

基于 FPGA 片上 PowerPC 和 VxWorks 的 TCP IP 通信

引言

随着应用的不断普及和深入，在设计嵌入式系统时，往往需要同时优化众多因素，如成本、功率、尺寸、性能、灵活性、产品上市时间、设计开发周期，以及可靠性等。Xilinx 公司推出的嵌入式处理器设计平台 Virtex-II Pro 和 Virtex-4 系列器件可以满足上述要求，其高性能的设计工具为设计者提供了完善的软硬件协同设计能力。本文以 Virtex-II Pro P20 芯片内嵌 PowerPC405 硬核为处理器，选用目前广泛使用的 VxWorks 操作系统，阐述了内嵌 PowerPC 嵌入式设计的过程和要点，详细介绍了 TCP/IP 实时传输的整个设计过程，实验结果表明这种构架模式具有很好的应用前景。

基于 FPGA 片上 PowerPC 的嵌入式系统开发

随着嵌入式技术的发展，针对当前嵌入式处理器的应用，Xilinx 公司推出了两款基于 FPGA 的 32 位嵌入式处理器内核。一种是采用先进 IP 植入技术实现的嵌入在 Virtex-II Pro 和 Virtex-4 器件中的 PowerPC405 处理器内核(这是业界目前唯一的嵌入式硬核)，另一种是 MicroBlaze 32 位嵌入式处理器软核。硬核的好处是能够提供更快的数据处理能力，而软核则具有更好的灵活性，在目标器件中可以进行任意配置。由于硬核在速度和资源上具有优势，因此本文采用了 PowerPC405 处理器，它支持复杂嵌入式的应用模式。由于同时需要嵌入式 RTOS VxWorks，而 FPGA 内部的 BlockRAM 往往不能满足提供较大存储空间的要求，因而一般要由外部的存储器阵列实现。

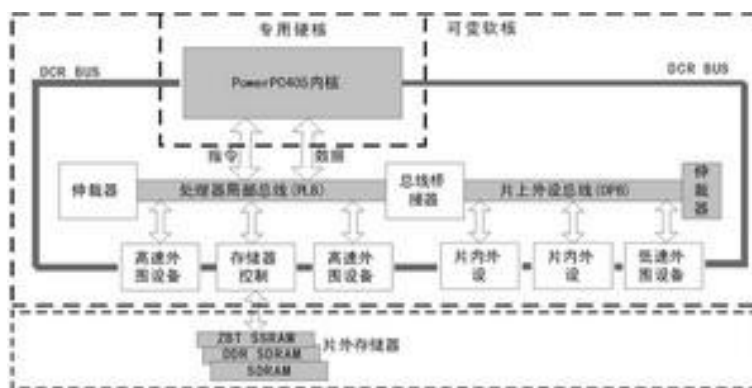


图 1 PowerPC405 复杂嵌入式应用模式

如图 1 所示，该体系结构主要应用于需要 RTOS 支持的数据处理、软件、控制等实时系统以及以 FPGA 为核心的复杂应用中。Virtex-II Pro 系列器件采用 IBM CoreConnect 总线技术，该总线包括：处理器局部总线 (PLB)、片上外设总线 (OPB) 和设备控制寄存器总线 (DCR)。PLB 总线为片内的高速数据通道，通常连接高速外设、DMA 存储控制器等，OPB 总线则通常用于连接速率较低的片上外设，二者通过总线桥接器与处理器总线连接，从而保证 PLB 总线的高速特性。DCR 总线用于实现 PowerPC405 的通用寄存器与逻辑设备控制寄存器的数据通信。

针对基于 PowerPC405 的平台 FPGA 设计，Xilinx 公司推出了完整的软硬件协同设计工具 EDK (Embedded Development Kit)，它是一个专用于 FPGA 内部 32 位嵌入式处理器的集成化开发工具包，并提供硬件和软件协同设计的能力，从而极大地缩短了设计周期。在软硬件协同设计开发平台上，嵌入式软件工程师仍然可以独立地进行软件设计，硬件逻辑工程师也就可以继续采用过去的设计方法，如图 2 所示。

