第11次课 2024-4-2 周二1-2节 第6周

继续syslink多核加速框架

是否已完成syslink的ex04\_sharedregion例子的编译、运行？ 🡪 进度计划是否符合预期？

改造ex04例程，让修改后的例程能够适配未来的目标！ 🡪 让DSP计算一组浮点sqrt ，实现验证DSP加速！

思考：如何实现DSP核的sqrt加速运算？ 🡪 设计！！

1. 在examples/目录下复制ex04\_sharedregion例程，改名ex04\_dsp\_math
2. 进入ex04\_dsp\_math/，host/和dsp/分别是Linux代码、DSP代码！
3. ARM端，Linux代码：修改

原来例程申请share空间的大小？在哪里？需要修改？（一个float类型占4B）

如何把送待计算的数据送到share空间？

如何发命令给DSP核，通知DSP开始计算？DSP如何知道需要计算sqrt数据的数量？

如何读出并打印DSP核返回的结果？

如何验证DSP的结果是正确的？

如何评估DSP核与ARM核各自的计算时间？

…. (先谋而后动！)

DSP端，BIOS，修改：

如何接受计算sqrt命令，以及数据，计算参数？

DSP核如何完成sqrt？

返回结果？

更多修改：

现在是单次工作模式，实际工作时是连续模式！

1. 从文件中一次读取一组数据，交给DSP做sqrt !，

重复，直到文件结束！ （实用性考虑）

1. 整体代码，包括ARM\DSP端，在结束退出时没有内存等资源的残留！ （稳定性考虑）

--> App\_exec:

App\_exec: Writing string "texas instruments" to shared region 1 buffer

To compare sqrt caculate by ARM & DSP !

Caculate\_time: 3297 ns

App\_exec: Received-> Operation complete

App\_exec: Transformed string: Sqrt(2.000000)=1.414214 Sqrt(23.200001)=4.816638 Sqrt(333334.593750)=577.351379 Sqrt(1873.800049)=43.287411

<-- App\_exec:

Sqrt by ARM: Sqrt(2.000000)=1.414214 Sqrt(23.200001)=4.816638 Sqrt(333334.593750)=577.351379 Sqrt(1873.800049)=43.287411

Caculate\_time: 4088815 ns