**数字逻辑与处理器基础实验加分申请报告**

**——32位MIPS流水线优化**

吴昆 无58 2015010625

黄鹏志\*

李登昊\*

2017年7月

\*The information of other two authors are removed for the github copy.

O gentlemen, the time of life is short;

To spend that shortness basely were too long.

— “Hotspur” in Henry IV, by Shakespeare

# 一、实验目的

1. 学习分析Vivado时序报告，优化电路设计；

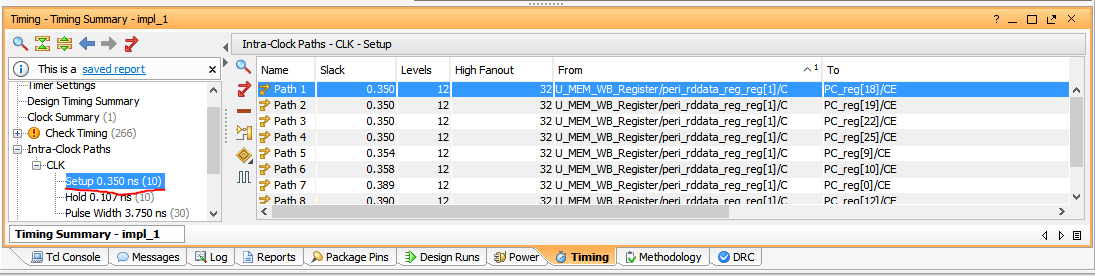
2. 掌握优化关键路径的方法和策略，尤其是针对FPGA上的自动布线电路。

# 二、优化方法

降低Worst Negative Slack，关键在于降低关键路径的时延。

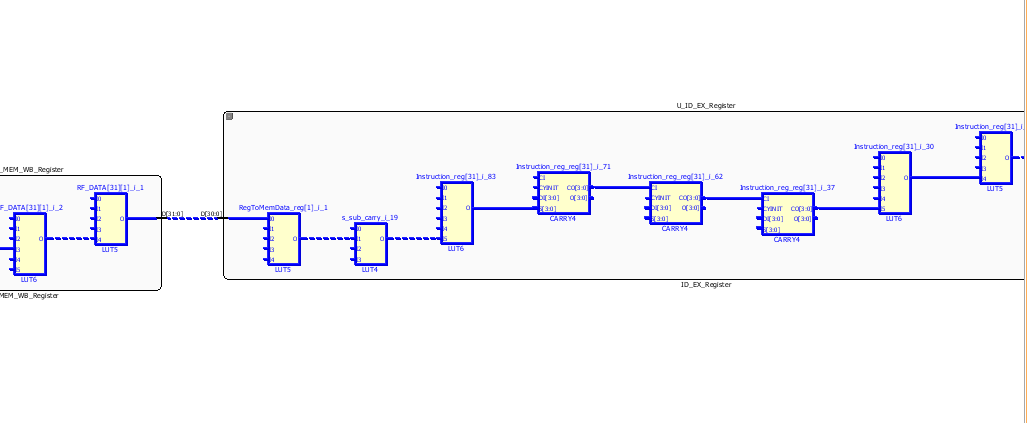
2.1. 找到关键路径

使用Vivado Implementation->Report Timing Summary:

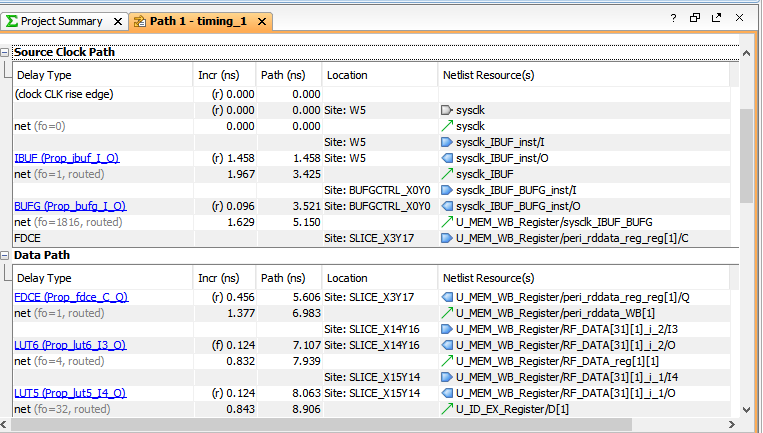


在汇报的界面，找到Intra-Clock Paths->Setup，这里显示了十条时延最长的路径，在未优化前，没有达到WNS大于10ns，因此会出现红色的failed path，说明这些路径时延不满足要求，从始至终失败和优化的都是Setup Time，而Hold Time始终满足约束。

