该项目的目的是设计和验证一个简单的CPU（中央处理单元）。 该CPU具有基本指令集，我们将利用其指令集生成一个非常简单的程序来验证其性能。 为简单起见，我们只考虑CPU，寄存器，存储器和指令集之间的关系。 也就是说我们只需要考虑以下项目：读/写寄存器，读/写存储器和执行指令。

至少有四个部分构成一个简单的CPU：控制单元，内部寄存器，ALU和指令集，这些是我们项目设计的主要方面，并将进行研究。

指令系统

单地址指令格式用于我们简单的CPU设计中。指令字包含两个部分：操作码（操作码），它定义指令的功能（加法，减法，逻辑运算等）; 地址部分，在大多数指令中，地址部分包含要操作的数据的存储位置，我们称之为直接寻址。 在某些指令中，地址部分是操作数，称为立即寻址。

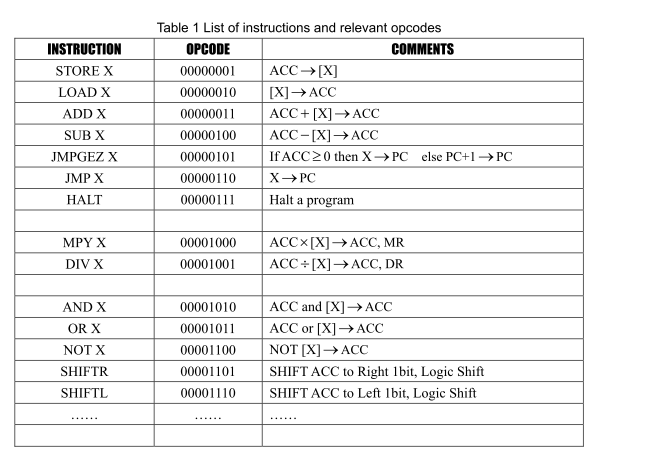
为简单起见，计算机中的内存大小为256\*16。 指令字有16位。 操作码部分有8位，地址部分有8位。 指令字格式可以用图表示



表1列出了相关说明的操作码。

在表1中，符号[x]表示存储器中位置x的内容。 例如，指令字00000011101110012

(03B916 )表示CPU将存储器中位置B916处的字添加到累加器（ACC）中; 指令字00000101000001112（050716）表示ACC（ACC [15]）的符号位是否为0，CPU将使用指令的地址部分作为下一条指令的地址，如果符号位为1，CPU将增加程序计数器（PC）并将其内容用作下一条指令的地址。



在您的设计中必须支持除除法指令之外的所有指令。 如果您可以实现指令更好：DIV X.此外，您可以根据需要设计更多指令。

您必须设计几个程序来测试这些指令。 一个程序作为一个

例：

Calculate the sum of all integers from 1 to 100.

1. programming with C language:

sum=0;

temp=100;

loop :sum=sum+temp;

temp=temp-1;

if temp>=0 goto loop;

end

2. Assume in the memory:

sum is stored at location A4,

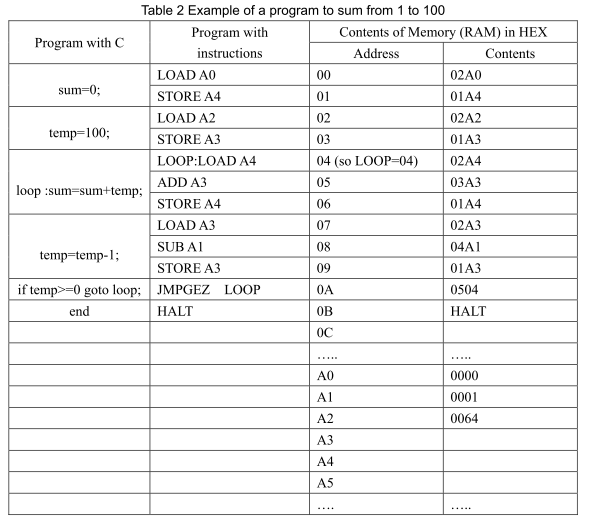
temp is stored at location A3,

the contents of location A0 is 0,

the contents of location A1 is 1,

the contents of location A2 is 100 10 =6416.

我们可以将上面的C语言程序与表1中列出的指令一起翻译成指令程序，如表2所示。



你可以编程使用MPY指令将6乘以5，-6乘以5和-6乘-5，然后检查结果。

注意：所有数据均以2s补码格式表示在内存中。