

STM32-DSP性能软件仿真测试

一、简介说明

如下展示通过MDK软件仿真进行STM32对应型号，执行DSP函数所需运行时间。

代码: https://github.com/SillyPigeon/stm32_dsp_flow_test_proj

*注意:

1、由于MDK软件仿真功能有限，只支持stm32f1x系列的完整芯片仿真

对于需求的型号，MDK只支持使用下述内核模型仿真 [μVision User's Guide \(arm.com\)](#)

- Arm Cortex-M0 / M0+ / M3 / M4 / M7* / M23* / M33* / M35P* / M55* / M85*
- Arm SecurCore 000 / 300
- ARM7 and ARM9

2、对于stm32的f7与h7模型，由于Arm Cortex-M7* 只支持在AVH(arm虚拟机)中执行，所以对应f7与h7型号是通过MDK中contex-M4的模型，执行对应指令模拟运行

3、对于stm32h745 双核芯片，MDK无法软件仿真模拟多核执行，仅测试其内部M7核的编码执行效率

4、对于stm32mp1 的contex-A7核，MDK无法进行软件仿真

二、DSP运行测试时间表

标题栏缩写: 芯片型号-内核-测试方式-主频

测试运行函数	stm32F407- CM4-实测 168Mhz	stm32F407- CM4-仿真 168Mhz	stm32F769- CM7-仿真 219Mhz	stm32h723- CM7-仿真 550Mhz	stm32h745- Main-CM7-仿 真-480Mhz	Cortex- M7-虚拟 机-25Mhz	stm32mp151- A7-1Ghz
4096点-32位 浮点-FFT运算	2.746ms	3.229ms	2.512ms	1.010ms	1.130ms	41.670ms	不支持软件仿 真
4096点-块大 小256-阶数 12-FIR低通滤 波	1.283ms	1.174ms	1.375ms	0.457ms	0.399ms	18.743ms	不支持软件仿 真
自定义数字信 号处理流程	6.853ms	6.598ms	6.064ms	2.234ms	2.285ms	86.630ms	不支持软件仿 真

*测试时间实机通过芯片内部定时器计时，仿真则通过计算MDK工具中的断点运行时间差：

Internal	Thread
Mode	Privileged
Privilege	MSP
Stack	1227
States	0.00000730
Sec	
FPU	

*自定义数字信号处理流程代码如下：

```

//***** DSP处理流程开始 *****//
//信号抽样
for(i=0; i<FFT_LENGTH; i++)
{
    fft_buffer[2*i] = data_buffer[1024*4 + 4*i];
    fft_buffer[2*i + 1]= data_buffer[1024*4 + 4*i + 1];
}
//FFT计算（基4）
TIM_SetCounter(TIM3,0); //重设TIM3定时器的计数器值
timeout=0;
arm_cfft_radix4_f32(&scfft, fft_buffer);
time=TIM_GetCounter(TIM3)+(u32)timeout*65536; //计算所用时间
//下变频
tranfer_length = FFT_LENGTH - (FFT_LENGTH / DOWN_FEQ_MAX_RATE) * DOWN_FEQ_RATE ;
memset(data_buffer, 0 , sizeof(data_buffer));
memcpy(data_buffer, fft_buffer , sizeof(fft_buffer));
for(i=0; i < tranfer_length ; i++)
{
    u32 tranfer_index = (FFT_LENGTH / DOWN_FEQ_MAX_RATE) * DOWN_FEQ_RATE * 2;
    fft_buffer[i] = data_buffer[tranfer_index + 2*i];
    fft_buffer[FFT_LENGTH + i]= data_buffer[tranfer_index + 2*i + 1];
}
//转化为幅度谱
//双通道fir滤波
for(i=0; i < numBlocks; i++)
{
    arm_fir_f32(&scfir, fft_buffer + (i * blockSize), data_buffer + (i *
blockSize), blockSize);
}

for(i=0; i < numBlocks; i++)
{
    arm_fir_f32(&scfir, fft_buffer + FFT_LENGTH + (i * blockSize), data_buffer +
FFT_LENGTH + (i * blockSize), blockSize);
}
//***** DSP处理流程结束 *****//

```