

# 异步流水线的设计与 petrify 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 petrify 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。

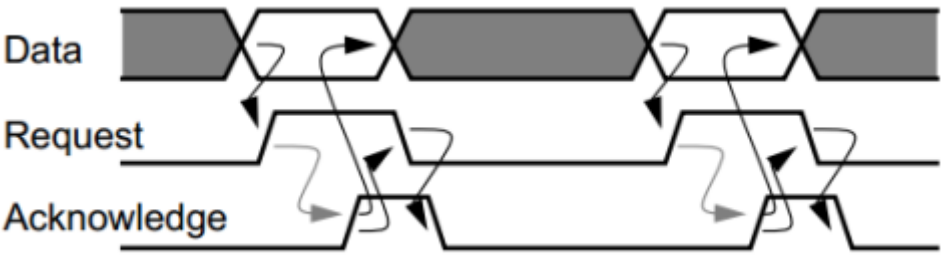


图 1

### 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+

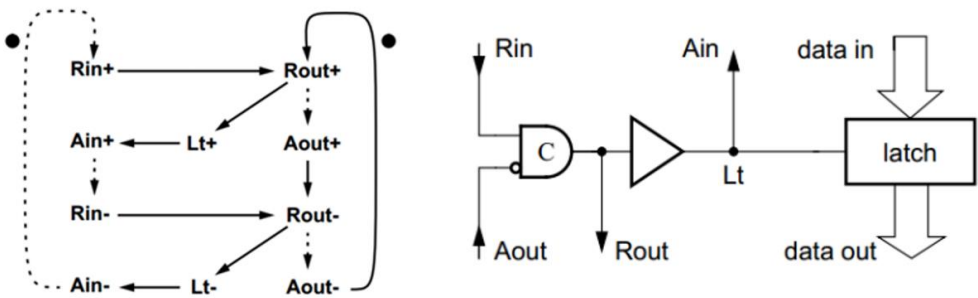


图 2

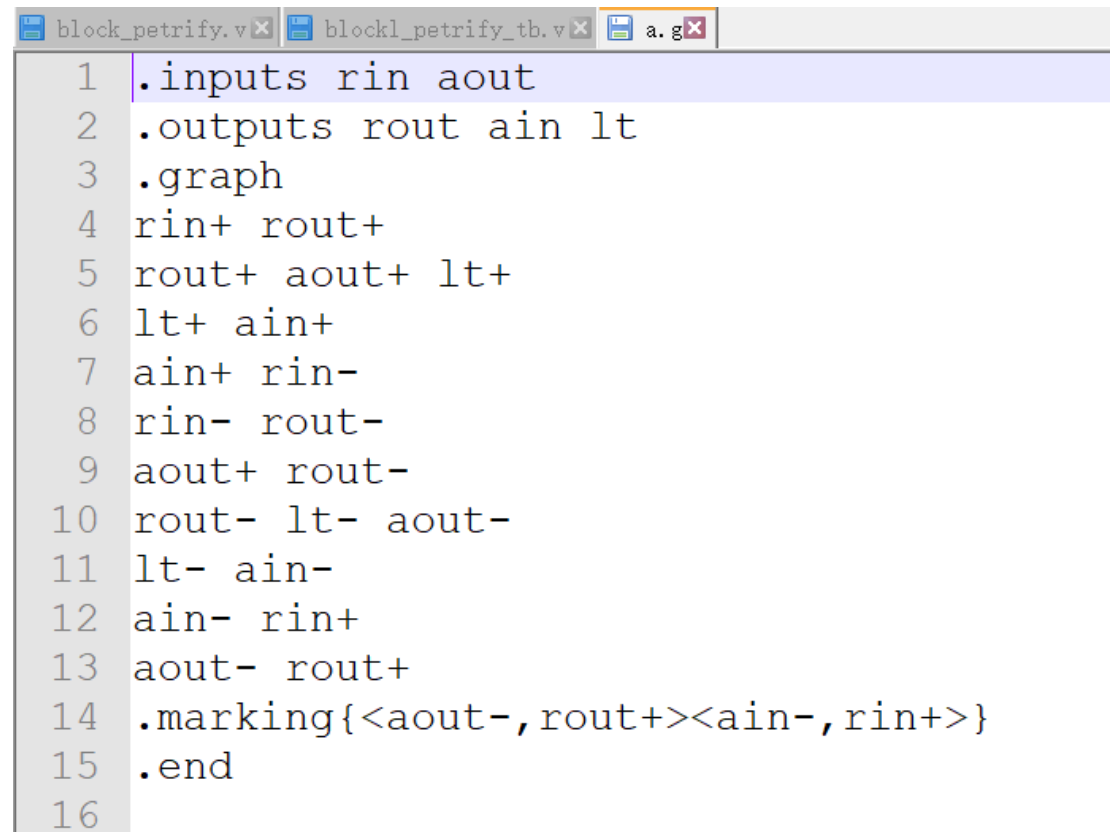
表示)，锁存器关闭（Lt+），对输入端做出应答之后（Aout+），锁存器可以再次打开（Lt-）；锁存器必须交替的打开和关闭。

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：



```
1 .inputs rin aout
2 .outputs rout ain lt
3 .graph
4 rin+ rout+
5 rout+ aout+ lt+
6 lt+ ain+
7 ain+ rin-
8 rin- rout-
9 aout+ rout-
10 rout- lt- aout-
11 lt- ain-
12 ain- rin+
13 aout- rout+
14 .marking{<aout-,rout+><ain-,rin+>}
15 .end
16
```

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

a.eqn - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
# EQN file for model a
# Generated by petrify 5.2 (compiled Tue 2 May 11:53:24 BST 2017)
# Outputs between brackets "[out]" indicate a feedback to input "out"
# Estimated area = 7.00
```

```
INORDER = rin aout rout ain lt;
OUTORDER = [rout] [ain] [lt];
[rout] = (aout' rin) + rout (rin + aout');
[ain] = lt;
[lt] = rout;
```

```
# Set/reset pins: reset(rout)
```

petrify.log - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
-----
| Input -> Input Delays: |
-----
```

```
Average delay = 2.00 events
Worst-case delay = 3.00 events
Input events with worst-case delay: rin+ rin-
> Input events preceding rin+: aout+(3) rin-(3)
> Input events preceding aout+: rin+(1) aout-(1)
> Input events preceding rin-: rin+(3) aout-(3)
> Input events preceding aout-: aout+(1) rin-(1)
```

```
=====
# Gates for signal rout #
=====
```

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，与原来的效果相同。

