

## 基于 MPC8270 的 VxWorks BSP 的移植

张寅生 雷 杰

中国空空导弹研究院 控制所 (河南 洛阳 471000)

**摘 要** 针对嵌入式实时操作系统的要求,文中详细分析了在整个系统结构层次中 VxWorks 所处的位置和所起承上启下的作用,具体介绍了 VxWorks 的主要功能,并深入阐述了 VxWorks 的启动过程。同时,对于不同的目标硬件平台,VxWorks 需要不同的板级支持包 BSP 来实现对其的板级接口,从而实现平台的可移植性。进一步地,详细说明了作为实现操作系统 VxWorks 硬件平台独立性的设备接口的板级支持包 BSP 的具体功能与结构,这样,程序开发者可以在完全不熟悉目标平台具体硬件的情况下,只在 BSP 中相应位置添加配置参数就可以完成底层硬件配置工作,最后讲述了 BSP 的主流设计方法。

**关键词** 嵌入式 实时操作系统 VxWorks 板级支持包

中图分类号 TP31

文献标识码 A

文章编号 1673-629X(2013)04-0096-04

doi 10.3969/j.issn.1673-629X.2013.04.023

### Transplant of VxWorks BSP Based on MPC8270

ZHANG Yin-sheng, LEI Jie

Institute of Control, China Airborne Missile Academy, Luoyang 471000, China

**Abstract** In view of the requirement of the embedded RTOS, analyze in detail VxWorks position in the whole system structure hierarchy and playing essential role, concretely introduce the VxWorks main function, and discuss the VxWorks start-up process in depth. Simultaneously, for different target hardware platform, VxWorks need different BSP to realize its board level interface, so as to realize the platform transplantability. Further give a detailed description of the function and structure of BSP which plays a role of the realization of the hardware platform of VxWorks. So, under the case which program developers can not familiar with the target platform specific hardware, only in the corresponding position of BSP adding configuration parameters can finish bottom hardware configuration, finally tell the main-stream of BSP design method.

**Key words** embedded RTOS VxWorks BSP

#### 0 引言

20 世纪 80 年代,美国风河公司推出了一款全新的具有强实时性的 RTOS 实时操作系统 VxWorks。自问世以来,受到越来越多的欢迎,其简洁高性能模块化的内核机制、非常友善的开发应用环境等特点深受开发者满意,并且产品不断升级换代,在 RTOS 领域所受欢迎程度与日俱增,美国 NASA 的火星探测器就是一个非常成功的应用实例<sup>[1,2]</sup>。

8270 是嵌入式 PowerPC 的 80 系列的又一款产品,其 G2\_LE 内核在原内核基础上有了进一步提高,该处理器芯片以其极高的性能指标能够胜任在成本、大小、容量、功耗等有严格要求的环境。MPC8270 就

是这样一款使用该处理器的开发板,它通过非常高的集成度来减小整个系统的空间要求,从而带来大大方便了 PCB 设计、提高了调试工作的效率等优点,有了这样高集成度微处理器,就可以充分使用 PCI 接口进行嵌入式系统的设计,所应用的领域涉及广泛,比如路由器、接线器、网络存储应用和图像显示系统等。特别地,HHPPC8270-3FEC-PCI-R1 提供了 3 个高速的百兆以太网接口,可以满足网口之间快速的转发,同时提供的 2 个 PCI 插槽可以方便地扩展多种 PCI 接口卡,实现包括硬盘存储、无线通讯、图像采集/显示等功能需求。适合于 SOHO 网关、无线 AP、监控系统、嵌入式防火墙/路由器、Web 服务器等产品应用<sup>[3,4]</sup>。

对于上层操作系统 VxWorks 和应用,通过构建一层独立的底层软件支持层——板级支持包(BSP)来实现与底层硬件的隔离及接口,该板级支持包向下面向硬件平台,具体为配置相应寄存器,向上则是屏蔽不同硬件平台各种的差别,提供给 VxWorks 统一的驱动接口。于是,板级支持包包括两个层次:初始化部分和

收稿日期 2012-06-16 修回日期 2012-09-23

基金项目 航空科学基金(2009EA12001)

