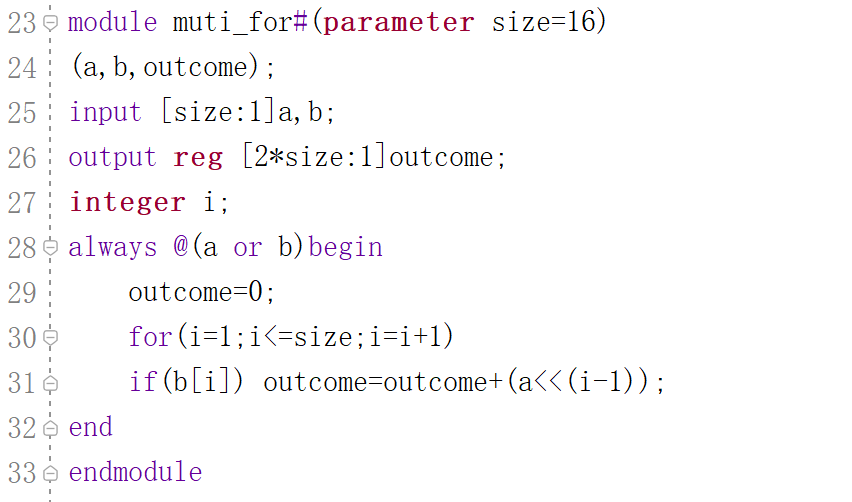
**1.for循环16位乘法器（vivado项目名：homework\_2）**

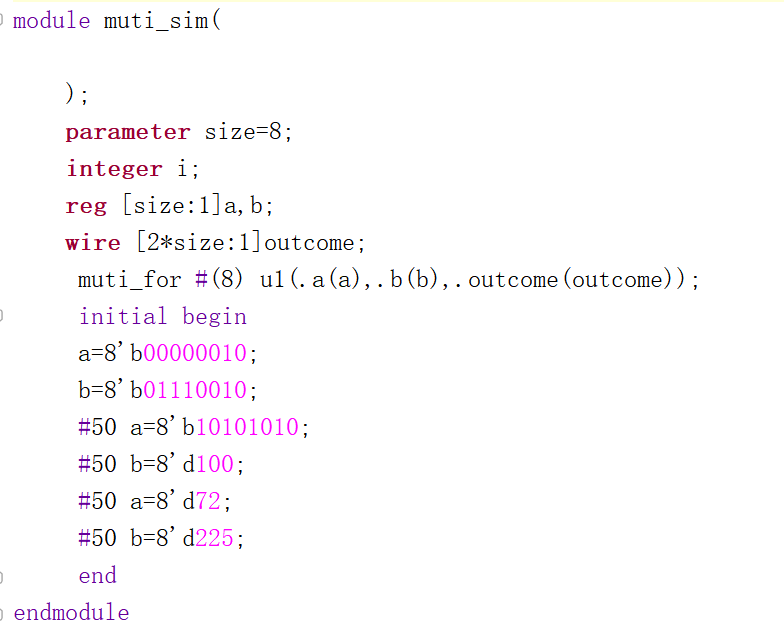
**（1）verilog代码**

乘法器的原理类似十进制多位数的乘法，即分为加法的累加，16位乘法器即分成16个通过在前后补0形成的32位数的加法。在代码设计中，通过移位实现前后补0的操作，通过for循环实现移位和相间的操作。为便于进一步操作和使用，代码中将位宽设为了待定值，默认为16。

