

KeyStone I DSP[C665x 与 C6678] 视频教程

王 斌

广州创龙电子科技有限公司

演示环境

开发套件 TL665x-EasyEVM + SOM-TL665x[DSP C6655/57]

TL6678-EasyEVM + SOM-TL6678[DSP C6678]

仿真器 TL-XDS200 / TL-XDS560v2

开发环境 CCSv5[5.5]

CCSv7

相关产品

DSP + FPGA 板卡 核心板 SOM-TL665xF[DSP C6655/57 + Xilinx Artix 7]

SOM-TL6678F[DSP C6678 + Kintex 7]

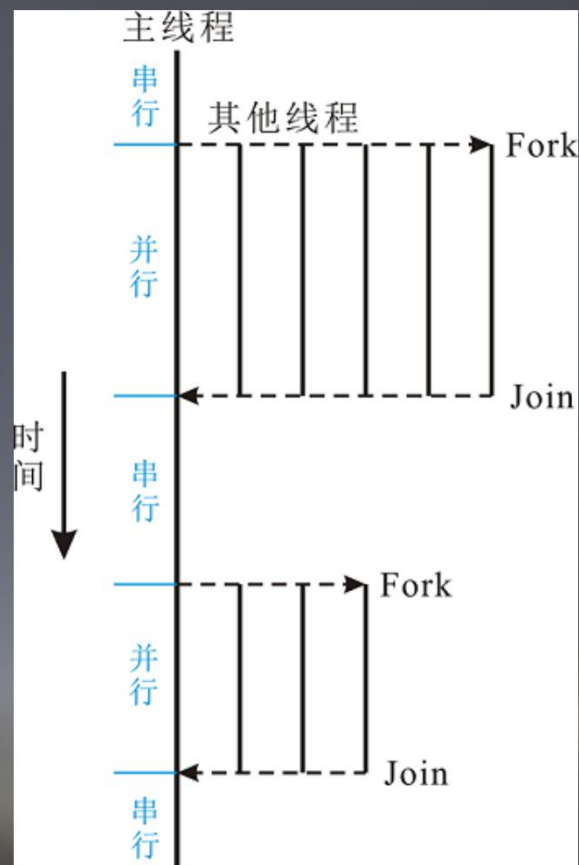
开发板 TL665xF-EasyEVM

TL6678F-EasyEVM

OpenMP 是跨平台的以共享内存为基础的依靠编译器预编译指令、运行时支持库以及一些影响运行行为的环境变量组成的多核开发软件框架。

- 跨平台
- 以共享内存为基础
- 支持 C/C++ 及 Fortran 语言
- 提供运行时 API

OpenMP 程序结构



```
#pragma omp 指令 [子句[ [,] 子句]...]  
{  
    .....  
}
```

parallel	开始并行执行语句。
for	在多个线程中并行执行 for 循环。
sections	包含多个可以并行执行的 sections 结构。
single	单线程（未必是主线程）执行。
master	主线程执行。
critical	任意时刻只能被单个线程执行。
barrier	指定屏障，用于同步所有线程。
taskwait	等待子线程完成。
atomic	确保指定内存位置将会原子更新。
flush	使线程当前内存数据与实际内存数据一致。
ordered	并行执行的 for 循环将按照循环体变量顺序执行。
threadprivate	指定变量为本地存储。

OpenMP 子句

子句	说明
default	控制 parallel 或 task 结构中变量数据共享属性
shared	parallel 或 task 结构中一个或多个变量为共享变量
private	一个或多个变量为本地变量
firstprivate	一个或多个变量为本地变量并且初始值为并行结构执行前的值
lastprivate	一个或多个变量为本地变量并且值为并行结构执行后的值
reduction	一个或多个变量为本地变量但是初始值根据不同的运算符来决定，执行完成后变量值会被更新
copyin	使线程本地变量值与主线程变量值相同
copyprivate	使属于 parallel 区域的变量的值在不同线程中相同
schedule	设置 for 循环并行执行方式 dynamic、guided、runtime 和 static
num_threads	线程数目
if	并行语句执行条件
nowait	忽略线程同步等待

TI OpenMP 支持 KeyStone I & II 及 Sitara 处理器 兼容 OpenMP 3.0 特性

TI OpenMP 运行时基于 GCC OpenMP Runtime(libgomp) 并且利用了 KeyStone SOC 架构硬件特性优势，从而降低系统开销。

OpenMP 2.x 版本支持 NonOS 及 RTSC 模式，在 RTSC 模式依赖 SYS/BIOS 及 IPC 组件。

- 多核导航（硬件队列）

- 硬件信号量

SOC		SDK
TMS320C667x Processor SDK RTOS for C667x		MCSDK
TMS320C665x Processor SDK RTOS for C665x		
AM572x	Processor SDK for AM57x	/
66AK2Hx	Processor SDK for K2H	MCSDK
66AK2Lx	Processor SDK for K2L	
66AK2Ex	Processor SDK for K2E	
66AK2Gx	Processor SDK for K2G	/

API	说明
<code>void omp_set_num_threads(int num_threads);</code>	设置 parallel 区域线程数目
<code>int omp_get_num_threads(void);</code>	获取线程数目
<code>int omp_get_max_threads(void);</code>	获取最大线程数目
<code>int omp_get_thread_num(void);</code>	获取当前线程序号
<code>int omp_get_thread_num(void) ;</code>	获取可用处理器数目
<code>int omp_in_parallel(void);</code>	返回为 true 代表当前出于并行区域
<code>void omp_set_dynamic(int dynamic_threads);</code>	允许或禁用动态线程调整
<code>int omp_get_dynamic(void);</code>	获取动态线程调整状态，允许或禁用
<code>void omp_set_nested(int nested);</code>	允许或禁用嵌套并行结构
<code>int omp_get_nested(void);</code>	返回并行结构嵌套状态，允许或禁用

OpenMP 资源使用

资源		使用情况
内存	L1P 缓存	32K 缓存
	L1D 缓存	32K 缓存
	L2 缓存	128K 缓存
	L2 内存	32K 堆 128K 栈
	MSMCSRAM 可缓存区域	不被 OpenMP 运行时使用 可以用于程序/数据
	MSMCSRAM 不可缓存区域	128K 必须为非缓存区域
	DDR3	不被 OpenMP 运行时使用 可以用于程序/数据
外设	硬件信号量	7 个
	QMSS 硬件队列	11 个
	QMSS 内存区域	1 个

```
#pragma omp parallel for  
for(i = 0; i < 240000; i++)  
{  
    Gray[i] = (Image[3 * i + 2] * 76 + Image[3 * i + 1] * 150 + Image[3 * i] * 30) >> 8;  
}
```

o(n_n)o~

KeyStone I DSP
[C665x 与 C6678]
视频教程

广州创龙电子科技有限公司