作者简介 张寅生(1986-)男,江苏淮安人,助理工程师,主要研究方向为弹载计算机嵌入式操作系统;雷杰,高级工程师,主要研究方向为弹载计算机嵌入式操作系统。

驱动部分。图 1 是 BSP 在系统中的层次结构<sup>[5, 6]</sup>。

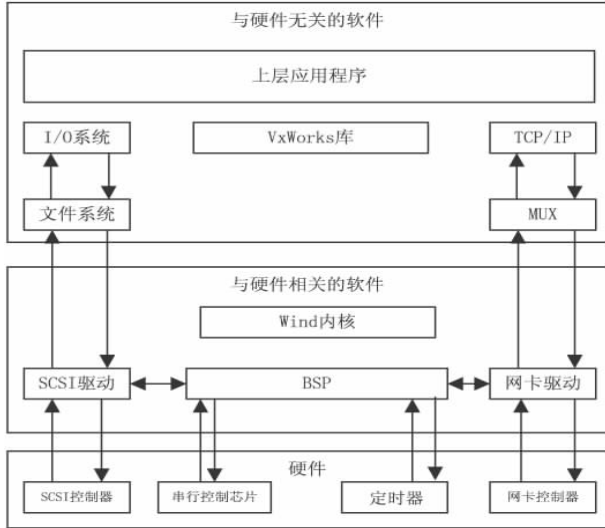


图 1 BSP 在系统中的层次

### 1 VxWorks 的启动流程

系统上电后，G2\_LE 内核首先根据硬件配置字 HCWR 完成最初内核状态的初始化，然后 PC 跳到入口地址 romInit.s 中的初始化函数 romInit，完成中断屏蔽、MMU 的最小初始化、堆栈的初始化等处理器级的最小初始化工作，这些完成后 PC 跳到 bootInit.c 中的内存搬移函数 ramStart，该函数根据内核类型，是否为 rom 驻留型，是否为压缩型等，将 boot 或内核从 rom 中搬移到 ram 中去，并将相应的内存初始化，接下来执行 sysALib.c 的系统初始化函数 sysInit，该函数进行缓冲禁用、中断锁存、初始化相应寄存器到相应值等，最后调用 usrConfig.c 中的用户初始化函数 usrInit，用户初始化主要包括异常向量表初始化、系统硬件初始化、

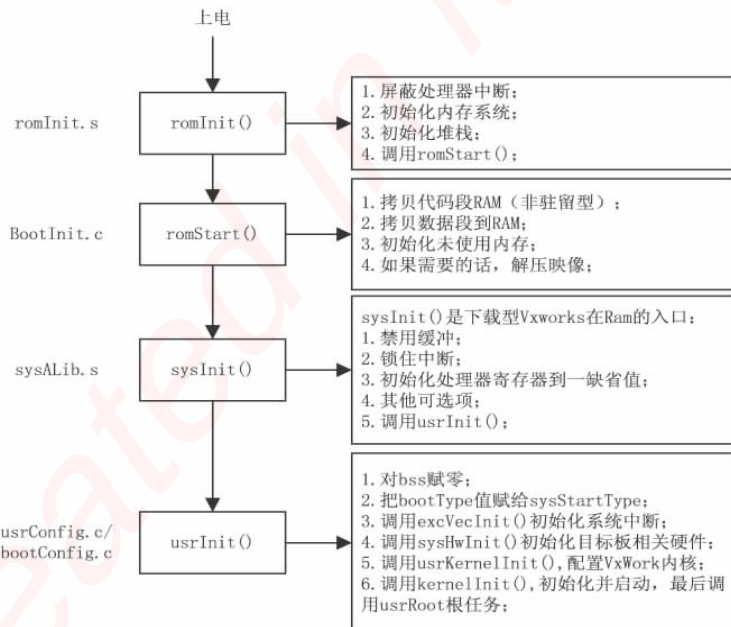


图 2 VxWorks 启动流程

内核机制配置初始化等，最后产生根任务。VxWorks 详细启动流程如图 2。

### 2 BSP 在 MPC8270 上的移植

文中以某飞控系统中基于 MPC8270 的计算机系统为实例，介绍 BSP 的移植开发过程，以及在开发过程中的注意点。图 3 为该计算机系统结构图<sup>[7, 8]</sup>：

