

一.多核通信方案总体流程

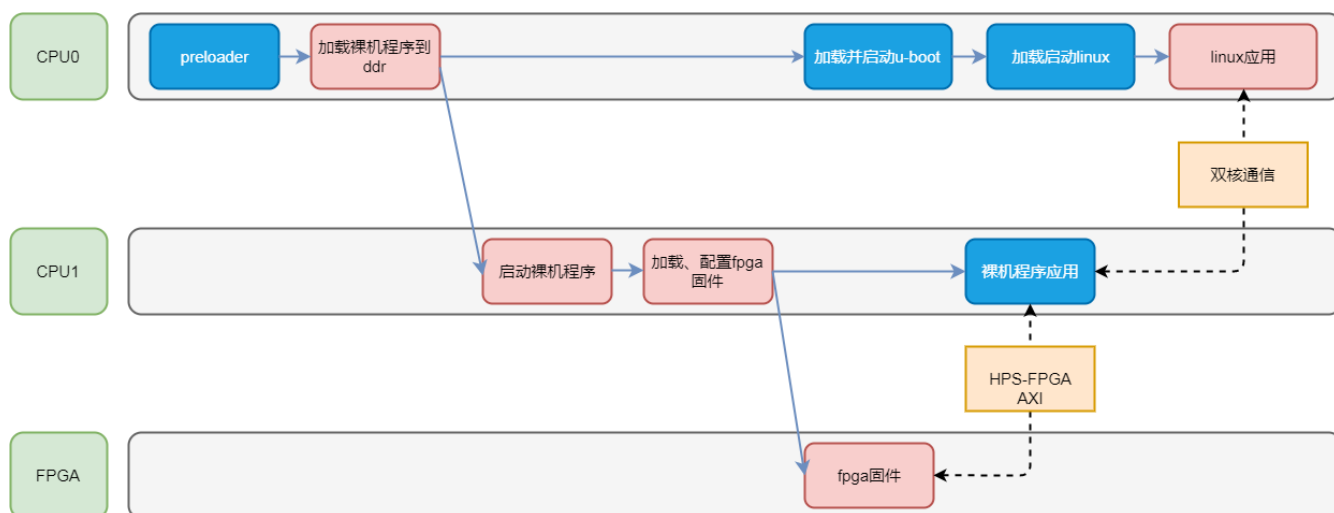


图1 多核通信方案流程

AMP多核通信方案如上图所示，分为硬核和软核，硬核部分有ARM的CPU0和CPU1两个核，软核部分为FPGA，系统上电时会先运行SOC厂家固化在引导ROM中的代码，该部分代码根据硬件设置选择对应的启动方式，然后从相应的存储介质中加载preloader到ddr中运行。接下来是每个模块主要功能：

1.preloader

- 初始化ddr，管脚复用，中断向量，定时器，串口，sd/mmc和qspi
- 加载裸机程序并从cpu1启动
- 加载u-boot并启动u-boot

2.u-boot

- 初始化其他需要的外设功能（如usb、lcd等）
- 设置相关环境和启动参数
- 加载linux内核并运行

3.linux

- 实现内存预留，用于裸机程序使用。
- 改造Linux内存fixmap映射，vmalloc映射。
- 改造Linux中断管理机制，避免中断影响裸机中断执行。
- 改造Linux的SMP机制，实现硬件SMP，实质软件是AMP模式。
- 改造Linux的L2缓存机制，实现Linux缓存机制不影响裸机的缓存。