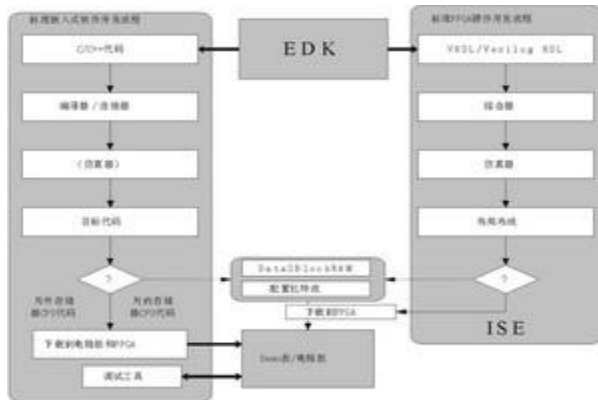


图 2 嵌入 PowerPC405 的平台 FPGA 软硬件协同设计

在实际的软硬件协同开发时，如果不使用嵌入式 RTOS，PowerPC405 可以起到类似单片机的作用，软件开发就变得相对简单。而在一些大型项目的开发过程中，嵌入式 RTOS 往往是必需的，因此，研究嵌入式 RTOS 应用开发及其 BSP 的移植具有关键意义。

基于 PowerPC405 的 VxWorks 下的 TCP/IP 通信整体系统设计

VxWorks 提供了丰富而标准的 socket 编程接口，可以方便地实现 TCP/IP 通信，同时，其多任务机制可以高效地完成实时任务的产生、调度、资源分配机制。本文在 Virtex

-II Pro 开发平台上利用 PowerPC405 内嵌处理器和 PC 机进行 TCP/IP 通信，在底层使用 100Mbps 以太网作为传输媒介，实现了 Vxworks 嵌入式系统控制的文件实时传输。可以利用 EDK 中的 BaseSystem BuilderWizard 来快速构建基于 PowerPC405 的系统。首先设定参考时钟频率、处理器时钟、总线时钟，如果在设计中包含有 100M 网口，那么总线时钟必须选择 100M 以上的频率。然后选取相应的外设，RS232 串口、100M 以太网口、SDRAM 等，其中 plb_bram_if_cntrl 外设是必选的，该控制器完成对 BlockRAM 的控制，保持 CPU 处于一个确定的状态。系统设计架构如图 3 所示，这里用到了 EDK 内带的 Ethernet MAC LogiCore(图中的 EMAC)。

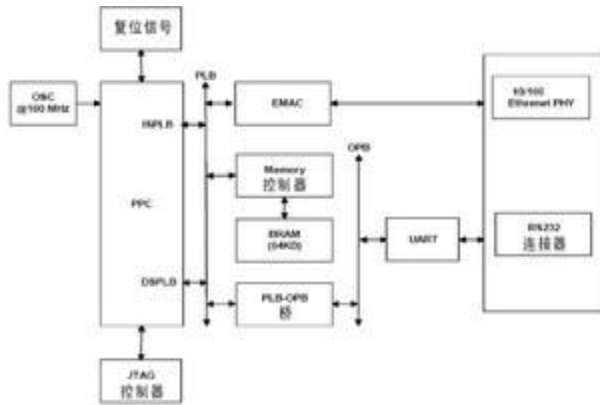


图 3 基于 PowerPC405 的 TCP/IP 通信整体设计框图

BSP 开发和移植

嵌入式操作系统开发中一项非常重要的任务，就是 BSP（板级支持包）的开发。这是一项非常复杂而繁琐的工作，Xilinx 公司提供的 EDK 套件，可以在很大程度上减轻开发者的工作。EDK 中的 BSP 生成器（BSPgen）可以根据不同的微处理器、外设和 RTOS 组合自动产生用户可裁剪的 BSP。它包含了系统所必需的支持软件，包括 Boot 代码、设备驱动和 RTOS 的初始化。利用 BSPgen 可以把 Xilinx 器件驱动打包到 BSP 的子目录下，并且把 Xilinx 器件驱动与 VxWorks 及其 Tornado 集成开发环境无缝集成，充分减少开发周期。