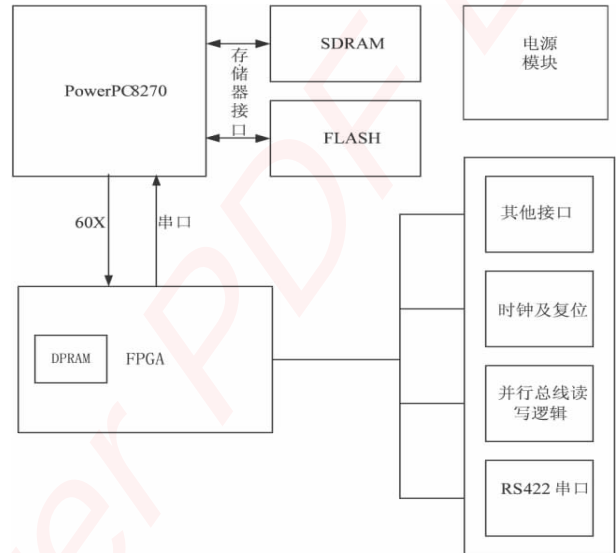


图 3 计算机系统结构图

在这个框图中可以看到 PPC 是一个最小系统的设计思想，该计算机系统用到的外部接口只有 60Xbus、串口和中断，这样系统硬件结构显得很简洁，该系统使用一路 SMC 作为调试串口，两路 SCC 作为通信使用<sup>[9, 10]</sup>。

#### 2.1 根据 CPU 确定 BSP 模板

由于飞思卡尔 ADS 板的 BSP ads828x 也是针对 MPC8270 的，所以笔者在开发过程中选择此 BSP 来进行修改。

VxWorks 的 BSP 组成文件主要包含在 4 个目录下<sup>[11]</sup>：

- 1( 安装目录 \target \config \all 各个板级支持包的共享部分) ；
- 2( 安装目录 \target \config \comps \vxworks 包含.cdf 格式的描述文件，用于模块配置时使用) ；
- 3( 安装目录 \target \config \comps \src 系统选择的模块源码部分) ；
- 4( 安装目录 \target \config \bsp-name 各个具体目标开发板私有部分)。

通过复制模板 ads828x 板级支持包到“target \config \”下，重新命名，再在此基础上进行适当修改就可以完成自己 BSP 的设计。

## 2.2 makefile 文件修改

Makefile 文件控制映像的创建,在 makefile 文件中使用了将近 135 个宏,最简单的 makefile 文件包含以下宏:

- a CPU 处理器型号;
- b TOOL 编译工具比如 Diab 等;
- c TGT\_DIR 默认目标目录;
- d TARGET\_DIR BSP 目录名;
- e VENDER 制造商名称;
- f BOARD 目标板的名称;
- g ROM\_TEXT\_ADRS 启动代码的入口地址;
- h ROM\_SIZE ROM 中代码的大小;
- i RAM\_LOW\_ADRS 加载 VxWorks 到 RAM 中的地址;
- j RAM\_HIGH\_ADRS bootROM 拷贝到 RAM 的目标地址。

根据计算机系统的 Flash 的大小及基地址和 SDRAM 的大小及基地址来修改相应的 makefile 宏,并注意与 config.h 中的配置保持一致。

## 2.3 config.h 文件修改

该文件包括所有涉及 CPU 主板的配置及定义,首先完成的工作是配置系统时钟,因为系统时钟是操作系统运行的基础,较之 ADS 板,并参考计算机系统的串口波特率要求,修改 BRG 的分频,系统默认的调试方式是通过网口来通信,而我们要求通过串口来进行调试,故需修改 DEFAULT\_BOOT\_LINE 为串口方式,同时配置调试串口的波特率、调试类型和设备选择等;对于不需要的文件系统、PCI 功能和网口部分等进行屏蔽,这样可以增加系统的稳定性、可靠性和缩减映像的大小,对应系统空间映射,根据计算机系统 Flash、SDRAM 和 FPGA 的要求以及实际硬件的条件来进行修改分配。

