

HLS 软件的使用

一、 基本使用步骤

1. **新建工程**: 选择路径和工程名, 需要有 2 层文件夹; 选择板子型号;
2. **编写代码**: 需要有顶层模块和测试模块 (main 函数);
3. **C 仿真**: 测试功能;
4. **C 综合**: c++代码转 verilog 代码, 需要进行时序、资源优化;
5. **C/RTL 联合测试**: C 测试应该可以过, 主要看 RTL 测试能否通过, 以及所耗时间, 代码不合理可能 RTL 测试需要花费很长时间!!!;
6. **导出 IP**: 生成 IP 核, 可以在 vivado 中使用。

二、 时序优化

1. **流水线**: 流水线是减小耗时最有效的方法, 主要用于循环里 (最内层的循环, 用于非最内层循环会将内层循环展开), 流水线的 II(Initiation Interval)可以先设为 1, 根据综合的结果再调整;
2. **数据流(Dataflow)**: 主要用于函数之间, 如果有使用 stream 类型的变量, 一定要使用数据流, 否则会报错;
3. **循环展开**: 用资源换时间, 用于小循环;
4. **循环扁平化 (默认执行)**: 将嵌套循环合并成一个循环, 可以减去循环的初始化时间;
5. **数组拆分**: 减少数组读写时间, 用于小数组;
6. **内联函数 (默认执行)**: 减去模块调用的初始化时间, 用于小函数;
7. **少用浮点数**: 浮点数的乘除、加减运算都比较费时, 在不影响精度的情况下可以用整型数或定点小数代替, 整型数和定点小数的运算耗时是一样的;
8. **使用缓存**: RAM 最多有 2 个口进行读(写), 如果需要同步读取很多数据, 可以使用缓存。

三、 资源优化

1. **小数组拆分**: 用寄存器代替 RAM, 减少 RAM 使用;
2. **数组合并**: 2 个或多个相关的数组合并成 1 个, 可以减小 RAM 数量。
例如 1 个 4k 的 sin 数组和 1 个 4k 的 cos 数组, 单独实现时需要 2 个 BRAM_18k 模块, 利用率只有 4/18, 如果合并成一个 2 维或 1 维数组, 则尺寸变为 8k, 实现时需要 1 个 BRAM_18k 模块, 利用率为 8/18;
3. **抑制循环展开**: 如果循环内有复杂运算, 可以抑制循环展开以节约资源;
4. **使用引用**: 对于结构体和类使用引用, 不额外消耗资源;
5. **数据位宽优化**: 合理减小数据位宽, 可以减少资源消耗。

四、 端口设置

1. 输入输出参数设置（数据接口）

- a. 方向设置: C++代码输出参数用引用或数组实现, 输入参数和普通参数一样, 不需要使用指令;
- b. 类型设置:
可选类型:
 - 普通信号**: 用于参数配置(在硬件中配置);
 - axis (AXI 流)**: 用于图像或视频信号;
 - m_axi (AXI 主机)**: 用于内存(BRAM 等)读写;
 - s_axilite (AXI 从机)**: 用于参数配置(在软核中配置)。

2. 返回值设置（控制接口）

可选类型:

ap_ctrl_hs: 默认类型

ap_start: 用于控制模块执行, 为 1 时模块开始运行, 必须在 **ap_ready** 置为 1 之前保持高电平;

ap_idle: 用于指示模块正在运行或空闲;

ap_ready: 用于指示输入已全部读完, 模块已经准备好接收新数据, 若此时 **ap_start** 为高, 则会继续读入新数据, 若设计不是流水线任务, 则此时要把 **ap_start** 拉低, 待下一次任务时拉高 **ap_start** 启动新任务并读入新数据;

ap_done: 用于指示模块运行结束;

ap_ctrl_chain: 和 **ap_ctrl_hs** 类似, 多了一根信号;

ap_ctrl_none: 无控制信号, 只有特定情况可以使用;

s_axilite: 使用软核控制, 其实是在软核中控制 **ap_ctrl_hs** 包含的信号的值。

常用 **ap_ctrl_none** 和 **s_axilite** 两种类型。

五、 常见问题

1. 设置的时钟不满足要求

解决方法:

- a. 降低时序优化程度, 例如去掉流水线优化;
- b. 降低设置的时钟频率。

2. C 仿真报错, 不给出原因, 但 C 综合不报错

可能原因:

- a. 数组越界: 检测代码, 找到可能发生数组越界的地方, 改正代码;
- b. 数组太大(超过 100000): 目前只能通过减小数组尺寸解决!!!。

3. C/RTL 联合测试时报出大量警告, 导致测试很慢

可能原因: 使用了浮点运算, 综合成 verilog 代码时使用了 DSP48 模块;

解决方法: 可以使用定点小数代替浮点数, 需要注意定点小数的位宽分配,

以减小误差。

4. 类或结构体类型的参数传递需要使用引用，否则会进行拷贝，额外消耗时间和资源。

六、 数学运算

1. 加法

数据类型	M18K	DSP	FF	LUT	Latency	备注	优先使用
int	0	0	0	39	0		√
ap_fixed	0	0	0	39	0		√
float	0	2	231	245	3		

注：ap_fixed 表示 ap_fixed<32,16>

2. 乘法

数据类型	M18K	DSP	FF	LUT	Latency	备注	优先使用
int	0	3	0	21	0		√
ap_fixed	0	3	0	21	0		√
float	0	3	130	152	1		

3. 除法

数据类型	M18K	DSP	FF	LUT	Latency	备注	优先使用
int	0	0	430	403	35		
ap_fixed	0	0	639	583	51	使用了 IP 核	
float	0	0	318	832	6		√

4. sin、cos

数据类型	M18K	DSP	FF	LUT	Latency	备注	优先使用
ap_fixed	4	23	383	1428	42	需要使用 ROM 查表	
float	0	37	1527	4502	21~22		√

5. log

数据类型	M18K	DSP	FF	LUT	Latency	备注	优先使用
ap_fixed	1	6	380	1332	3	需要使用 ROM 查表	√
float	0	13	373	610	5		

6. log2

a. float 类型

方法	M18K	DSP	FF	LUT	Latency	备注	优先使用
log2(x)	8	23	1314	3286	9	需要使用 ROM 查表	
log(x)/log(2)	0	13	723	1422	12		√

b. ap_fixed<32,16>类型

方法	M18K	DSP	FF	LUT	Latency	备注	优先使用
log(x)/log(2)	0	10	414	1535	4		√

7. pow

数据类型	M18K	DSP	FF	LUT	Latency	备注	优先使用
ap_fixed	5	25	1335	1121	10~46	需要使用 ROM 查表	
float	1	10	1275	3780	11		√

8. exp

数据类型	M18K	DSP	FF	LUT	Latency	备注	优先使用
ap_fixed	5	12	597	667	4	需要使用 ROM 查表	
float	0	7	700	2408	10		√

9. sqrt

数据类型	M18K	DSP	FF	LUT	Latency	备注	优先使用
ap_fixed	0	0	1086	4763	9		
float	0	0	734	2038	12		√

七、 取整和取余运算

1. round 四舍五入

a. x 为 float 类型

方法	M18K	DSP	FF	LUT	Latency	备注	优先使用
round(x)	0	0	116	672	2		√
x+0.5	0	2	264	703	4	需要使用浮点加法运算	

b. x 为 ap_fixed<32,16>类型

方法	M18K	DSP	FF	LUT	Latency	备注	优先使用
round(float(x))	0	0	401	1295	5	需要使用定点转浮点 IP 核	
x+(ap...)0.5	0	0	0	111	0		√

2. floor 向负无穷取整

a. x 为 float 类型

方法	M18K	DSP	FF	LUT	Latency	备注	优先使用
floor(x)	0	0	121	762	2		
(int)x	0	0	0	452	0		√

b. x 为 ap_fixed<32,16>类型

方法	M18K	DSP	FF	LUT	Latency	备注	优先使用
floor(x)	0	0	0	113	0		
(int)x	0	0	0	68	0		√

3. ceil 向正无穷取整

a. x 为 float 类型

方法	M18K	DSP	FF	LUT	Latency	备注	优先使用
ceil(x)	0	0	121	762	2		√

b. x 为 ap_fixed<32,16>类型

方法	M18K	DSP	FF	LUT	Latency	备注	优先使用
ceil(x)	0	0	0	152	0		√

4. 取余

a. x 为 float 类型

方法	M18K	DSP	FF	LUT	Latency	备注	优先使用
fmod(x,y)	0	4	250	1551	2~19	结果范围 (abs(y),abs(y))	
x=(x>=y)? (x-y):y	0	2	297	381	3	需要使用浮点加法运算	√
x=(x>=y)? (x-y):((x<0)? (x+y):y)	0	2	366	511	4		√

b. x 为 int 类型

方法	M18K	DSP	FF	LUT	Latency	备注	优先使用
x%y	0	0	430	403	35	结果范围 (abs(y),abs(y)) 需要使用整数除法	
x=(x>=y)? (x-y):y	0	0	0	89	0		√