4.裸机程序

- 内存映射划分
- 加载并启动fpga固件
- 划分中断和设置中断处理函数
- 进入main loop循环

二.内存分配

1.物理内存划分

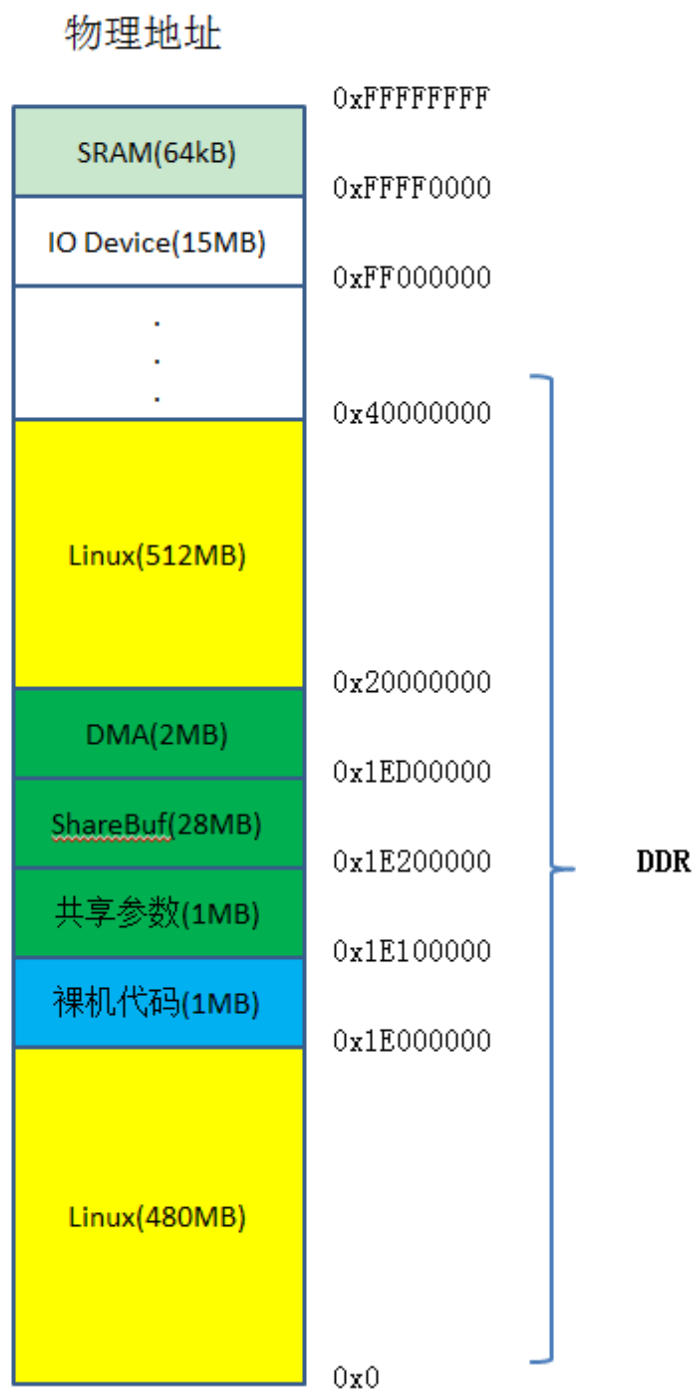


图 2 物理内存分布

上图是amp方案的物理内存分布。具体分配如下：

- 片上静态内存SRAM：0xFFFF0000~0xFFFFFFFF，大小64KB
- SOC IO空间：0xFF000000~0xFFFFEFFF，大小15MB
- linux私有空间：0x00000000~0x1DFFFFFF和0x20000000~0x3FFFFFFF，大小982MB
- 裸机代码存放空间：0x1E000000~0x1E0FFFFF，大小1MB
- 共享区域：0x1E100000~0x1FFFFFFF，大小31MB

- 共享参数区域：0x1E100000~0x1E1FFFFF，大小1MB
- 共享传输数据块：0x1E200000~0x1ECFFFFF，大小28MB
- DMA：0x1ED00000~0x1FFFFFFF，大小2MB