例如,以下为调试串口配置代码:

```
/* Number of TTY definition */
#undef NUM_TTY
#define NUM_TTY N_SIO_CHANNELS
#undef CONSOLE_TTY
#define CONSOLE_TTY NONE
#undef WDB_TTY_CHANNEL
#define WDB_TTY_CHANNEL 0
#undef WDB_COMM_TYPE
#define WDB_COMM_TYPE WDB_COMM_SERIAL
#undef CONSOLE_BAUD_RATE
#define CONSOLE_BAUD_RATE 115200
#undef WDB_TTY_BAUD
#define WDB_TTY_BAUD 115200
#define INCLUDE_TSFS_BOOT
```

```
#undef WDB_TTY_DEV_NAME
#define WDB_TTY_DEV_NAME
#define INCLUDE_SHELL
```

## 2.4 romInit.s 文件

该文件主要负责 bootrom 代码和内核搬移前必要的初始化工作,为内核搬移做准备,主要执行 romInit 函数,其功能主要是对 CPU 进行最小的初始化,其必须包括屏蔽处理器中断、初始化内存系统以及初始化堆栈指针等。较之 ADS 板,计算机系统地址分配包括 Flash、SDRAM 和 FPGA 三个部分,共占用 MMU 的 CS0、CS2、CS3 和 CS4 共 4 个块,同时修改其相关配置寄存器,诸如 OR0/BR0 等,同时根据计算机系统两路通信串口的要求,修改 SCCR 的配置,修改 BCR 配置寄存器,将系统工作方式由兼容模式切换到单机模式;堆栈初始化工作和流程是标准的,无需修改。

例如,以下为 FPA 接口配置部分代码:

```
#ifdef FPGA_8270
/* cs 3 space 16bit */
lis r5, HIADJ (CFG_BR3_PRELIM )
addi r5, r5, LO (CFG_BR3_PRELIM )
lis r6, HIADJ M8270_BR3 INTERNAL_MEM_MAP
_MAP_ADDR ) )
addi r6, r6, LO M8270_BR3 INTERNAL_MEM
_MAP_ADDR ) )
stw r5, 0 r6 )

lis r5, HIADJ (CFG_OR3_PRELIM )
addi r5, r5, LO (CFG_OR3_PRELIM )
lis r6, HIADJ M8270_OR3 INTERNAL_MEM
_MAP_ADDR ) )
Addi r6, r6, LO M8270_OR3 INTERNAL_MEM
_MAP_ADDR ) )
stw r5, 0 r6 )
```

## 2.5 bootInit.c 文件、bootConfig.c 文件和 usrConfig.c 文件

bootInit.c 主要是完成 bootrom 的搬移工作,把固化在 Flash 的代码搬移到 SDRAM 中去执行。最后再跳转到 bootConfig.c,如果是非下载型,则是 usrConfig.c 的 usrInit 函数。bootConfig.c 完成 bootrom image 的初始化和控制,包括对中断向量设置、系统硬件的初始化、配置操作系统的内核、开 cache 功能、调用 usrRoot 根任务、初始化内存池和系统时钟、对 IO 文件系统初始化和串口驱动进行安装、初始化 WDB 代理、创建 bootcmdloop 任务和初始化 bootline 参数、下载操作系统的映像文件到内部的 RAM 并且运行操作系统,如果是非下载型映像,则直接跳到 usrConfig.c 的 usrInit 函数,主要完成 VxWorks image 的初始化代码。包括对中断向量设置、系统硬件的初始化、配置操作的内

核、开 cache 功能、调用 `usrRoot` 根任务、初始化内存池和系统时钟、对 IO 文件系统初始化和串口驱动进行安装、初始化 WDB 代理、调用用户的应用程序<sup>[12]</sup>。

