

# 基于 Zynq 多核运行设计

邢艳芳<sup>1</sup>, 朱金付<sup>2</sup>, 周晓梅<sup>1</sup>

(1. 中国传媒大学南广学院, 江苏 南京 211172;

2. 东南大学, 江苏 南京 210000)

**摘要:**多核处理器主要包括两种,即对称多核处理器(SMP)和非对称多核处理器(AMP)。目前大部分通用多核处理器是SMP,各个CPU是平等的,共享操作系统、内存和外设等资源;AMP大多是嵌入式多核处理器采用的架构,由一个主CPU控制系统运行和资源分配,从CPU执行主CPU的命令或者预定义任务。Zynq是Xilinx推出的全可编程芯片,是以ARM为核心,以FPGA作为可编程外设的全新架构处理器。Zynq包括两个可以同时独立运行可执行程序的Cortex-A9处理器,是一种非对称多核芯片。主处理器控制整个系统,从处理器执行主处理器的指令或者预定义任务,两个处理器是一种主从关系。文中CPU0是主处理器,控制系统和共享资源,CPU1是从处理器,OCM做CPU0和CPU1通信的共享内存。实现了启动Zynq的双核CPU,各自同时运行裸机程序,通过共享内存,实现了CPU之间的通信,并将运行信息在OLED上显示出来。

**关键词:**Zynq;非对称多核芯片;共享内存;片上内存;主从关系

中图分类号:TP319

文献标识码:A

文章编号:1673-629X(2018)03-0060-03

doi:10.3969/j.issn.1673-629X.2018.03.012

## Design of Multi-core Processing Based on Zynq

XING Yan-fang<sup>1</sup>, ZHU Jin-fu<sup>2</sup>, ZHOU Xiao-mei<sup>1</sup>

(1. Nanguang College of Communication University of China, Nanjing 211172, China

2. Southeast University, Nanjing 210000, China)

**Abstract:** Multi-core processors mainly consists of two types, symmetrical multi-core processors (SMP) and asymmetric multi-core processors (AMP). At present most general multi-core processors are SMP which are equal among different CPUs, sharing the operating system, memory, peripherals and other resources. Most of AMP is the architecture adopted by the embedded multi-core processor, where a master CPU controls system operation and resources allocation, and the slave CPUs execute commands or predefined tasks from master CPU. Zynq is fully programmable chip launched by Xilinx recently, and is a processor with new architecture with ARM as its core and FPGA as its programmable peripheral. As an AMP, it includes two Cortex-A9 processors which can be configured to concurrently run independent software executables, in which a master processor controls the system, the slave processors execute the instruction or predefined tasks from the master processor, and it is a master-slave relationship between them. In this paper, CPU0 is treated as the master which conducts the system control and resources sharing, and CPU1 as the slave, and the on-chip memory (OCM) is used as shared memory for communication between CPU0 and CPU1. The startup of double-kernal CPU of Zynq is realized, and they run respective bare-metal program simultaneously. By sharing memory, the communication between CPUs is realized and the running information are displayed in the organic light-emitting diode (OLED).

**Key words:** Zynq; asymmetric multiprocessing; shared memory; on-chip memory; master-slave relationship

## 0 引言

多核处理器主要包括两种,即对称多核处理器(symmetrical multi-core processors, SMP)和非对称多核处理器(asymmetrical multi-core processors, AMP)。目前大部分通用多核处理器是SMP,各个CPU是平等

的,共享操作系统、内存和外设等资源,Windows和Linux都有比较成熟稳定的任务调度算法。AMP大多是嵌入式多核处理器采用的架构,由一个主CPU控制系统运行和资源分配,从CPU执行主CPU的命令或者预定义任务,其实时性比较强,适合于一个CPU做

收稿日期:2017-04-15

修回日期:2017-08-24

网络出版时间:2017-12-04

基金项目:江苏省高校自然科学基金项目(14KJD520011)

作者简介:邢艳芳(1981-),女,硕士,讲师,研究方向为嵌入式系统;朱金付,副教授,研究方向为信息系统。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20171204.1647.002.html>

