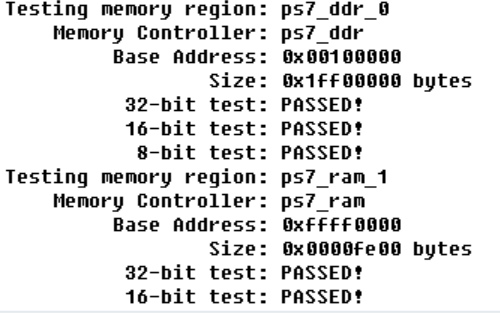
lab1

lab1只用到了ZYNQ板的ps部分，还没有用到pl，也就是FPGA部分。

添加一个名为ZYNQ7 Processing System的IP核。

然后在SDK中添加一个memory test的代码。

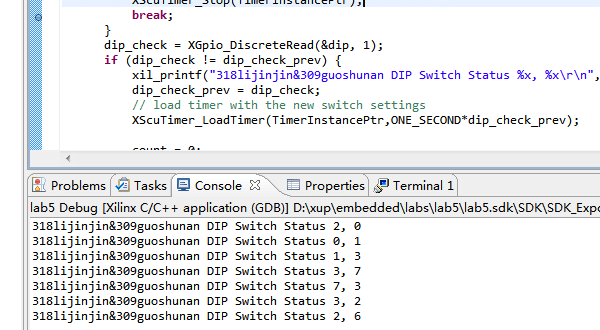


lab2

lab2是5个lab里理解最深入的，因为使用按钮控制小灯的期末项目就是以它为基础的。

lab2基于lab1，在这个实验中我们需要用到板子的pl部分。

添加两个GPIO口，分别是switch和button，都是4bit，使用AXI总线将它们与ps部分连接上。



lab3

lab3没有用到板子，因为它的任务就是创建一个用户自己的IP核。这个custom IP实现了4个LED的功能。

在block design添加了一个BRAM（块存储器）和BRAM控制器。

lab4

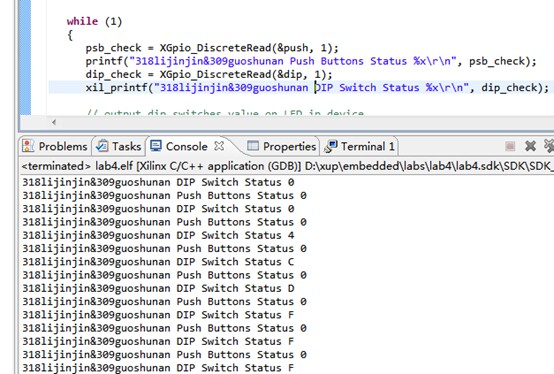
这个实验的结果是按一下板子上的按钮，就可以对应地让zynq板子上自带的led灯亮起来，用到了lab3例创建的IP核。在这个实验中直接注意的地方就是用到了lab3中添加的8K的bram和bram控制器。在这个实验中我们会通过改变一个名叫link script的东西然后把代码放在bram里，运行的时候就从bram中调取。之前代码是放在哪里的呢。

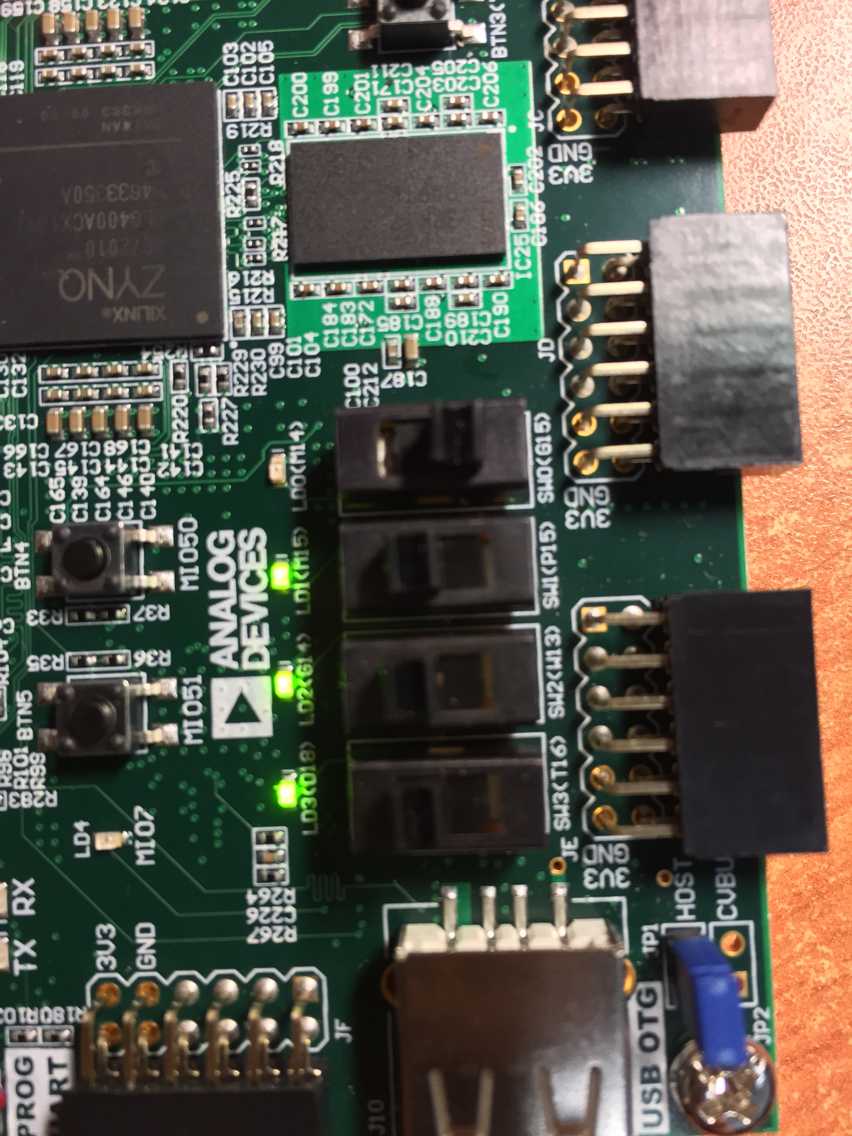
将lab3中创建的ip block添加到系统中。可以看出自己创建的led\_ip

和提供的GPIO很像。

按一下按钮，对应的led灯就会亮起来。这时候代码部分在bram中，而数据和堆栈部分在ddr中（其中ddr就是我们在lab1的时候就已经添加的）。

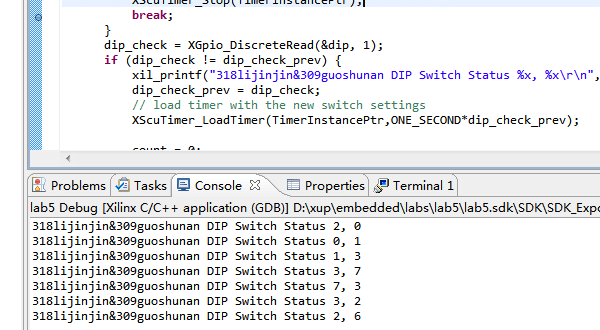
最后，我们做一下改变，将代码部分放入ddr，堆栈部分放入ab3中创建的bram。在改变了设置之后，按一下按钮，led灯亮起来的速度比之前大大下降，照文档的意思这是因为放堆栈的位置发生了改变，现在放在没有cache的bram中，所以运行速度下降。





lab5

出现如DIP Switch Status 5,4的结果。第一个“5”是上一次dip switch的结果，第二个“4”是这次dip switch的结果。



之后对程序进行debug