这三个文件的流程为标准流程,无需修改。

## 2.6 sysLib. c 文件

该文件主要包含一些系统相关的函数例程,由于本系统没有用来自带的接口所以不需要进行修改,主要关心的是其完成的 MMU 的初始化表描述,由于计算机系统的 CS3 和 CS4 空间挂载在 FPGA 中,所以要在内存映射表中描述,这样系统才能正常地访问外部 CS3 和 CS4 空间,部分代码如下:

```
void * FPGA_BASE1_ADDR ,
void * FPGA_BASE1_ADDR ,
    FPGA_8270_SIZE ,
VM_STATE_MASK_VALID |
VM_STATE_MASK_WRITABLE |
VM_STATE_MASK_CACHEABLE |
VM_STATE_MASK_GUARDED ,
VM_STATE_VALID |
VM_STATE_WRITABLE |
VM_STATE_CACHEABLE_NOT |
VM_STATE_GUARDED
```

## 2.7 sysSerial. c 文件

该文件为可选文件,用于所有的串口设置和初始化,ADS 板默认驱动是两路的 SCC 驱动,计算机系统使用的是一路 SMC 和两路 SCC 的驱动,修改代码如下:

```
定义一路 SMC 通道
LOCAL M8260_SMC_CHAN m8260SmcChan1 ;
在串口中断初始化的函数中增加
/* connect serial interrupts */
void IntConnect (I2M_TO_IVEC I2M_SMC1 ) ,
VOIDFUNCPTR Int8260SmcInt ,
int *m8260SmcChan1 ) ;
在串口通道初始化的函数中增加
if (channel == 0 )
return (SIO_CHAN * m8260SmcChan1 )
```

## 3 测试验证

编译生成启动代码 `boot` 和 `VxWorks` 后,通过 `codewarrior` 或其他方式将代码烧写到 FLASH 指定位置后,上电启动,因为系统支持打印串口输出信息,可以看见系统启动信息,进一步,可以通过串口或网口方式加载 `target server` 上的 `VxWorks`,如果不能成功,需

进一步修改板级支持包。

比如,在串口初始化之后,看用作调试的 SMC 串口是否可以打开主机端的虚拟窗口,是否能打印相关的信息,通过命令来控制是否正常下载操作系统的映射,如果都能够完成上面的功能,则说明 SMC 串口是正常的。两路 SCC 功能的测试是通过在操作系统跑起来后,通过与主机之间进行大量的数据收发来完成功能的测试。

## 4 结束语

针对嵌入式实时操作系统的要求,文中具体介绍了 `VxWorks` 的主要功能,并深入阐述了 `VxWorks` 的启动过程,进一步地,详细说明了具体功能与结构,最后讲述了 BSP 的主流设计方法,说明程序开发者可以在完全不熟悉目标平台具体硬件的情况下,只在 BSP 中相应位置添加配置参数就可以完成底层硬件配置工作。

参考文献:

- [1] 武华,刘军伟.基于 `VxWorks` 的多任务程序设计[J].计算机技术与发展,2011,21(9):163-166.
- [2] 周雪峰.实时操作系统 `VxWorks` 下驱动程序的设计[J].现代雷达,2007,29(1):44-45.
- [3] `VxWorks Programmer's Guide (Version 5.4 Edition 1)` [M]. Atlantic Wind river system, 1999.
- [4] Windriver system. Tornado BSP Training Workshop [M]. Atlantic Wind river system, 1998.
- [5] Windriver system. Tornado User' Guide (Windows Version 2.2) [M]. Atlantic Wind river system, 2002.
- [6] 黄启军,李晓光.基于 S3C44B0X 微处理器的 `VxWorks` BSP 开发与实现[J].计算机技术与发展,2006,16(7):45-48.
- [7] 李文成.基于 `VxWorks` 多任务信控系统的软件设计与实现[J].计算机应用与软件,2012,29(4):86-88.
- [8] 刘华伟,黄国荣. `VxWorks` 在某导航计算机设计中的应用[J].计算机应用与软件,2009,26(9):256-257.
- [9] 戴娅,何亦征.基于 `VxWorks` 多任务的无线接收机系统设计与实现[J].计算机应用与软件,2007,24(9):159-161.
- [10] 洪洋,白岚. `VxWorks` 实时操作系统的 BSP 开发[J].光电技术应用,2003,3(2):59-63.
- [11] 王彦刚,吕遵明,万留进. `VxWorks` 系统的 BSP 设计与实现[J].计算机系统应用,2011,20(10):91-94.
- [12] 孟英谦,王玲.嵌入式 `VxWorks` 实时操作系统中串口通信的实现[J].兵工自动化,2003,20(1):52-54.