任务调度,其他 CPU 各自独立处理实时任务,并通过共享内存实现各个 CPU 之间的通信。AMP 目前还没有成熟的任务调度算法。

Zynq 是 Xilinx 推出新的全可编程芯片<sup>[1-2]</sup>,是以 ARM 为核心,以 FPGA 为可编程外设的一种全新架构处理器,其 ARM 是由 2 个 Cortex-A9 CPU 组成的非对称多核处理器系统,目前对 Zynq 的应用研究大都是基于单核处理器。文中提出在 Zynq 的 CPU0 上运行一个裸机程序,由 CPU0 控制系统的初始化、共享资源和启动 CPU1。CPU1 启动后,同时也运行一个裸机程序,通过共享 OCM 和 CPU0 通信,分时共享 OLED,实时显示 CPU 运行的相关信息。

## 1 Zynq 双核运行原理

Zynq 是一个 FPGA 做外设的 A9 双核处理器,是一个可扩展处理平台。因此,它的启动流程与传统的 ARM 处理器类似,和 FPGA 完全不同。Zynq 的启动配置需要多个处理步骤,通常情况,需要包含以下三个阶段<sup>[3-6]</sup>。

(1)阶段 0:在器件上电运行后,处理器就自动开始 Stage-0 Boot,即片内 BootROM 中的代码被执行,上电复位或者热复位后,处理器执行的不可修改的代码。

(2)阶段 1:在 BootROM 初始化 CPU 和一些外设后,读取下一个阶段所需的程序代码 FSBL,即 first stage boot loader。它可以由用户修改控制的代码。

(3)阶段 2:这是用户基于 BSP 的裸机程序,也可以是第二阶段操作系统的启动引导程序。这个阶段代码完全是在用户的控制下实现的。

CPU0 作为主处理控制共享资源,如果 CPU1 需要控制一个共享资源,必须请求 CPU0 并让 CPU0 控制资源。OCM 被作为两个 CPU 通信共享资源,相比 DDR 内存,OCM 具有非常高的性能和低延迟访问的特点。

CPU0 和 CPU1 共享 OLED 外设,并显示相关运行信息。CPU0 和 CPU1 分时占用 OLED 资源,各占用 5 s,相互之间通过 OCM 的 0xFFFF0000 地址作为共享内存,进行通信协调。当 0xFFFF0000 中的内容为 0 时,OLED 由 CPU0 占有,当 0xFFFF0000 中的内容为 1 时,OLED 由 CPU1 使用。OCM 是片上内存,访问速度比片外 DDR 快,因此,两个 CPU 通信的实时性比较好。

Zynq 是非对称多核处理器系统,CPU0 和 CPU1 各自占用独立的 DDR 空间,其中 CPU0 使用的地址空间为 0x00100000 到 0x001FFFFFF,而 CPU1 使用的地址空间为 0x00200000 到 0x002FFFFFF。

## 2 硬件设计

Zynq 由 PS + PL 构成,其中 PS 部分和普通的 ARM 芯片一样,可以独立使用;PL 部分就是传统意义的 FPGA,可以方便定制外设电路 IP,也可以进行相关的算法设计。普通的 ARM,外设是固定的;而 Zynq 硬件外设是不固定的,可以利用 PL 部分灵活地定制外设挂在 PS 上,这是 Zynq 最大的特点,也是 Zynq 灵活性的一个重要表现。两个应用程序运行在 PS 部分的 Cortex-A9 双核处理器,因此,PS 部分自带的相关硬件使用 Xilinx 默认配置即可,但双核 CPU 运行的相关信息需要在 OLED 上显示<sup>[7-9]</sup>,因此需要在 PL 部分定制一个 OLED 的 IP 核,通过 AXI 总线和 PS 进行通信。在 PS 部分对 OLED 进行驱动设计和显示设计。

Zynq 的测试板 ZedBoard 上使用 Inteltronic/Wise-chip 公司的 OLED 显示模组 UG-2832HSWEG04。这是一款单色被动式显示屏,分辨率为 128 \* 32,驱动电路采用所罗门科技的 SSD1306 芯片。ZedBoard 测试板使用的 OLED 采用 SPI 模式控制,使用的信号线和电源线有如下几条:

RST(RES):硬复位 OLED;

DC:命令/数据标志(0,读写命令;1,读写数据);

SCLK:串行时钟线;