仿真文件中，为方便验证计算，将位数设定为8位，并设定了不同的值。



设计文件代码



仿真文件代码

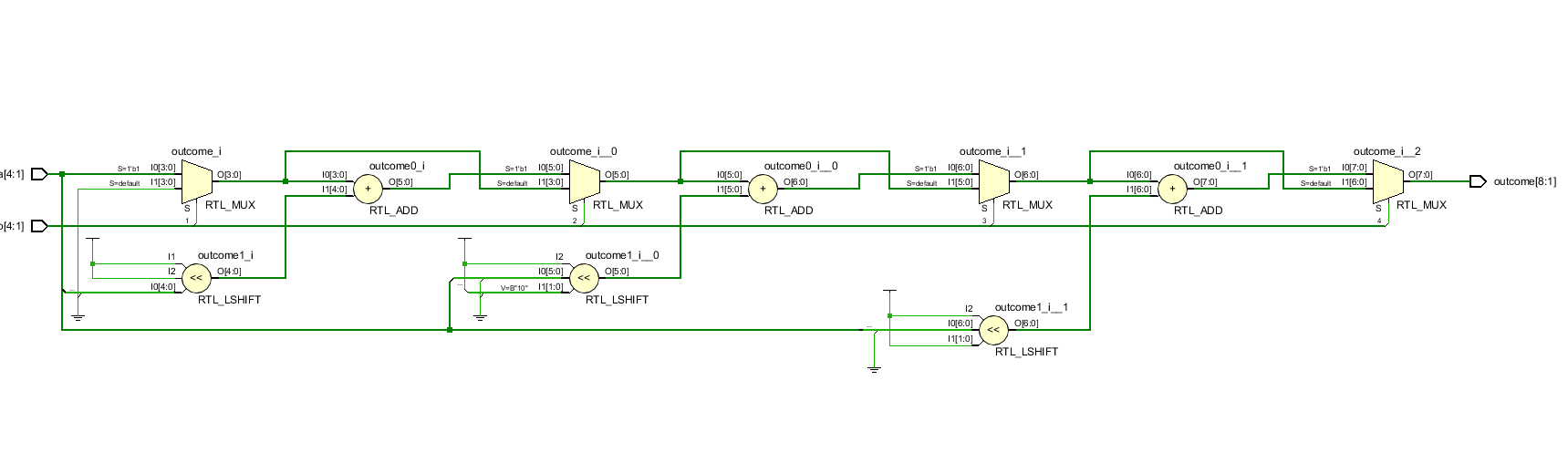
**（2）仿真结果**

仿真结果如下图所示，结果符合相乘的结果。



**（3）电路分析和能耗分析**

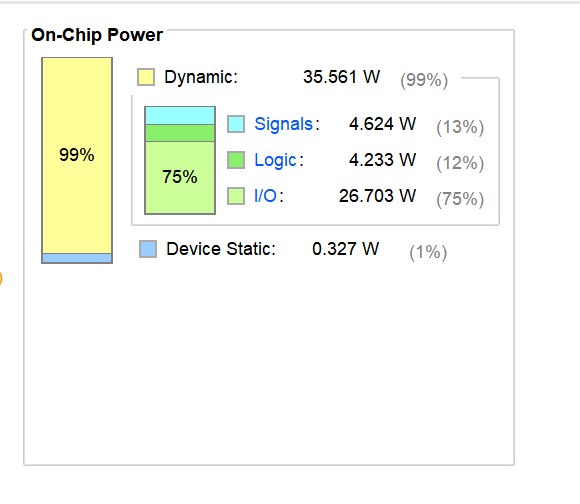
由于16位所形成的电路过于冗长，将默认位宽修改为4位，并利用verilog的RTL分析功能分析代码所生成的电路，结果如下图所示。



