数字部件设计 Project

单周期 CPU 电路设计

一、项目介绍、要求和提示

在这个项目中，我们最终需要完成一个能够实现指定 MIPS 指令 功能的简单单周期 CPU 的电路设计。附录中，提供了关于这两者的介绍和一个单周期 CPU 电路设计的实例。当然，我们需要完成的工作比这个实例还要简单得多，因为我们只要求完成的单周期 CPU 接收并正确处理 MIPS 指令的一个很小的子集，同时也减少了对电路中存储设备规模的要求。

对于这个单周期 CPU 电路，我们有一些具体的要求：

1. 该 CPU 需要完成的指令有以下 13 个：

   add，sub，and，or，slt，addi，andi，ori，slti，sw，lw，j，nop

   关于每条指令对应的格式和完成的功能，在 MIPS 指令集中有具体说明。

2. 该 CPU 的指令序列输入方式不做特殊要求，即允许在指令寄存器的代码中写定输入，或者在特定时刻、特定信号的指示下写入指令寄存器等。

3. 项目中设计的存储空间通过数组来得到，考虑到实际情况，一次性运行的指令将不会大于 16 条，Memory 的大小不会使用超过 256byte，Registers 的大小不会超过 64byte，具体实现时，申请的空间务必不要小于这些值。

4. 由于受到采用的指令的限制，寻址模式中，除了 PC 相对寻址不做要求外，其他寻址模式在设计中，都必须得到支持。

5. 最后的波形仿真应当采用功能仿真，且所有存储器件中的数据都应当被显示。

为了保证大家的项目进度，同时为了让大家对层次化的设计有进一步的体验，我们把整个设计拆成了两个部分：基本部件的设计和通过利用基本部件来完成单周期 CPU 的功能。

提示：

在前一个部分中，大家首先需要考虑，在单周期 CPU 中可能需要哪些有一定功能的元件，附录中的介绍应当能够给大家一些提示，但是请大家不要拘泥于附录中的设计，而是多进行一些自己的思考。在考虑元件的实现时，大家应当综合考虑各元件将来可能出现的配合、元件的复用性、运行性能等因素来权衡自己采用的实现方式。

在后一个部分中，最重要的是控制单元的设计，该单元因为涉及到具体的 MIPS 指令编码，在前一个部分不要求实现。控制单元如何通过控制已经完成的单元的行为来完成指令的功能？控制单元到底需要输出那些控制信号？这些信号在时序上是否有其特定的要求？这些问题都是需要反复思考的，有时候可能还会需要一些小的技巧，相信能够有力的深化同学们对数字电路设计的认识。

设计的正确性，将由功能仿真的结果来检验。一方面，大家应当自行进行功能仿真的测试，并将结果提交；另一方面，我们也会对大家提交的程序进行相关测试。对于最终的简单单周期 CPU 设计，我们会在后面给出一些测试指令，当然我们最后的测试不会拘泥于这些提供的指令。

二、项目提交

1. 最终提交时间为期末考试之前，提交内容包括：所有基本部件的程序代码（以 Project 的形式，不能只有.v 文件），基本部件的功能仿真波形图以及对每个部件及波形图的说明和文件清单，所有内容应当被打成一个压缩包上传。

注意事项：

a) 提交的部件由个人需求决定，可以使用直接的运算符。

b) 对部件的说明包括但不限于部件功能说明、部件设计思路以及部件会在单周期 CPU 的那些地方能够用到；对波形图的说明包括但不限于对波形图的设定如何能测试出部件设计的正确性以及仿真结果的解释。

c) 推荐部件集中在一个.v 文件中完成，但每个部件应当有它独立的波形图仿真图和相关说明。

2. 单周期 CPU 设计的提交内容包括：单周期 CPU 的程序代码（以 Project 的形式，不能只有.v 文件），单周期 CPU 的功能仿真波形图，整个 Project 的说明和设计文档和文件清单，所有内容应当被打成一个压缩包上传。