SDIN:串行数据线;

VDD:逻辑电路电源;

VBAT:DC/DC 转换电路电源。

在 SPI 模式下,每个数据长度均为 8 位,在 SCLK 的上升沿,数据从 SDIN 移入到 SSD1306,并且是高位在前的。

其硬件建构框图如图 1 所示。

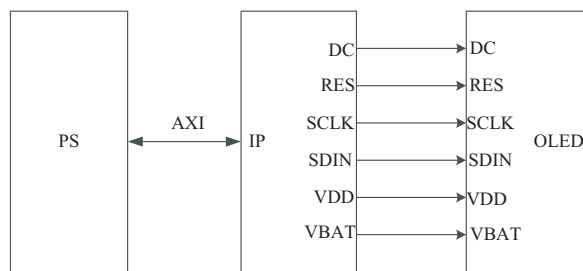


图 1 硬件设计图

## 3 软件设计

软件设计部分包括 CPU0 应用程序和 CPU1 应用程序。CPU0 上运行裸机程序是主系统,主要负责系统初始化、控制 CPU1 的启动、与 CPU1 的通信和在 OLED 上显示信息。CPU1 上运行的裸机应用程序主要负责与 CPU0 上运行的程序通信、与 CPU0 共享 OLED<sup>[10-13]</sup>。

FSBL 总是运行在 CPU0 上,这是上电复位后 PS

运行的第一个软件应用程序。负责配置 PL 和将应用程序 ELF 文件复制到 DDR 内存。加载应用程序到 DDR 内存后,FSBL 开始执行第一个被加载的应用程序。FSBL 加载完应用程序后,跳转到 0x00100000 处执行 CPU0 程序。

CPU0 里需配置 MMU,关闭 Cache,使 OCM 的物理地址为 0xFFFF0000 到 0xFFFFFFFF 和 0x00000000 到 0x0002FFFF。

关闭 Cache 后,CPU0 执行 SEV 汇编指令,激活 CPU1,CPU1 到 OCM 的 0xFFFFFFFF0 地址读取其数值,即 CPU1 应用程序的地址,CPU1 应用程序将从该地址执行。

判断共享内存 COM\_VAL 是否为 0,如果不为 0,将继续等待判断。如果为 0,将在 OLED 上显示相关信息,延时 5 s,将 COM\_VAL 设置为 1,把 OLED 资源让给 CPU1 使用。然后继续判断共享内存 COM\_VAL 是否为 0,等待 CPU1 把 OLED 使用权让给 CPU0。

CPU0 在 DDR 执行的物理地址由链接脚本设置,将其运行地址设置为 0x00100000,链接脚本文件相关内存地址设置见表 1。

表 1 CPU0 内存地址

Name	Base Address	Size
ps7_dds_0_AXI_BASEADDR	0x00100000	0x00100000
ps7_ram_0_AXI_BASEADDR	0x00000000	0x00030000
ps7_ram_1_AXI_BASEADDR	0xFFFF0000	0x0000FE00

CPU0 程序流程如图 2 所示。

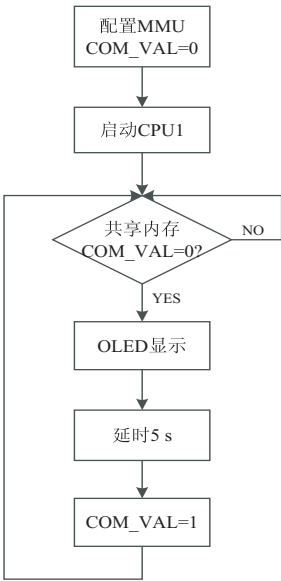


图 2 CPU0 程序流程

在 AMP 系统中,因为各 CPU 独立使用资源,所以 CPU1 里仍需要配置 MMU,关闭 Cache。激活 CPU1 后,将从 DDR 的 0x00200000 地址开始执行应用程序。

