南京信息工程大学

应用技术学院

课程报告

课程名称： 电子技术基础

实验名称：基于logisim平台的

MIPS指令CPU设计与实现

班 级： 19软工3班

姓 名： 陈 冰

指导教师： 王 其

提交日期： 2021年12月23日

成 绩：

# 背景

单周期处理器是指所有指令在一个始终周期内完成的处理器。景尽管不同指令执行时间不同，对于单周期处理器而言，时钟周期必须设计程对所有指令都等长。单周期处理器中，任何资源都不能被重复使用，任何需要多次使用的资源都需要设置多个，否则会发生资源冲突，这是在设计单周期处理器过程中的重要环节。取指令与执行阶段均需要使用存储器，所以单周期处理器智能采用指令存储器和数据存储器分离的结构。

# 1运算器设计

#### 1.1可控加法器设计

##### (1)一位全加器

多位定点加法可以由一位加法器通过进位链串联实现，实现基础部件为一位全加器，输入两个相加数Xi，Yi，低位进位输入Ci-1，输出为和数Si，高位进位输出Ci，对应逻辑表达式如下：

具体逻辑电路如图1所示，假设所有逻辑门的延迟都是T，则一位加法器的总时间延迟为3T。

图示

描述已自动生成

图1：一位加法器

##### (2)多位串行加法器

将n个一位全加器的进位链串联即可得到n位加法器，如图2所示为一个八位串行加法器，由于补码符号位也可以参与运算，所以此电路既可以用于有符号数的运算，也可以用于无符号数的运算，但是二者在溢出检测上有所区别。假设相加数符号位分别为，，和符号位，则溢出检测信号Overflow的逻辑表达式为：

如果已知符号位进位和最高位，可以利用如下表达式进行溢出检测：

图示

描述已自动生成

图2：八位串行加法器

#### 1.2快速加法器设计

在之前章节已经对多位串行加法器逻辑进行描述，全加器高位运算依赖于低位运算的进位输入，所以全加器不能并行运行，多位加法器整体延迟与位数成线性关系，位宽较大时性能较差。

假设进位生成函数，进位传递函数，则加法器进位逻辑变换为以下公式：

对以上公式进行重复迭代，可以得到如下公式：

由此可知，高位进位输出可以直接由已知变量，和经过电路运算所得到，根据此公式可以利用额外电路计算进位信号，提前产生各位加法运算所需要的所有进位输入，再利用即可得到最终的和数，这就是先行进位的基本原理，此处进位信号也要的产生也是需要时间延迟的，n越大则延迟越大，未来简化电路，通常按照四位一组进行先行进位，根据公式可得到如下逻辑表达式：

假设具有2~5个输入逻辑门的延迟，则该电路的总延迟为2T。生成所有的P、G输入需要一级电路延迟，最终生成信号延迟为3T，当所有进位信号产生后，再增加一级异或门延迟即可完成所有位的和数，完成多位加法运算。最终四位全加器延迟为4T，相比四位串行加法器有较大的性能提升，四位快速加法器的逻辑图如图3所示。

手机屏幕的截图

中度可信度描述已自动生成

图3：四位快速加法器

#### 1.3 32位ALU设计

算术逻辑运算单元(ALU)是计算机的核心部件，实现的基本功能包括加,减、乘、除等算术运算以及与、或、非等逻辑运算，实现上述算术以及逻辑运算功能的部件就是构造ALU的基本逻辑单元部件。ALU模块封装示意图如图4所示其中。

图示

描述已自动生成

图4：ALU封装

X\Y为输入数，CF为运算选择控制端，通过该控制端选择ALU进行不同的算术逻辑运算， S为进位输出端, R为运算结果，R2为运算溢出标志，MIPS虽然没有状态寄存器，但也会产生溢出信号，例如MIPS中add指令就会根据假发运算是否溢出决定数据通路。ALU子电路调用如图5所示。

图示

描述已自动生成

图5：ALU子电路调用

# 2存储器设计

#### 2.1存储器的扩展

##### (1)位扩展（数据总线扩展、字长扩展）

当储芯片的数据位小于CPU对数据位的要求时,可采用位扩展方式解决此类问题。此时，所有存储芯片的地址线、读写控制线并联后与CPU的地址线和读写控制线连接,各存储芯片的数据数据总线汇聚成更高位宽的数据总线与CPU的数据总线相连,所有存储芯片的片选信号并联后与CPU连接。 假设一个存储系统容量为N位，若使用K位的芯片，且K<N，则共需要（N%K）个芯片实现。