注意事项：

a) 单周期 CPU 如果用到了第一部分提交的部件，基本的或、与、非门以及数位扩展之外的元件，必须在文档中予以说明，并补充该元件的相关文档。

b) 在说明文档的内容中包括但不限于对功能仿真波形图的介绍和说明，以及指令处理的具体过程，以及对 CPU 所能完成的所有功能和相关性能。

c) 在设计文档的内容中包括但不限于 CPU 的电路图、CPU 如何实现其功能、怎样通过改进设计提高其性能以及自己在项目中的体会和感受。

三、项目的评分

程序会考察程序的正确、规范等性质；波形图会考察其代表性和体现出的程序性能；文档会考察文档内容是否完整，是否能体现出正确的设计思路，以及是否对单周期 CPU 的实现进行了自己的思考等。本 project 对文档有较高的要求，如果有同学能够提出更好的改进意见和建议，请在文档中特别注明，我们会酌情予以加分。

附录

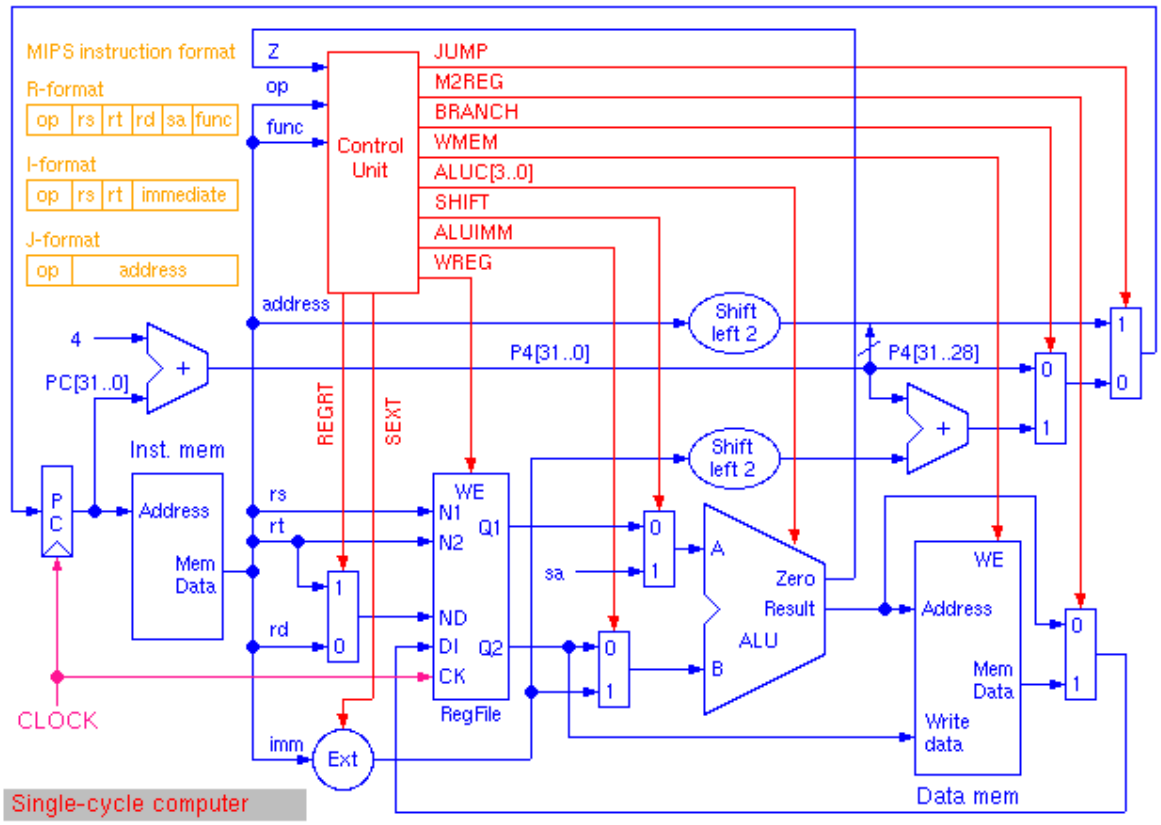
I单周期 CPU 简介

单周期 CPU 的特点是每条指令的执行需要一个时钟周期，只有一条指令执行完后才能执行下一条指令，由于每个时钟周期的时间长短是相同的，因此，在确定时钟周期的时间长度时，要以指令集中最复杂的指令需要执行的时间为第一考虑。