在关闭 Cache 后,CPU1 需等待 CPU0 将共享内存设置为 1,然后在 OLED 上显示相关信息。延时等待

后,清除共享内存,将 OLED 让给 CPU0 使用。

CPU1 在 DDR 执行的物理地址由链接脚本设置,将其运行地址设置为 0x00200000,链接脚本文件相关内存地址设置见表 2。

表 2 CPU1 内存地址

Name	Base Address	Size
ps7_dds_0_AXI_BASEADDR	0x00200000	0x00300000
ps7_ram_0_AXI_BASEADDR	0x00000000	0x00030000
ps7_ram_1_AXI_BASEADDR	0xFFFF0000	0x0000FE00

CPU1 程序流程如图 3 所示。

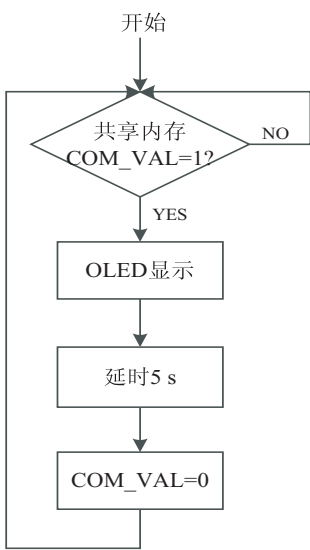


图 3 CPU1 程序流程

4 设计验证

在完成软硬件设计后,需要把软件部分下载到 DDR 中运行,把硬件比特流配置到 Zynq 的 PL 部分。通过 Xilinx 的 BootGen 工具,生成 Zynq 能识别的合法镜像文件 BOOT. BIN,其中包含 FSBL 文件、system. bit 文件、CPU0 文件和 CPU1 文件。将 BOOT. BIN 复制到 SD 卡,将测试板 ZedBoard 设置为 SD 卡启动,启动 ZedBoard 后会看到 OLED 被 CPU0 和 CPU1 分时占用信息,如图 4 所示。

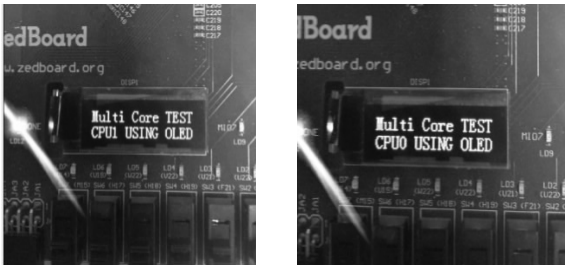


图 4 实验结果

5 结束语

Zynq 器件包括一个 ARM 双核 AMP 处理器,通过

## 5 结束语

采用 SketchUP 软件与 ArcGIS 平台相结合的方式实现三维空间数据库,利用二者的交互解决二维 GIS 数据批量生成三维模型的方法,解决了以往三维场景中三维模型只能显示,不能查询编辑的问题,实现了数据的查询与编辑,对三维模型进行了优化处理,以最少的系统资源消耗,较好地对三维建筑实景进行显示浏览。三维空间数据库体验效果好,立体感强,给人一种身临其境的感觉,且三维场景浏览速度快。实验结果证明了该方案的可行性,同时为三维空间数据库的建设提供了技术参考。

### 参考文献:

- [1] 万宝林. 3DSMAX 与 SketchUp 的三维城市建模技术实验对比分析[J]. 测绘地理信息, 2015, 40(2): 23-25.
- [2] 黄风华, 晏路明. 一种基于 CityGML 的虚拟三维数字城市建模研究[J]. 计算机应用与软件, 2013, 30(5): 104-107.
- [3] 王星捷, 李春花. 基于 Unity3D 平台的三维虚拟城市研究与应用[J]. 计算机技术与发展, 2013, 23(4): 241-244.
- [4] 邓德标, 方源敏, 高晋宁. 数字城市三维景观模型的批量添加及管理研究[J]. 测绘通报, 2012(S1): 249-252.
- [5] 方光辉, 胡金星, 周廷刚, 等. 基于 GIS 的三维数字城市基础平台原型系统设计与实现[J]. 西南大学学报: 自然科学版, 2011, 33(12): 139-144.
- [6] 冯梅. 基于 LiDAR 和航空影像的三维建模方法探讨[J]. 测绘通报, 2011(12): 12-14.
- [7] 许捍卫, 房晓亮, 任家勇, 等. 基于 SketchUp 的城市三维建模技术[J]. 测绘科学, 2011, 36(1): 213-214.

