现，UART 使用春季学期第四次实验的 UART 发送和接收模块，将发送模块中 Tx_Status 的定义取反，即 1 表示发送端忙碌。UART 的控制同样在 Peripheral.v 中实现，当 0x40000018 写入要发送的数据时，串口控制器自动产生一个发送使能信号。

²作者：王晗

2.4 控制器和 ALU 控制器³

控制单元采用两级控制的实现方法，在主控制器中根据 OpCode 和 Funct 产生 PCSrc、RegWrite、RegDst、MemRead、MemWrite、MemToReg、ALUSrc1、ALUSrc2、ExtOp、LuOp、ALUOp 等控制信号，其中 ALUOp 经过 ALU 控制器进一步解码生成 ALUFunc、Signed 信号，控制 ALU 的运算，其余信号控制数据通路中的多路选择器。控制器还产生了 UndefinedInst 信号，用于识别未定义指令的异常。

在单周期 CPU 中, PCSrc 信号位宽为 2, 分别指示从 PC+4、Branch Target、Jump Target、Jump Register 取出下一个指令地址, 当发生中断或异常时, 由 Single-Cycle Core 直接跳转至中断或异常服务程序入口。在流水线 CPU 中, 为了方便流水线寄存器操作, 将 PCSrc 信号位宽扩展至 3, 当发生中断或异常, PCSrc 变为 100 或 101, 指示中断或异常服务程序入口。

2.5 单周期数据通路⁴

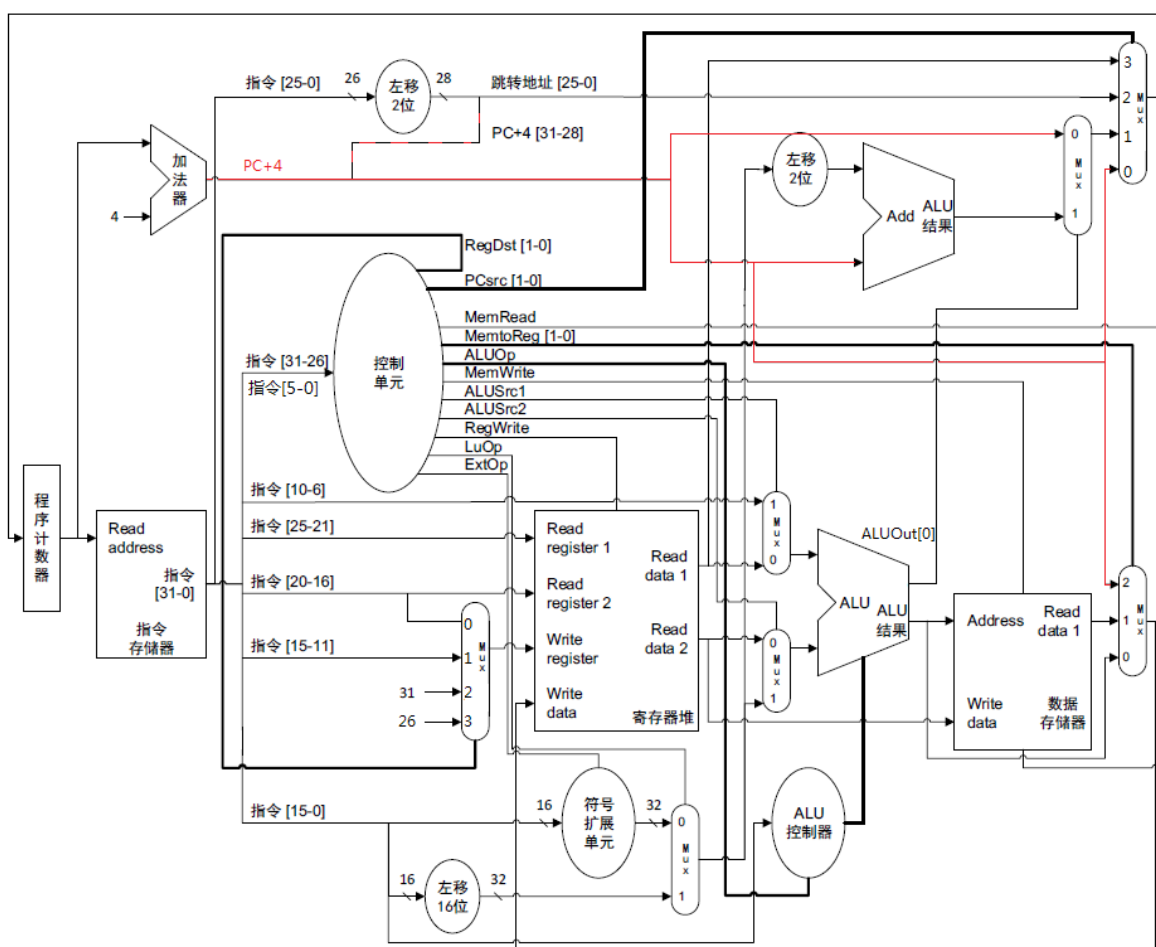


Figure 6: 单周期数据通路

³作者：王晗

⁴作者：王晗

2.6 流水线数据通路⁵

2.7 汇编代码⁶

2.8 汇编器⁷

3 关键代码及文件清单

4 仿真结果及分析

5 硬件调试情况

硬件调试情况

6 心得体会

a. 我没有

b. 心得

c. 体会

⁵作者：王晗

⁶作者：耿天毅

⁷作者：耿天毅