##### (2)字扩展（地址总线扩展、字数扩展）

字扩展也称容量扩展,当存储芯片的存储容量不能满足CPU对存储容量的要求时，可采用字扩展方式来扩展存储器。字扩展时,所有存储芯片的数据总线、读写控制线各自并联与CPU数据总线、读写控制线相连，各存储芯片的片选信号由CPU高位多余的地线译码产生，CPU给出一个地址时,只有一个存储芯片工作，具体哪一个芯片工作由存储系统地址高2位译码决定，假设一个存储系统容量为M,若使用容量为l位的芯片,且l<M,则共需要(M%l)个芯片。

##### (3)字位同时扩展（综合扩展）

当存储芯片的数据位和存储容量均不能满足存储器的数据位和存储总容量要求时，采用字位同时扩展的方式来组织存储器。其中，通过位扩展满足数据位的要求，通过字扩展满足存储总容量的要求。假设一个存储系统容量为M×N位,若使用l\*K位的芯片,且l<M，K<N,则共需要(M/l)\*(N/K)个芯片。

##### 2.2MIPS RAM设计

计算机主存储器通常既能按位访问，也能按半字节访问，还能按字访问。例如，x86中MOV指令就提供如下三种形式：

**按字节访问：mov ah,[200]**

**按16位访问：mov ax,[200]**

**按32位访问：mov eax,[200]**

需要四个4KB\*8位的RAM组件进行扩展，设计完成能按位访问，也能按半字节访问，还能按字访问的32位存储器，封装原件如图6所示。

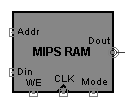


图6：MIPS RAM封装

##### 2.3MIPS寄存器文件设计

寄存器文件使CPU中通用寄存器的集合，以MIPS为例子，MIPS32指令支持32个32位通用寄存器，32个通用寄存器均包含在寄存器文件中，其中每个寄存器可以通过对应寄存器编号进行访问，类似于一个具有多个地址端口和多个数据端口的高速存储器。图7为MIPS寄存器文件的具体封装形式。

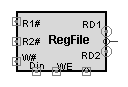


图7：MIPS寄存器文件封装

# 3处理器整体架构设计

#### 3.1主要功能部件

CPU主要功能部件包括：程序计数器PC、指令寄存器IM、数据存储器DM、立即数扩展器、地址转移逻辑NPC、运算器ALU。整体架构如图8所示。

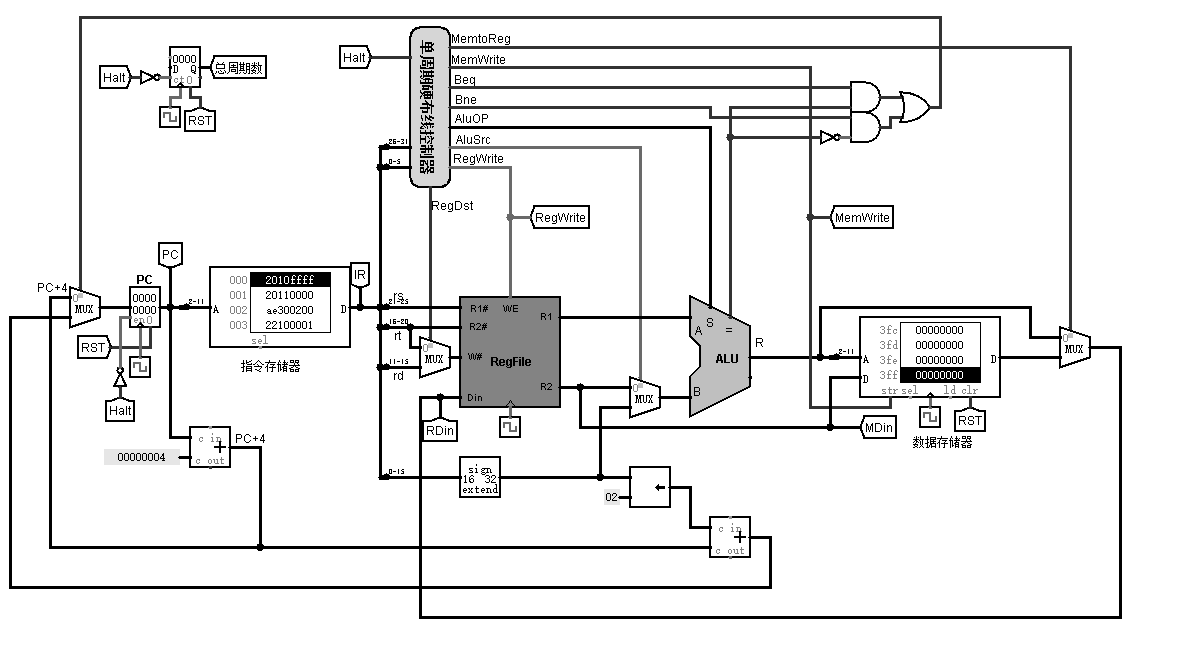


图8：MIPS架构处理器顶层视图