(上接第 62 页)

CPU0 控制系统和启动 CPU1, 实现了两个 CPU 同时运行裸机程序, 并通过 OCM 实现了两个 CPU 的通信, 经过 24 小时不间断测试, 双核运行稳定, 解决了该处理器双核同时运行的问题。

### 参考文献:

- [1] 何宾. Xilinx all programmable Zynq-7000 SoC 设计指南[M]. 北京: 清华大学出版社, 2013: 384-391.
- [2] 陆佳华, 江舟, 马岷. 嵌入式系统软硬件协同设计实战指南: 基于 Xilinx Zynq[M]. 北京: 机械工业出版社, 2013: 17-18.
- [3] 陆启帅, 陆彦婷, 王地. Xilinx Zynq SoC 与嵌入式 Linux 设计实战指南[M]. 北京: 清华大学出版社, 2014: 138-152.
- [4] 叶琴, 谢捷如. 基于 SoC 和嵌入式 Linux 的数据采集系统设计[J]. 计算机技术与发展, 2015, 25(8): 203-207.
- [5] 肖灵芝, 蒲林, 韩俊刚, 等. 异构多核图形处理器存储系统设计与实现[J]. 电子技术应用, 2013, 39(5): 38-40.

- [8] 吴书金, 叶华平, 唐洋洋. 基于二维 GIS 数据和三维模型库快速构建营房土地三维模型的研究[J]. 后勤工程学院学报, 2014, 30(6): 83-89.
- [9] 肖坤, 闫浩文, 王中辉. 基于 3ds MAX 的三维数字城市可视化模型构建方法[J]. 测绘与空间地理信息, 2014, 37(10): 68-70.
- [10] 曹兆峰, 何燕兰, 李胜才. 基于 Sketchup 和 ArcGIS 的数字城市三维建模技术[J]. 地理空间信息, 2014, 12(5): 46-47.
- [11] GRABE V, STRUTZ A, BASCHWITZ A, et al. Digital in vivo 3D atlas of the antennal lobe of *Drosophila melanogaster* [J]. Journal of Comparative Neurology, 2015, 523(3): 530-544.
- [12] BETTAMER A, ALLAOUI S, HAMBLI R. Using 3D digital image correlation to visualise the progress of failure of human proximal femur[J]. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 2015, 5(4): 233-240.
- [13] GELFGAT A Y. Visualization of three-dimensional incompressible flows by quasi-two-dimensional divergence-free projections in arbitrary flow regions[J]. Theoretical and Computational Fluid Dynamics, 2016, 97(4): 143-155.
- [14] ZHAO Zhongyuan. Research on modeling precision and standard of 3D digital city[J]. Procedia Environmental Sciences, 2012, 12: 521-527.
- [15] YANG X, WENG J N, XIA Y B, et al. 3D building modeling, organization and application in digital city system[J]. Science China Technological Sciences, 2010, 53(1): 134-142.

- [6] 李金泉, 王东, 胡文振, 等. 基于 ZYNQ 异构多核处理器的人体动作识别系统[J]. 单片机与嵌入式系统应用, 2016, 16(8): 27-29.
- [7] 邢艳芳, 张延冬. 基于 Zynq 的 OLED 驱动设计[J]. 液晶与显示, 2014, 29(2): 224-228.
- [8] 张龙, 冯帅. 基于 ZYNQ 的视频图像处理平台的设计[J]. 计算机与数字工程, 2016, 44(12): 2519-2523.
- [9] WANG Jiannong, WANG Wei. The common data acquisition system based on Arm9 [C]//Proceedings of international conference on electronic measurement & instruments. [s.l.]: IEEE, 2011: 324-327.
- [10] STEVENS W R, RAGO S A. Advanced programming in the UNIX environment[M]. [s.l.]: Addison-Wesley, 2014.
- [11] JONES M T. GNU/LINUX application programming[M]. [s.l.]: Charles River Media, 2008.
- [12] 孟繁星. 基于 ZYNQ-7000 的视频图像处理系统设计[J]. 信息通信, 2015(8): 60-62.
- [13] Solomon-Systech Instruments. The datasheet of the SSD1322 Solo-mon-Systech Instrument [R]. Hong Kong: Solomon Systech Limit-ed, 2008.