RTL电路分析图

由于for循环是串行进行的，只有知道上一次相加的结果才能进行下一次计算，所以电路整体也是线性串行的。16位乘法器的电路即这样的电路继续向前循环4次得到。

利用vivado的功耗分析功能，对代码进行综合、实现后，得到的功耗分析如下图。



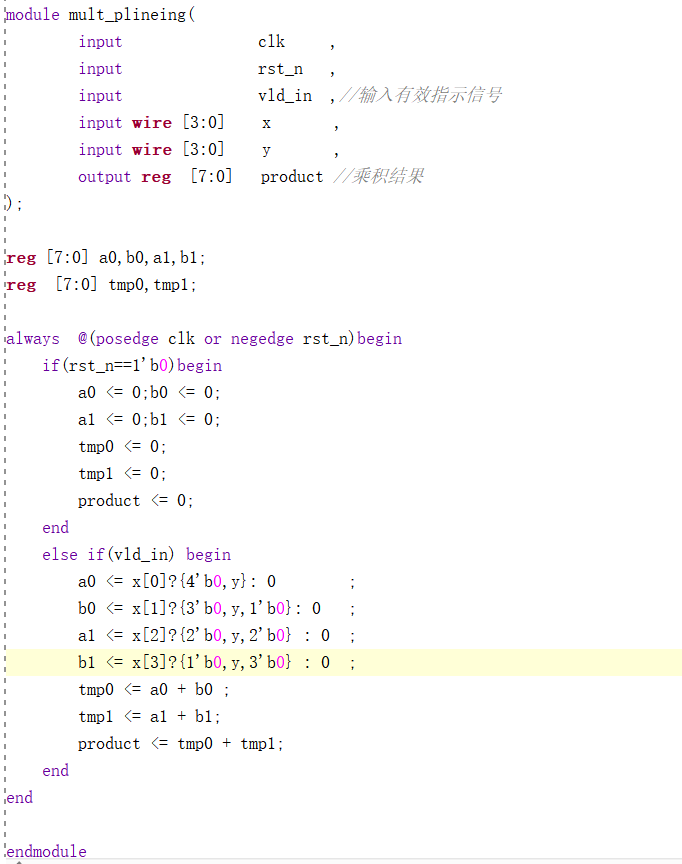
功耗分析

由于此处是对于16位乘法器的功耗分析，输入输出端口有六十多个，所以I/O资源和signals资源消耗占比较大。但同时由于for循环代码电路自身的局限性，执行效率不是很高，所以logic资源消耗也较大。

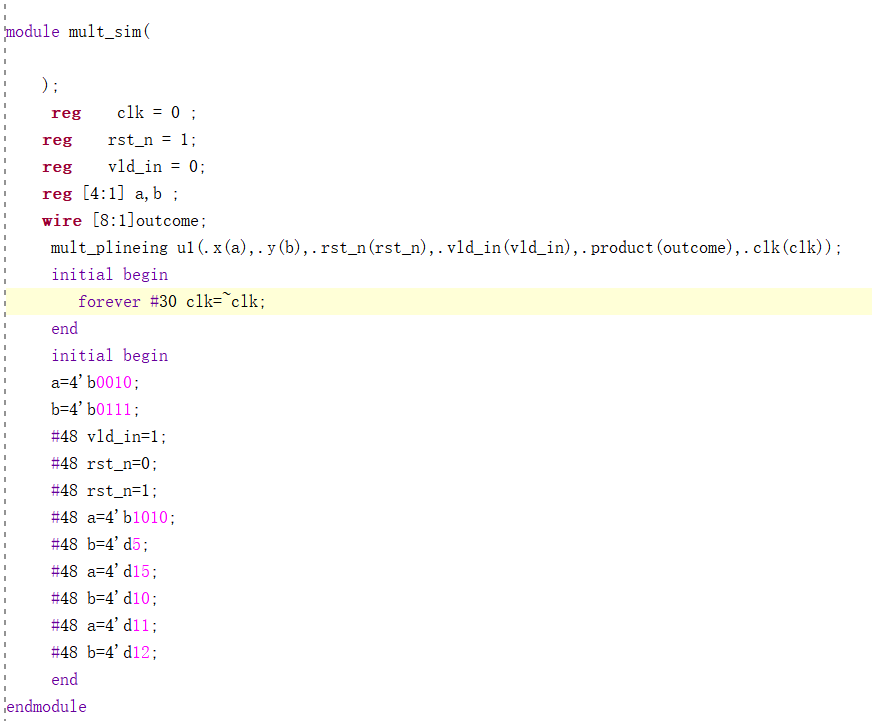
**2.流水线4位乘法器（vivado项目名：homework\_2\_part2）**

**（1）verilog代码**

流水线乘法器实现乘法的基本原理与for循环乘法器原理一致，都是通过加法来实现乘法。流水线的实现方式则是，对于各个补零的相加同时进行。如对于前两个的相加和后两个的相加同时进行，然后再进行相加，同时采用时序逻辑设计，具体代码和仿真文件代码如下。