通过两种办法观察关键路径，一种是右键点击一条路径选择Schematic：



从Schematic中可以看到路径经过的各个元件。由于比较长，图只显示了一部分。Vivado综合时开启了flatten hierarchy，所以元件的名字有一些面目全非，但是不看最后的i和数字等乱七八糟的东西，还是可以猜出大概。

第二种就是右键选择view path report来看路径各个地方消耗的时间：

表格列出了路径从头到尾经过的各个元件，每个两行，第一行给列出了这个原件内部的logic delay，第二行列出了从这个元件到下一个元件的net delay。

net delay由线的长度和扇出决定，扇出越大，线越长，net delay越大。

## 2.2 优化关键路径的办法

在我们发现最长路径以后，我们可以找到一些办法来优化它：

1. 简化逻辑，如果是降低元件内部的逻辑那么会降低logic delay，如果降低了逻辑元件级联的数量就会降低net delay，3.1 3.2使用了就是做到了这一点。

3. 减小扇出，3.3就是使用这种方法。

# 三、具体策略

## 3.1 ALU组件去除级联依赖（吴昆）

在Vivado中观察时序报告，我们发现最长路径为转发->ALU输入->减法->比较器->PC。经过考虑，我们认为这是一条发生转发的b指令。

最长路径中出现了ALU输入->减法->比较器，由于我们对于相等的实现是减法结果Z输入到比较器中，导致了最长路径。因此我们对比较器单独设计逻辑，不依赖加法、减法器的输出。

## 3.2 分支指令独立比较器（吴昆）

经过上述调整后，关键路径明显缩短，但仍然经过了减法，我们分析后觉得这是相等判断==综合器自动使用作差实现。由于输入ALU要经过控制信号的重重判断，我们认为为B指令布置独立的比较器可以改善性能，这样做了以后发现果真如此。

通过亮相调整，我们的电路可以在FPGA上以100MHz的速度运行。

## 3.3 控制信号扇出的改进（黄鹏志）

继续优化，仔细观察关键路径的path details，我们发现一些控制信号的扇出十分大（fo=169， net delay= 1~2ns），严重拖慢了频率。因此我们针对一些控制信号进行了优化：

#### Flush信号

在IF\_ID\_Register和ID\_EX\_Register的Flush信号使用频繁且控制了很多寄存器的值，扇出巨大。由于Flush指令的效应是取消对应指令的影响，因此指令只要不影响：（1）PC选择；（2）寄存器堆的数据；（3）内存的数据，就可以了。

因此，对于IF\_ID\_Register，只需要在Flush==1时额外执行

1. Instruction\_reg = 0;
2. Flushed\_reg = 1;

对于ID\_EX\_Register，只需要额外执行

1. PCSrc\_reg = 0;
2. RegWr\_reg = 0;
3. MemWr\_reg = 0;
4. MemRd\_reg = 0;
5. Flushed\_reg = 1;

即可，其它和Flush==0时一致。

这样就大大减少了Flush的扇出，减小了时钟周期。

#### FORWARD信号

在EX\_MEM\_Register中的FORWARD信号与Flush相同，减少了部分不必要的扇出。

# 四、优化结果

1. 在优化前，Vivado2014汇报Worst Negative Slack为12.3ns左右，折合频率为80MHz左右。

2. 通过3.1 和3.2节的优化，Worst Negative Slack下降到9.9ns，低于10ns，并且FPGA可以在100MHz的最大频率运行最大公约数的计算，并正确通过UART接受参数和返回结果。

3.通过3.3节的优化，Worst Negative Slack下降到9.55ns，折合频率105MHz。