但是，BSPgen 生成的 BSP 只是一个固定的 BSP 模板文件，不能自动设置 RAM/ROM 的存储器映射，不支持用户通过 BSPgen 流程自定义的核/驱动，不能自动集成总线错误检测，而且，Caches 在缺省情况下被禁止的。

对于 BSP 开发人员来讲，借助 EDK 的 BSPgen，还必须要做以下几项工作：

- 1) 准确定义 RAM/ROM 边界地址，修改 configure.h 和 makefile 文件(两者的地址定义必须相匹配)；
- 2) 增加不能与 VxWorks 无缝集成的其他器件的驱动文件，如总线错误检测及报告、关键性中断、GPIO 接口、I2C 接口、SPI 接口等；
- 3) 配置以太网参数，设置缺省的 IP 地址、MAC 地址(sysNet.c 文件中)；
- 4) 配置内核服务选项，禁止/使能 Cache 和 RAM，设置 TCP/UDP/IP 参数等；
- 5) 增加总线错误检测报告和关键中断支持文件。

经过以上修改之后，将 BSPgen 产生的 BSP 放在 Tornado 的安装目录...target\config 下，在 Tornado 集成环境下生成 Bootloader，然后就可以进行一般的嵌入式操作系统开发过程了。在这里，对 BSP 的修改只有 config.h 和 makefile 文件的 RAM/ROM 地址定义，修改如下：

```
#define ROM_BASE_ADRS      0xff800000
#define ROM_TEXT_ADRS      (ROM_BASE_ADRS)
#define ROM_WARM_ADRS      (ROM_TEXT_ADRS+8)
#define ROM_SIZE            0x00400000
#define RAM_HIGH_ADRS      0x00200000
#define RAM_LOW_ADRS       0x00100000
```

TCP/IP 通信实验

本文建立的实验环境包括：一块 Virtex-II Pro 开发板 DS-BD-2VP20-FF1152(客户端)，一台 Pentium4 计算机主机(服务器端)，一根 RS232 串口线、网线，parallel IV 下载线。基于 TCP/IP 的 Socket 应用框架原理如图 4 所示，经过实际测试得知，网络吞吐效率可超过 30%。

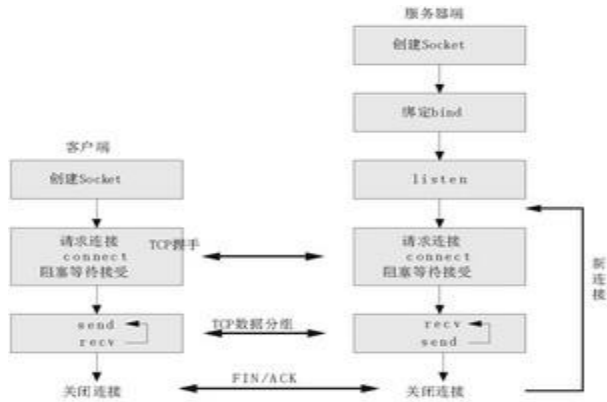


图4 面向连接的 Socket 应用

客户端程序：

```
sFd=socket (AF_INET, SOCK_STREAM, 0);
optval = 60000;
setsockopt (sFd, SOL_SOCKET, SO_SNDBUF, (char *)&optval, sizeof (optval));
sockAddrSize=sizeof (struct sockaddr_in);
bzero((char *)&serverAddr , sockAddrSize);
serverAddr.sin_family=AF_INET;    serverAddr.sin_len=(u_char) sockAddrSize;
serverAddr.sin_port=htons (SERVER_PORT_NUM);
serverAddr.sin_addr.s_addr = inet_addr ("128.0.43.102");
connect(sFd, (struct sockaddr *)&serverAddr, sockAddrSize);
send (sFd , (char *)&myrequest. message, REQUEST_MSG_SIZE, 0);
服务器端的程序可以由 VC 类库来实现，这里就不再给出。
```

结语

开发基于 FPGA 的嵌入式系统是迈向最终 SoC 的必由之路，传统的 FPGA 厂商纷纷涉足嵌入式领域，FPGA 所具有的低成本、低功耗、小尺寸、高性能、开发周期短等优势，必将使其在当今嵌入式开发的热潮中具有广阔的应用前景。