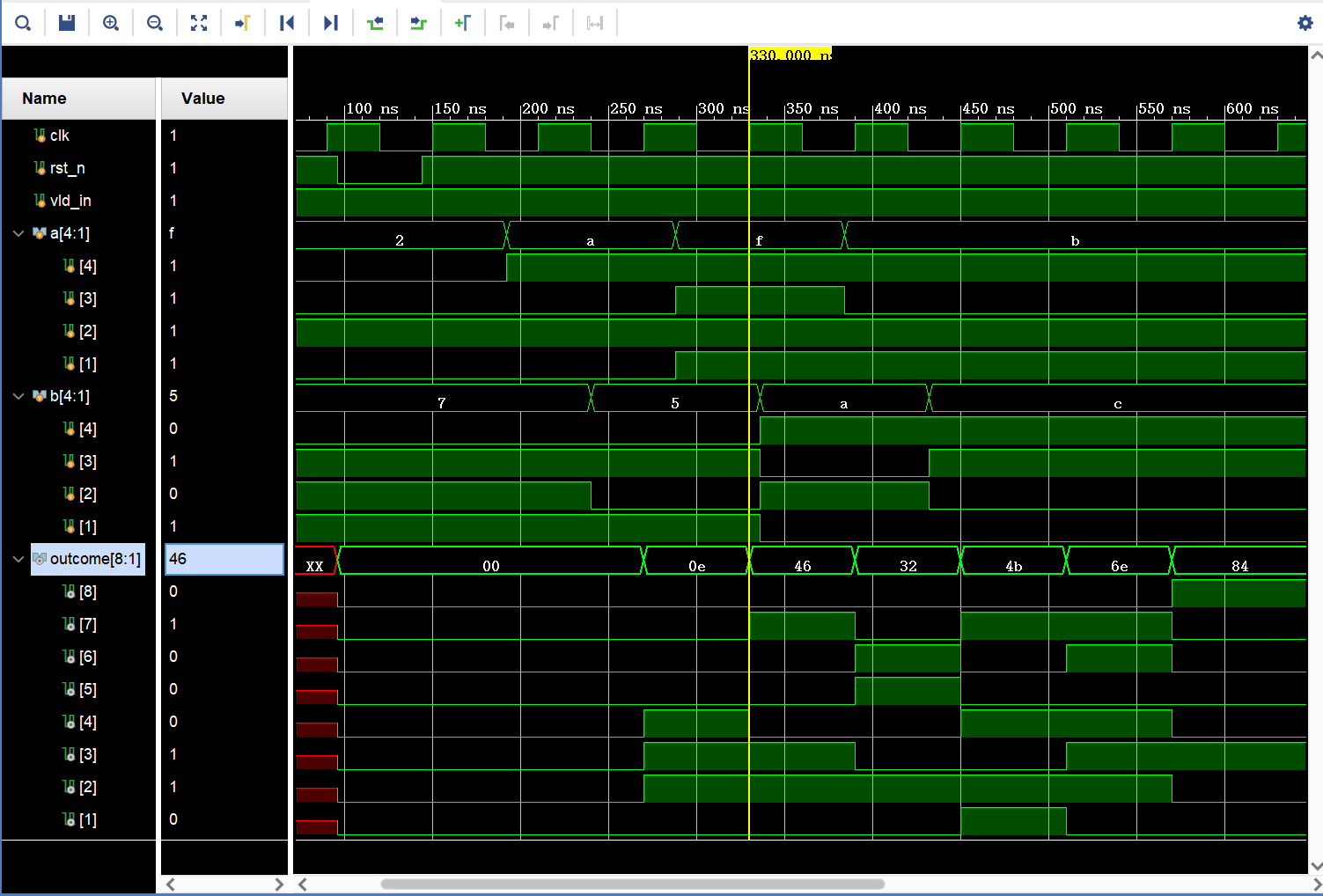
设计文件代码



仿真文件代码

**（2）仿真结果**

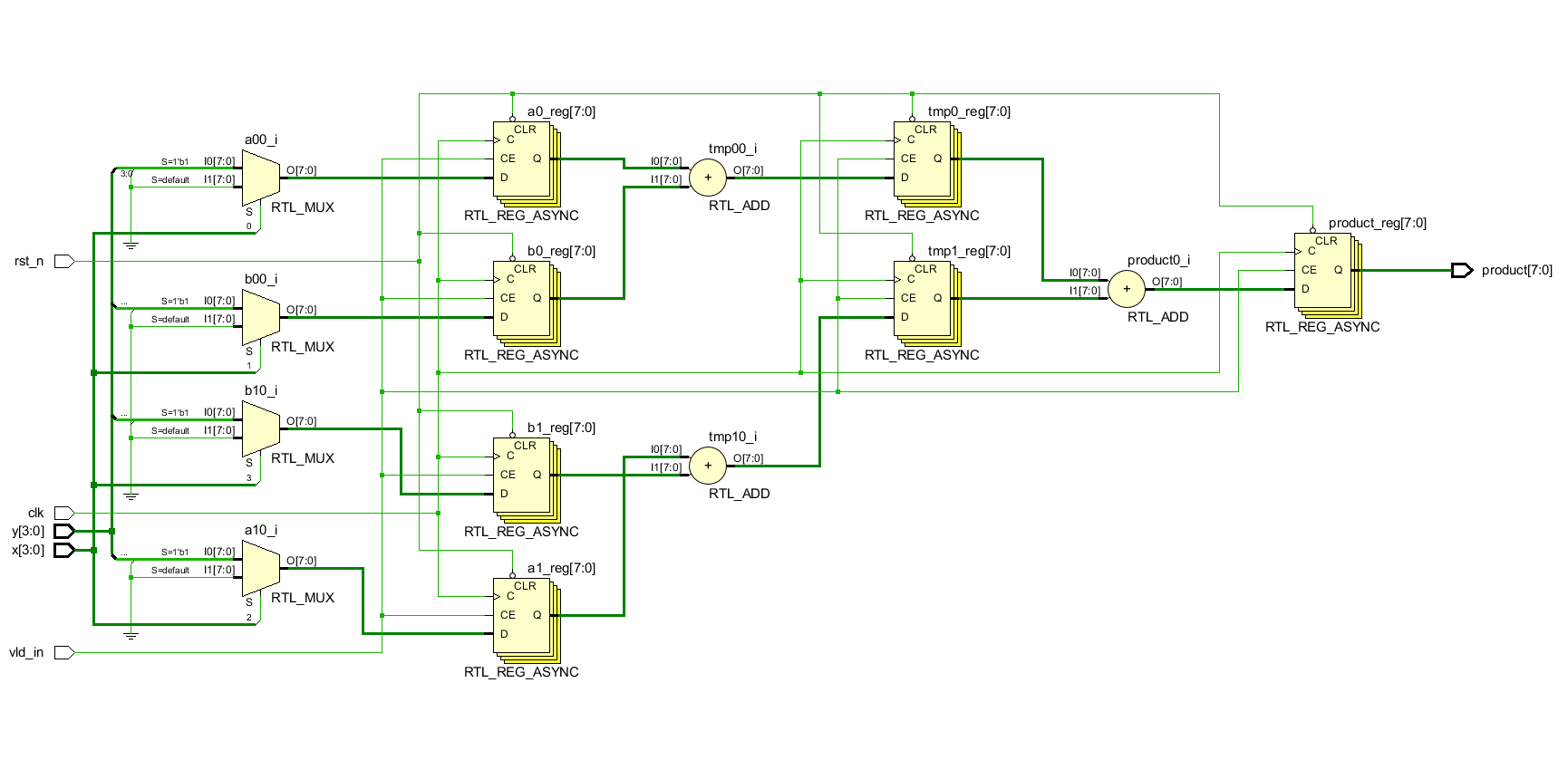
仿真结果如下图所示。



由于在同一个时间周期，流水线加法器无法完成所有的加法，只能通过寄存器将得到的结果暂时储存起来，所以乘法器得到的结果有2个周期的延时。

**（3）电路分析和能耗分析**

利用verilog的RTL分析功能分析代码所生成的电路，结果如下图所示。

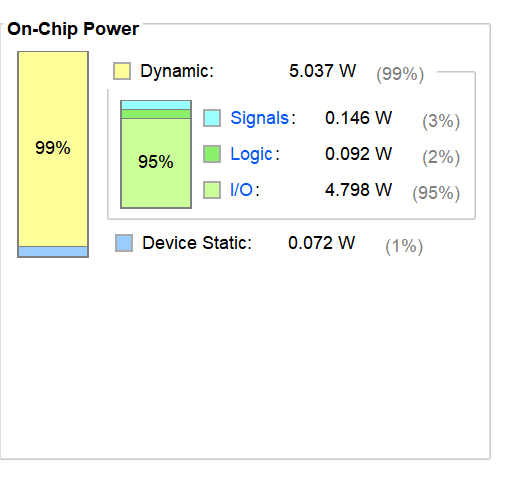


RTL分析生成电路

由于各个数移位和加法的过程是同时进行的，所以电路的总体结构是并行的，整个过程只需要用到3个加法器，电路的效率也得到了一定的提高。

同时，代码是根据时序逻辑编写的，在生成的电路中存在多种D触发器，这些触发器的使用对数值的暂时保存起到了作用，使得两个周期后乘积结果能够正常输出。

利用vivado的功耗分析功能，对代码进行综合、实现后，得到的功耗分析如下图。



功耗分析

可以看到，流水线乘法器的并行电路结构和更少的加法器使得它所占用的逻辑资源更少了，logical资源消耗和siginal消耗都明显减少。