2. linux虚拟地址映射

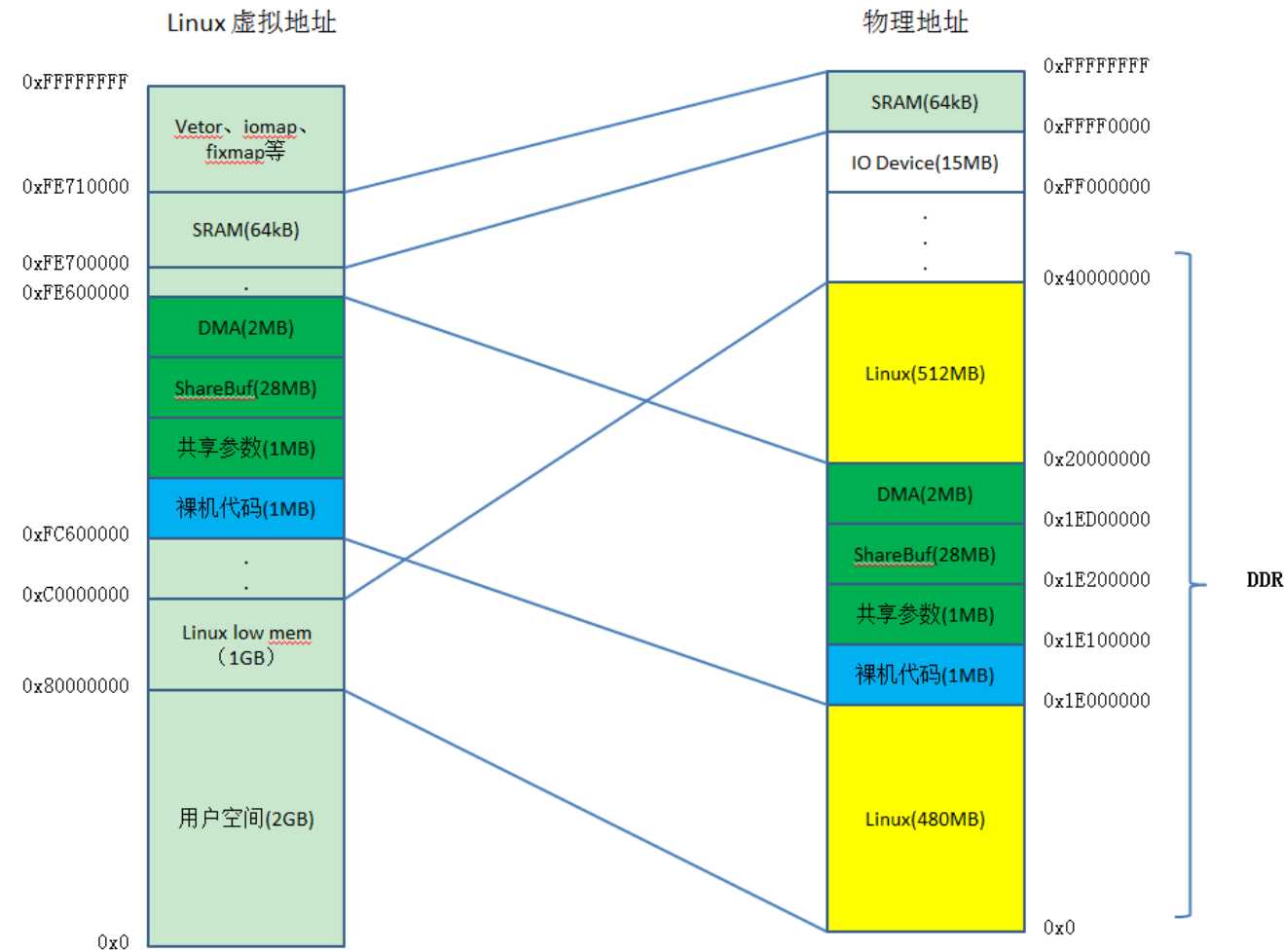


图 3 linux虚拟地址映射

3.裸机虚拟地址映射

- CPU1收到中断通知，读取并验证64KB的共享内存数据
- CPU1往ddr共享内存区域写入另外一个特定的64KB数据，触发CPU0中断
- CPU0收到中断通知，读取并验证64KB的共享内存数据

