异步流水线的设计与工具实现

张里蒙 18023077

1.实验目的

异步电路的各个模块采用握手方式进行通信。异步控制通路生成异步电路本地时钟的握手控制电路。握手电路产生模块间锁存器或者寄存器的使能信号，也称为本地时钟。利用petrigy工具综合异步STG图，根据逻辑结构编写verilog代码。

2.实验内容

2.1四段握手协议

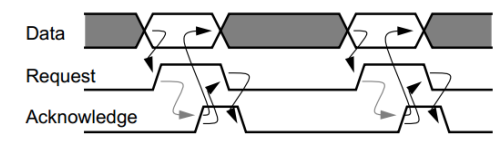
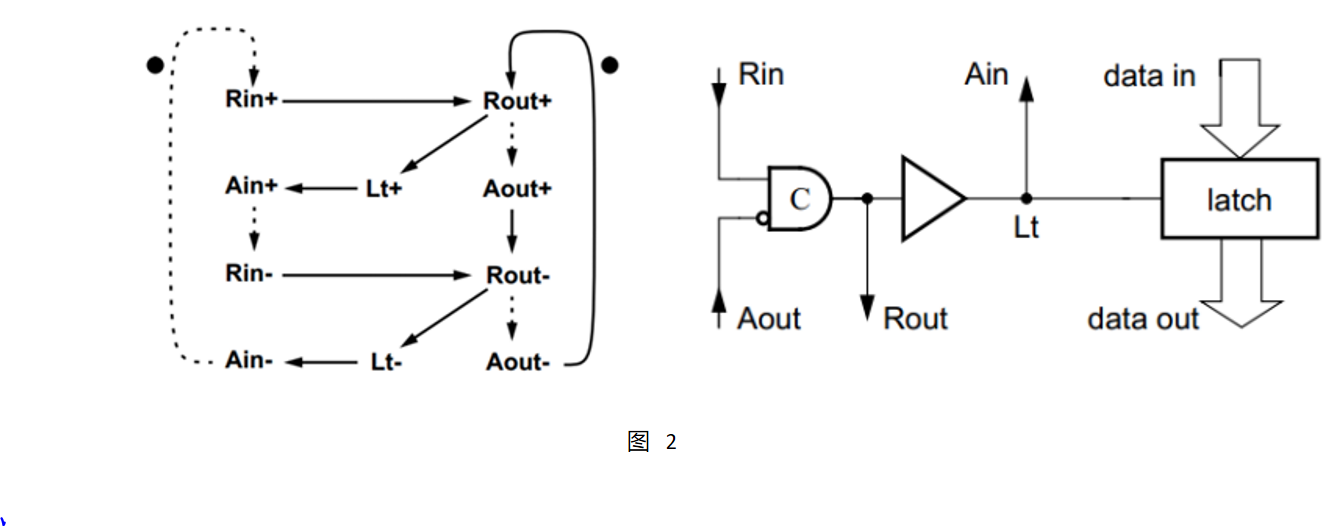
Sutherland提出的为为流水线采用两端握手控制电路，信号的上升沿和下降沿表示同样的含义，因而信号的电平没有特定的含义。但是数据通路内部很多电路需要电平敏感的控制，通常使用两端握手协议到四段握手协议的转换来实现。而直接将微流水线的握手协议实现为四段握手协议可以减少协议转换电路的耗费。

图1

2.2简单四段握手协议

图2中给出了最基本的四段握手协议的STG和其电路实现。虚箭头表示必须由环境保证的时间发生顺序，实线箭头表示必须由电路保证的事件发生顺序。“+”表示信号值从0到1的变化顺序，“-”表示信号从1到0的变化顺序。Rout+表示输出数据已经就绪，该事件必须发生在时间Rin+之后；当输入数据有效（Rin+表示），锁存器关闭（Lt+），对输入端做出应答之后（Aout+），锁存器可以再次打开（Lt-）；锁存器必须交替的打开和关闭。

假设Lt为高时，锁存器锁存数据，Lt为低时锁存器为透明。通过分析图1简单四段握手协议的STG可以看到，当本机Lt要升高时，除了本级的Rin升高以外，还要本级的Aout是由下一级的Ain输入，后一级的Ain降低，必须是在下一级的Lt降低以后，即必须要下一级的锁存器才能锁存数据，这就导致了这种传统的四段数据打包流水线出现一段空一段满的现象，性能较差。

3.实验步骤

首先对STG图进行程序描述，如图3所示：

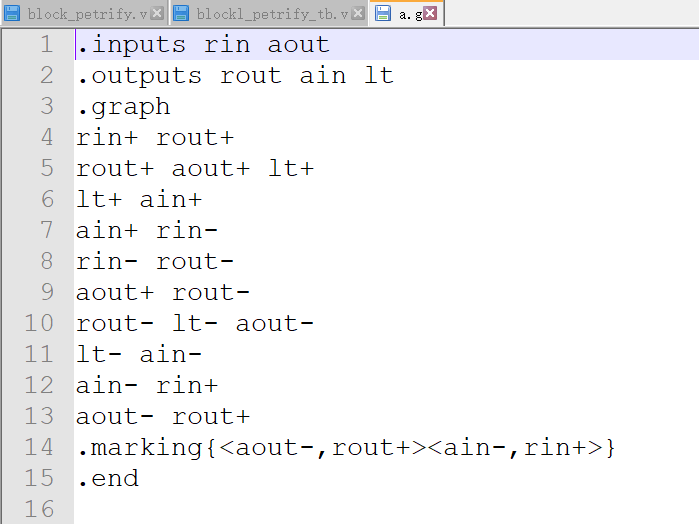
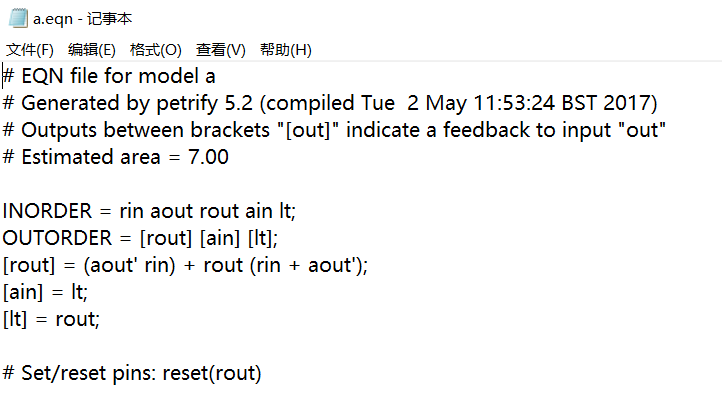


图3

3.1写好的程序命名为a.g，并使用petrify工具运行，运行命令如下：

petrify a.g -eqn a.eqn -cg -no

生成a.eqn文件和petrify.log,前者包含对电路图的描述，后者是对电路细节的描述。

3.2根据a.eqn文件进行verilog代码的编写并进行仿真，结果如下:

