

一、题目要求

(一)赛题一:采用硬件 RTL 代码方式实现串行 FFT 算法(要求工作频率达到 1Ghz)

1. 描述及要求:

- (1) FFT 支持串行 64、128、256、512 点运算, 支持 FFT/IFFT 两种运算模式。
- (2) 运算支持自适应压缩移位, 以减少设计电路面积。
- (3) 对采用的基运算组合方式(基 2、4、8)不做强制要求, 最大支持基 8 运算。
- (4) 为减少面积存储单元采用两块单口 ram(同一时刻只读或只写), 每块大小 256*32(数据位宽 32bits, ram 深度为 256(最大支持 512 点))。
- (5) FFT 输入 I/Q 数据为 12bits 有符号数, 输出 I/Q 数据 30bits, 30bits 为理论计算最大值(512 点基 2 串行实现)。

2. 各种运算单元带来的位宽扩展情况:

- (1) Radix-2 2 点 FFT 最大会带来 2bit 的数据位扩展
- (2) Radix-4 4 点 FFT 最大会带来 3bit 的数据位扩展
- (3) Radix-8 8 点 FFT 最大会带来 4bit 的数据位扩展

注: 自适应移位、ram、复乘器、旋转因子表等均有相应的 IP 可供选择, 选手也可自行设计。

3. 评审得分点:

- (1) 500 条用例(预设), 用例跑通越多, 得分越高。
- (2) FFT 运算处理时间越短, 得分越高。
- (3) 7nm 工艺, 常压下, 综合面积越小, 得分越高。
- (4) 7nm 工艺, 常压下, 综合时钟频率越高, 得分越高。
- (5) 时钟门控率越高, 得分越高。

4. 输出要求:

- (1) 算法设计文档和算法代码。
- (2) 详细设计文档和逻辑代码、软件代码。

(二) 赛题二:采用软硬件结合的方式实现 SM9 算法(要求具备防 DFA\SPA\DPA 能力)

1. 描述及要求:

- (1) SM9 需要的模乘、模加等基本运算使用硬件实现。其他高层算法可采用软件实现。具备防止各种已知 SPA\DPA\DFA(二阶或者高阶)攻击的能力。
- (2) 可只实现点乘运算, 其它高层算法不强制要求。
- (3) 256 点乘至少可以达到 20 次/S(对应时钟频率为 120MHz, 其他时钟频率其它密钥位宽可等比例折算)。
- (4) 是否素域等不做强制要求。

2. 评审得分点:

- (1) 防攻击能力越强越全面, 没有任何防护漏洞。得分越高。

特别声明: 本赛题知识产权完全归属答题人所有。

(2) SM9 密钥位宽至少为 256 位，位数越长，得分越高。

(3) 性能越高，得分越高。

3. 输出要求:

(1) 算法设计文档和算法代码。

(2) 详细设计文档和逻辑代码、软件代码。

(三) 赛题三：逻辑实现 ZUC 算法(要求具备防 DFA\SPA\DPA 能力)

1. 描述及要求:

(1) 逻辑实现一个完整的 ZUC 算法设计。具备防止各种已知 SPA\DPA（二阶或者高阶）攻击的能力。防护手段不限。

(2) 理论分析 SPA\DPA（二阶或者高阶）攻击对 ZUC 的理论破解时间。

(3) 时钟频率不限，资源不限，功耗不限。采用 VHDL\VERILOG 实现。

2. 评审得分点:

(1) 具备防止各种已知 SPA\DPA（二阶或者高阶）攻击的能力，无安全漏洞。

(2) SPA\DPA 防攻击理论清晰，理论破解时间越长得分越高。

3. 输出要求:

(1) 算法设计文档和算法代码。

(2) 详细设计文档和逻辑代码。

(四) 赛题四：基于 sigma_delta 的高性能 Audio Codec 设计

1. 描述及要求:

(1) 实现完整的 Digital+Analog 系统建模；

(2) Verilog 实现 Digital 逻辑，完成仿真验证及资源开销；Analog 部分完成建模仿真；

(3) 给出完整的性能报告，包含 SNR、THD+N、频响、带内平坦度等指标。

2. 评审得分点:

(1) 关键性能指标 $SNR > 100dB$, $THD+N < -80dB$ ；性能越好得分越高；

(2) Digital 逻辑面积越小得分越高。

3. 输出要求:

(1) 算法设计文档和算法代码。

(2) 详细设计文档和逻辑代码。

(3) 仿真报告。

(五) 赛题五：低功耗的语音识别系统设计

1. 描述及要求:

(1) 能通过语音端点检测 (VAD:Voice Activity Detection) 实现语音唤醒，满足在有语音环境下才进行后续的语音识别，从而达到低功耗的要求。

(2) 能对简单的词语和短命令（“播放”、“暂停”等）进行高精度的识别。

(3) Verilog 实现 Digital 逻辑。

2. 评审得分点:

(1) 高噪声条件下 ($SNR < 5DB$) 语音唤醒的虚警率（误唤醒）和漏检率均不高于 30%；低噪声条件下虚警率（误唤醒）和漏检率均不高于 15%。

(2) 语音识别模块的识别率越高越好。

3. 输出要求:

特别声明：本赛题知识产权完全归属答题人所有。

- (1) 详细设计文档及算法设计文档。
- (2) 性能测试或仿真报告。

(六) 赛题六：麦克风阵列算法建模及方案实现

1. 描述及要求：

(1) 麦克风阵列是利用一定数目，一定空间构型的声学传感器（一般是麦克风）组成，用来对声场的空间特性进行采样并处理的系统。

(2) 麦克风阵列近场（2~3 个 mic）或远场算法（大于 3 个 mic）建模（波束形成/声源定位/去混响技术...），任选其一。

(3) 方案实现：资源不限，平台不限。选择一个语音应用场景，完成对该算法硬件平台方案实现。

2. 评审得分点：

(1) 对现有主流算法对比分析，可从抗干扰性、识别率、运算速度、算法代价等方面进行分析。

(2) 选择一种算法进行代码实现，仿真结果分析。

(3) 硬件平台实现，对比算法仿真结果。能够对 mic 器件选择、mic 数量及摆放位置进行理论分析。

(4) 能够对算法优化改进分析。

3. 输出要求：

(1) 算法分析设计文档和算法代码(matlab/c/c++)。

(七) 赛题七：实现一个 Sparse Matrix-Multiply-Vector

Accelerator

1. 命题描述：

(1) 实现一个 Sparse Matrix-Multiply-Vector (SpMV) Accelerator，提供 RTL code, 加速算法, 并演示计算流程。我们提供下列矩阵集合(包含 MATLAB mat-file 格式, Matrix Market 格式, 和 Rutherford/Boeing 格式, 做题时选其中一种格式即可)：

<http://www.cise.ufl.edu/research/sparse/matrices/HB/beause.html>

<http://www.cise.ufl.edu/research/sparse/matrices/Bai/rbsa480.html>

<http://www.cise.ufl.edu/research/sparse/matrices/Bai/qc2534.html>

<http://www.cise.ufl.edu/research/sparse/matrices/DRIVCAV/cavity07.html>

<http://www.cise.ufl.edu/research/sparse/matrices/Fluorem/GT01R.html>

<http://www.cise.ufl.edu/research/sparse/matrices/HB/arc130.html>

http://www.cise.ufl.edu/research/sparse/matrices/HB/bp_1600.html

<http://www.cise.ufl.edu/research/sparse/matrices/HB/mbeause.html>

<http://www.cise.ufl.edu/research/sparse/matrices/Hollinger/g7jac010.html>

http://www.cise.ufl.edu/research/sparse/matrices/JGD_Homology/ch6-6-b3.html

http://www.cise.ufl.edu/research/sparse/matrices/JGD_Homology/n2c6-b4.html

特别声明：本赛题知识产权完全归属答题人所有。

http://www.cise.ufl.edu/research/sparse/matrices/JGD_GL7d/GL7d11.html

(2) 请自行下载矩阵，并将其转换为 32-bit 浮点。以 Dense 或者 CSR/COO/Hyb 等常见稀疏矩阵存储格式（或者自定义的某种稀疏存储格式），将上面所列 12 个矩阵分次存入内存空间。利用所提供的脚本生成与这些矩阵 ($M_i \times N_i$) 尺寸相配的随机向量 ($N \times 1$) 并存入内存。向量和矩阵的存入不计入运算时间。

(3) 所设计的加速器，需要从内存中读取矩阵和向量，并传入加速器内部实现矩阵和向量相乘，并最后将结果存入内存。

2. 实现要求：

(1) 所有矩阵，向量元素均为 single-precision floating point (32 bits) 长度

(2) 加速器的硬件逻辑中最多存在 256 个 fp32 浮点乘法器

(3) 12 个矩阵的格式预处理可由软件处理；但对随机产生的 12 个向量的预处理必须由加速器的硬件逻辑完成。

(4) 允许将附加和预处理后得到的信息存入内存

(5) 不允许用有损的方式

(6) 加速器与内存之间的读、写的数据位宽各为 128-bit。为简化非关键特性，内存频率和加速器同频，接口为 Dual Port SRAM，单 cycle 延迟。（见图 1）

(7) 由脚本随机产生的向量也会有一定程度的稀疏率（30%~80%）。参赛者可以结合稀疏向量一同加速。（注意不能对向量预处理。注意处理向量的硬件代价）

(8) 脚本：

向量取决于矩阵的高度，请自行用脚本产生一组随机向量（稀疏率为 30%~80%），或使用提供脚本生成向量。

提供的脚本为 python 3 脚本，参赛者可根据需求修改为 python 2 或其他脚本的语法。

```
import random as rn
import math
import sys
length = sys.argv[1]
lower = int(length)*30/100
upper = int(length)*80/100
ran = rn.randint(lower, upper)
array = []

for x in range(ran):
    array.append(0.0)
for y in range(int(length) - ran):
    array.append(rn.uniform(-10.0, 10.0))
rn.shuffle(array)
for elem in array:
    print(str(elem) + ", ", end='')
print("")
```

3. 评审得分点：

(1) 计算结果要正确，可忽略 32bit 精度结果的误差

(2) 所费的运算时间越少，得分越高

特别声明：本赛题知识产权完全归属答题人所有。

(3) 通常, 如果矩阵不做任何稀疏存储, 仅以 Dense 格式进行运算, 则运算时间肯定会比稀疏化后矩阵的时间长; 如果根据 CSR/COO/HYB (或者自定义的某种稀疏存储格式) 稀疏存储, 并采取与此有关的优化手段(跳 0 等), 则运算时间可以大为降低;

(4) 逻辑规模越小, 得分越高

(5) 加速器对内存访问带宽越小, 得分越高

(6) 功耗越小, 得分越高

4. 输出要求:

(1) 详细的算法解释文档, 和算法代码。请在文档中写明加速亮点。

(2) 详细的设计文档, 和 RTL 代码(Verilog)。请在文档中写明低功耗设计点。

(3) 硬件对上述 12 个 benchmarks 的加速性能分析文档。

(八) 赛题八: 时序加扰防护的理论分析及攻击。

1. 描述及要求:

(1) 选择 AES/DES/SM4 中任意一种对称加解密引擎, 在 FPGA 平台实现对对称加解密引擎的时序加扰防护, 防护方法包括但不限于随机时钟门控、随机时钟抖动及伪操作等;

(2) 从理论上比较各个加扰防护的防护能力, 说明各个防护的优缺点;

(3) 在 FPGA 平台上选择一种或几种时序加扰防护进行实际攻击测试, 攻击可以获得正确的密钥信息。

(4) 实现的时钟频率不限, 但是时序加扰对于性能的影响不能过大。

2. 评审得分点:

(1) 在 FPGA 平台上实现的时序加扰方法越多、防护能力越强得分越高。

(2) 不同时序加扰防护比对理论分析清晰, 结论越合理得分越高。

(3) 时序加扰攻击的攻击能力越强, 攻击效率越高得分越高。

3. 输出要求:

(1) 时序加扰防护的设计文档和实现代码。

(2) 不同时序加扰防护的比较文档。

(3) 时序加扰攻击的算法文档和攻击实现代码。

(九) 赛题九: 实现随机数后处理

1. 描述及要求:

(1) 设计一套随机数后处理的模块, 真随机数/伪随机数通过该模块后, 所有数据均能够通过 sp800 测试套件(测试数据不少于 30 组, 每组推荐值 $10^6 \times 1073$ 比特);

(2) 输入的随机数来源自行确定。

(3) 后处理输出的数据通过 sp800 测试套件的概率分析及实测;

(4) 后处理不能直接采用 SP800-90A 推荐方式, 可在其基础上进行修改。

(5) sp800 优化测试套件可参考: Sýs M, Říha Z. Faster randomness testing with the nist statistical test suite[C]//International Conference on Security, Privacy, and Applied Cryptography Engineering. Springer, Cham, 2014: 272-284.

(6) 可采用软件实现, 但硬件实现方式优先。

2. 评审得分点:

特别声明: 本赛题知识产权完全归属答题人所有。

- (1) 后处理输出的数据通过 sp800 测试套件的概率越大越好;
- (2) 逻辑规模越小, 得分越高。
- (3) 后处理对输入输出数据的压缩比越接近 1, 得分越高。

3. 输出要求:

- (1) 算法设计文档和算法代码。
- (2) 详细设计文档和逻辑代码、或软件代码。

(十) 赛题十: 实现真随机数(模拟 RO 环/数字 Garo 环)的理论熵

源模型

1. 描述及要求:

- (1) 对随机源及随机数的随机性建立模型进行分析。
- (2) 给出单比特熵值可以达到的熵值;
- (3) 给出最佳熵值建议方案

2. 评审得分点:

- (1) 对影响随机性的因素刻画越精确得分越高;
- (2) 单比特熵值越高越好;

3. 输出要求:

- (1) 详细的理论熵源模型文档;

二、作品提交要求:

由于华为赛题的专项奖是线下评审, 没有现场答辩环节, 请参赛者除按竞赛组委会要求提交 PPT 外, 还需按华为赛题要求提供详细文档和代码。如果是硬件作品, 需提供照片或视频(含竞赛组成员合影)。

三、华为赛题专项奖设置:

- (1) 特等奖 2 队, 每队奖金 5 万元, $2 \times 5 = 10$ 万
- (2) 一等奖 4 队, 每队奖金 3 万元, $4 \times 3 = 12$ 万
- (3) 二等奖 8 队, 每队奖金 1 万元, $8 \times 1 = 8$ 万
- (4) 竞赛组织突出贡献奖, 1-2 名

特别声明: 本赛题知识产权完全归属答题人所有。

“兆易创新杯”第十三届中国研究生电子设计竞赛

“Arm 杯”上海赛区组委会

(中共上海大学委员会研究生工作部 代章)

2018 年 4 月 10 日

特别声明: 本赛题知识产权完全归属答题人所有。