2.测试

以上方案的测试方法是利用示波器测量每个循环切换GPIO管脚的电平频率来计算通信速率。计算公式如下：

$$\text{freq(Hz)} * 2 * 2(w) * 2(r\&check) * \text{size(KB)}$$

公式中各项含义是：

freq(Hz)：示波器频率，代表每秒的周期数

freq(Hz) * 2：每秒的通信循环次数

2(w)：每个通信循环有两次写操作

2(r&check)：每个通信循环有两次读取验证操作

size(KB)：每次读写的共享内存数据的长度

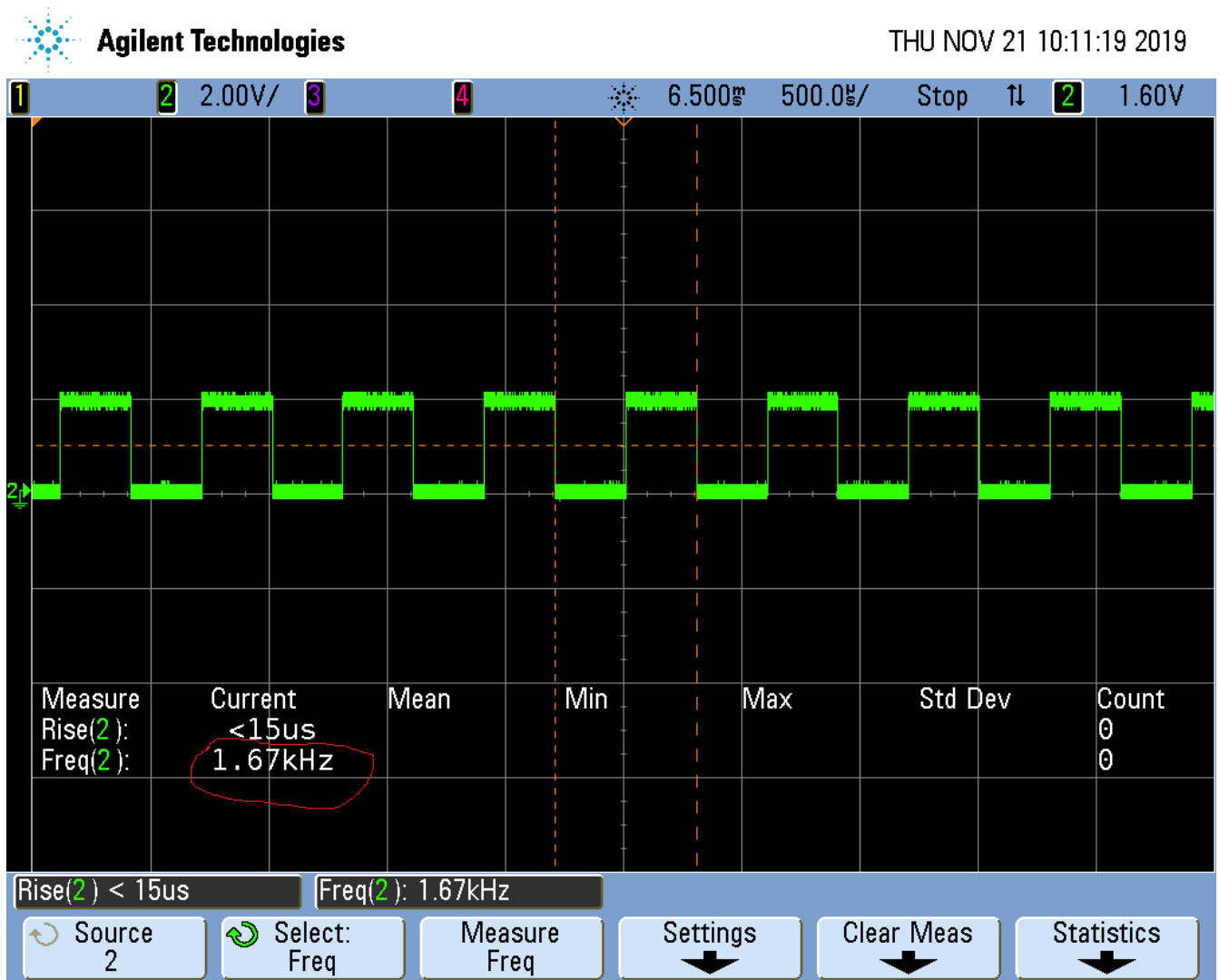


图6 ddr共享内存测试

图6是使用ddr作为共享内存测试的波形图，freq为1670，size为64，因此通信速率为 $1670\text{Hz} * 2 * 2w * 2r\&check * 64\text{KB} = 875560960\text{B/s} = 7.00\text{Gbps}$

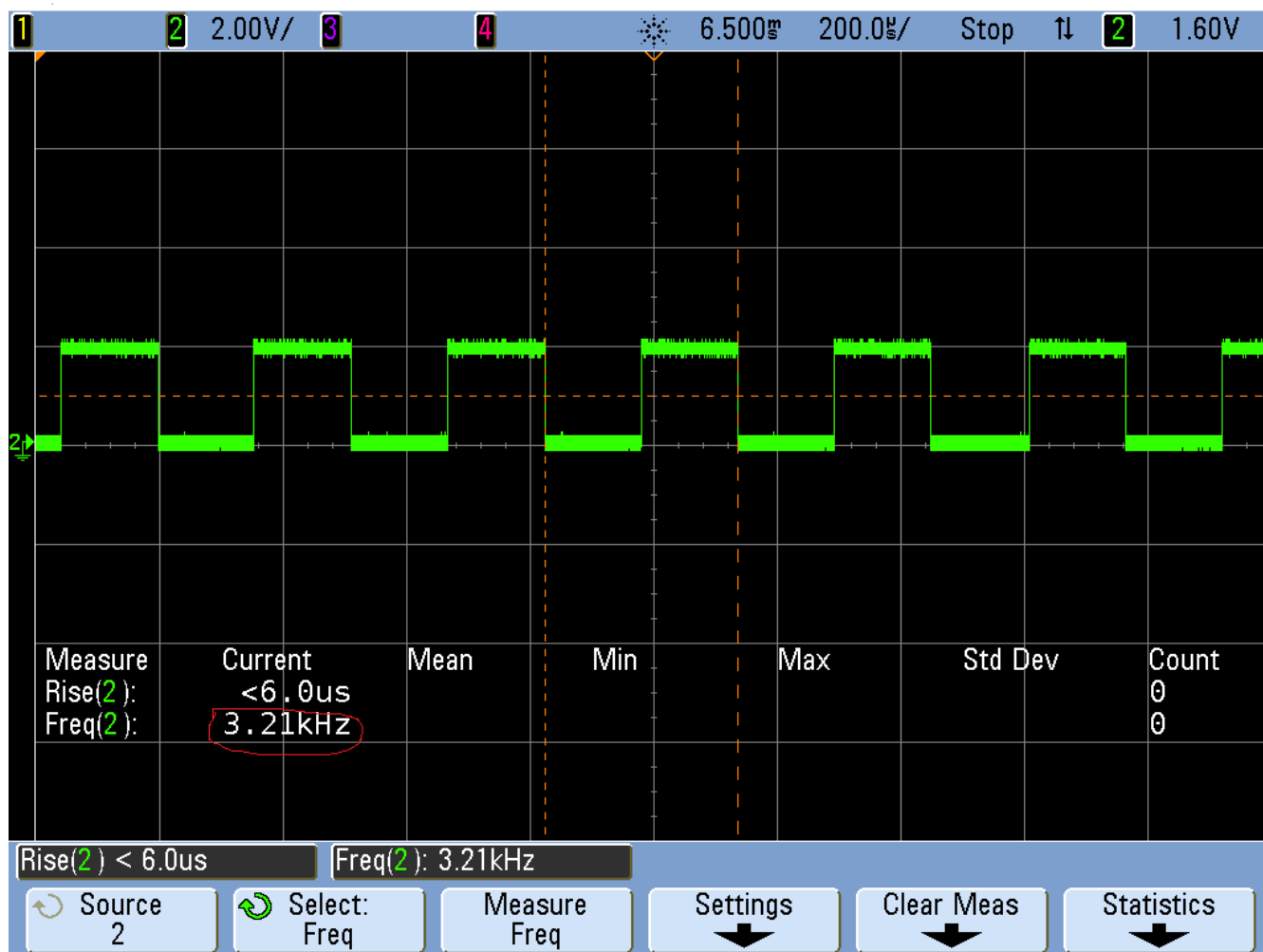


图7 sram共享内存测试

上图是使用sram作为共享内存测试的波形图，freq为3210，size为32，因此通信速率为 $3210\text{Hz} * 2 * 2w * 2r\&check * 32\text{KB} = 841482240\text{B/s} = 6.73\text{Gbps}$

3.小结

1，上面的结果是实际测量得到的，但是理论上的通信速率会比测试结果要快。原因是CPU响应中断后的模式切换和中断函数处理需要消耗额外的时间。另外操作系统有任务调度的功能，在通信过程中操作系统会调度相应的时间给其他任务，在相应CPU1触发的中断后也需要任务调度，保存现场，因此会消耗一定的CPU时间。

2，上面测试标准，ddr传输大小为64k，sram为32kb，同时也基于ddr 64k，128k，256k大小测试，速度基本恒定在6.5-7Gbps。

四.总结

此次软件框架设计涉及的技术面比较广泛，技术深度高，主要体现在以下方面：

1，对arm核的流水线机制，缓存机制，中断控制机制等核心机制熟练。

2，对soc芯片的电源域、时钟域、System manager、Scan Manager、FPGA Manager等核心模块熟悉。

3，操作系统方面：除了一些必要的驱动编写与测试外，主要难度在于操作系统内核改造：如内存管理机制改造，中断管理机制改造，缓存一致性改造，SMP机制改造等

此方案不足之处：

- 1，涉及了操作系统，编写应用的研究人员必须在操作系统下编程，因此对操作系统初步了解。
- 2，涉及了操作系统，linux端应用在线调试不能用仿真器在线调试，只能用网络手段调试（裸机部分可以）。
- 3，由于没有使用二所提供的IDE界面进行开发，属于纯代码编程。后续若需要，将修改IDE安装包的源代码。
- 4，由于裸机与fpga通信二所已经实现了，此方案没有做这方面测试，只实现了arm端启动fpga固件加载并启动运行。
- 5，arm双核通信方面，没在有压力测试环境下进行测试。