单周期 CPU 使用 PC 作为地址从程序存储器中获得指令，根据指令的意义完成相应操作，在时钟上升沿将结果写入寄存器，与此同时，PC 也一并被更新。

下图是一个简单的单周期 CPU 的电路图。主要元件有 PC、程序存储器，寄存器堆，ALU，数据存储器和控制部件等，所有控制信号简单说明如下：

1. JUMP ：选择跳转目标地址 / 由 BRANCH 选择的地址
2. M2REG：选择存储器数据 / ALU 输出数据
3. BRANCH：选择转移目标地址 / 下一地址
4. WMEM：是 / 否写入存储器
5. ALUC：ALU 操作码
6. SHIFT：选择寄存器数据 / 输入的移位数
7. ALUIMM：选择扩展的立即数 / 寄存器数据
8. WREG：是 / 否写入寄存器
9. SEXT：符号扩展 / 0 扩展
10. REGRT：选择 rd / rt



II　MIPS 指令集简介

MIPS（Microprocessor without interlocked piped stage）是世界上很流行的一种 RISC（Reduced Instruction Set Computing）处理器，其机制是尽量利用软件办法避免流水线中的数据相关问题。它最早在 80 年代初期由斯坦福（Stanford）大学 Hennessy 教授领导的研究小组研制出来，之后得到了广泛的应用。MIPS 指令集即指该处理器使用的 RISC 指令集合。

MIPS 指令集具有以下特点：

1. 简单的 LOAD / STORE 结构。所有的计算类型的指令均只涉及寄存器堆的存取，只有 LOAD 和 STORE 指令涉及存储器访问。

2. 易于流水线 CPU 的设计。MIPS 指令集中所有的指令均为 32 位，而且指令操作码也处于固定位置。

3. MIPS 指令的寻址方式和指令操作上的简易性适合编译器的开发。

MIPS 有 0~31 共 32 个通用寄存器，统称寄存器堆（register file），其中寄存器 0 的内容总是 0。另外，MIPS 还有 32 个浮点寄存器，以及一些专用寄存器，例如 PC（program counter）。

MIPS 支持的数据类型有整数和浮点数，前者可以是 8 位、16 位、32 位或 64 位，后者包括 32 位单精度和 64 位双精度。

MIPS 的指令格式有 3 种：

1. R（register）类型：该类型指令从寄存器堆中读取两个源操作数，计算结果写回寄存器堆。

2. I（immediate）类型：该类型指令使用一个 16 位的立即数作为一个源操作数。

3. J（jump）类型：该类型指令使用一个 26 位的立即数作为跳转的目标地址。

#Bits	[31……26]	[25……21]	[20……16]	[15……11]	[10……6]	[5……0]
R-Type	op	rs	rt	rd	sa	func
I-Type	op	rs	rt	immediate		
J-Type	op	target				

其中，op（opcode）是指令操作码；rs（register source）是源操作数的寄存器号；rd（register destination）是目的寄存器号；rt（register target）作为源寄存器或作为目的寄存器有具体指令决定；func（function）是扩展的操作码；sa（shift amount）由移位指令使用，定义移位位数；immediate 是 16 位立即数；target 由 jump 指令使用，用于产生跳转目标地址。

MIPS 的寻址方式有 5 种：寄存器寻址，立即数寻址，基址偏移量寻址（寄存器内容与常数相加），PC 相对寻址（转移指令计算地址时使用，相对值为指令中的常数），伪直接寻址（在跳转指令中，指令中的 26 位目标地址值与 PC 的高四位拼接，形成 30 位的存储器“字地址”）。

MIPS 的指令种类有 9 种：算术运算（ADD 等）、逻辑运算（AND 等）、数据传送（LW 等）、条件转移（BEQ 等）、无条件跳转（J 等）、特殊指令（BREAK 等）、例外指令（TGE 等）、协处理器指令（LWCz 等）、以及系统控制协处理器指令（MTC0 等）。

更多的关于 MIPS 指令集的内容，可以通过查阅 MIPS 指令集了解。