

数字逻辑与处理器基础 第五次作业

单周期 MIPS 处理器

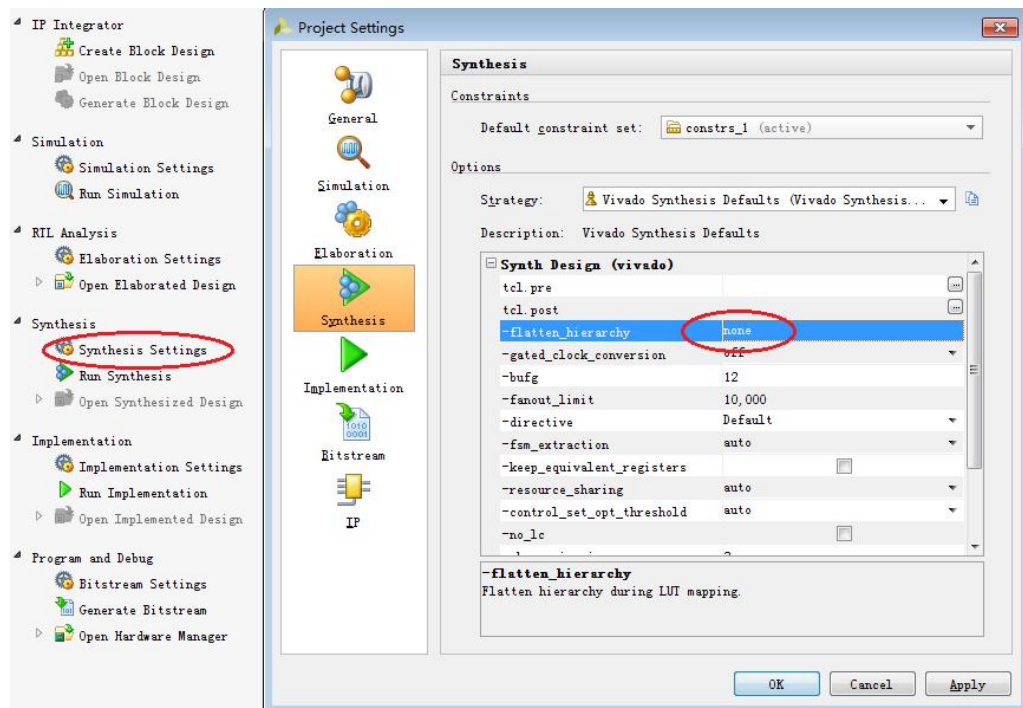
实验内容：

- (1) 编写 Verilog 代码，完成《数字逻辑与处理器基础》理论课程布置的单周期 CPU 大作业，并综合实现并通过后仿真。
- (2) 仿真执行代码可二选一：
 - a) 运行大作业中第 3 节给出的汇编程序：
MIPS Assembly

```
0          addi $a0, $zero, 3
1          jal sum
2      Loop: beq $zero, $zero, Loop
3      sum:  addi $sp, $sp, -8
4          sw $ra, 4($sp)
5          sw $a0, 0($sp)
6          slti $t0, $a0, 1
7          beq $t0, $zero, L1
8          xor $v0, $zero, $zero
9          addi $sp, $sp, 8
10         jr $ra
11      L1:  addi $a0, $a0, -1
12         jal sum
13         lw $a0, 0($sp)
14         lw $ra, 4($sp)
15         addi $sp, $sp, 8
16         add $v0, $a0, $v0
17         jr $ra
```
 - b) 运行你在理论课“汇编大作业”中实验内容 2 中完成的任何一种排序算法，对 128 个可预先设定的随机数进行排序。
- (3) 观察 PC 及其他关键寄存器的变化以及程序执行结果，验证处理器正确性；
- (4) 实验报告中要分析单周期处理器的时序性能（最高工作频率，限制最高工作频率的关键路径是什么，路径上有哪些单元，哪些单元是最耗时的），FPGA 资源消耗情况等。

实验提示:

- (1) 时序性能的分析对象是 MIPS 处理器。
- (2) 单周期 MIPS 处理器的指令存储器和数据存储器的读取是组合逻辑，和作业四 MEM_UART 中描述的存储器不一致，请留意之间的差别。
- (3) 为了避免因缺少输出，工具将大量电路优化掉，可以利用 SW1 和 SW0 选择 \$a0, \$v0, \$sp 和 \$ra 的低 8 位引出到 CPU 顶层端口，并接在 LED7~LED0 上作为输出。或者利用 MEM_UART 将数据 RAM 通过串口输出。
- (4) 在 FPGA 中，寄存器（触发器和 RAM）的值可以在 Verilog 代码进行设置，具体可以参考网络学堂上上传的 UG901（综合和可综合的代码）中，第四章中“Initializing RAM Contents”中相关内容。务必注意在一般的 ASIC 设计中，不能使用这样的方法。
- (5) 为了优化性能，Vivado 工具可以将层次化的设计打平，然后突破 HDL 的描述层级进行优化。跨层的信号，以及对信号的重命名会给调试带来不便，因此，可以将这个优化选项关掉。如下图所示。



小学期实验预告:

内容二选一:

- (1) 实现流水线的 MIPS 处理器，执行排序算法，并和单周期处理器的性能进行比较；
- (2) 用数字逻辑电路实现排序，并和用处理器实现的排序算法进行比较。