# 嵌入式资源免费下载

## 总线协议:

1. [基于 PCIe 驱动程序的数据传输卡 DMA 传输](#)
2. [基于 PCIe 总线协议的设备驱动开发](#)
3. [CANopen 协议介绍](#)
4. [基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计](#)
5. [FPGA 实现 PCIe 总线 DMA 设计](#)
6. [PCI Express 协议实现与验证](#)
7. [VPX 总线技术及其实现](#)
8. [基于 Xilinx FPGA 的 PCIE 接口实现](#)
9. [基于 PCI 总线的 GPS 授时卡设计](#)
10. [基于 CPCI 标准的 6U 信号处理平台的设计](#)

## VxWorks:

1. [基于 VxWorks 的多任务程序设计](#)
2. [基于 VxWorks 的数据采集存储装置设计](#)
3. [Flash 文件系统分析及其在 VxWorks 中的实现](#)
4. [VxWorks 多任务编程中的异常研究](#)
5. [VxWorks 应用技巧两例](#)
6. [一种基于 VxWorks 的飞行仿真实时管理系统](#)
7. [在 VxWorks 系统中使用 TrueType 字库](#)
8. [基于 FreeType 的 VxWorks 中文显示方案](#)
9. [基于 Tilcon 的 VxWorks 简单动画开发](#)
10. [基于 Tilcon 的某武器显控系统界面设计](#)
11. [基于 Tilcon 的综合导航信息处理装置界面设计](#)
12. [VxWorks 的内存配置和管理](#)
13. [基于 VxWorks 系统的 PCI 配置与应用](#)

## Linux:

1. [Linux 程序设计第三版及源代码](#)
2. [NAND FLASH 文件系统的设计与实现](#)
3. [多通道串行通信设备的 Linux 驱动程序实现](#)
4. [Zsh 开发指南-数组](#)
5. [常用 GDB 命令中文速览](#)
6. [嵌入式 C 进阶之道](#)

## Windows CE:

1. [Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计](#)
2. [Windows CE 的 CAN 总线驱动程序设计](#)
3. [基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究](#)
4. [基于 Windows CE.NET 平台的串行通信实现](#)
5. [基于 Windows CE.NET 下的 GPRS 模块的研究与开发](#)
6. [win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码](#)
7. [Windows 下的 USB 设备驱动程序开发](#)
8. [WinCE 的大容量程控数据传输解决方案设计](#)
9. [WinCE6.0 安装开发详解](#)
10. [DOS 下仿 Windows 的自带计算器程序 C 源码](#)
11. [G726 局域网语音通话程序和源代码](#)
12. [WinCE 主板加载第三方驱动程序的方法](#)
13. [WinCE 下的注册表编辑程序和源代码](#)
14. [WinCE 串口通信源代码](#)
15. [WINCE 的 SD 卡程序\[可实现读写的源码\]](#)

## PowerPC:

1. [Freescale MPC8536 开发板原理图](#)
2. [基于 MPC8548E 的固件设计](#)
3. [基于 MPC8548E 的嵌入式数据处理系统设计](#)
4. [基于 PowerPC 嵌入式网络通信平台的实现](#)
5. [PowerPC 在车辆显控系统中的应用](#)
6. [基于 PowerPC 的单板计算机的设计](#)

## ARM:

1. [基于 DiskOnChip 2000 的驱动程序设计及应用](#)
2. [基于 ARM 体系的 PC-104 总线设计](#)
3. [基于 ARM 的嵌入式系统中断处理机制研究](#)
4. [设计 ARM 的中断处理](#)
5. [基于 ARM 的数据采集系统并行总线的驱动设计](#)
6. [S3C2410 下的 TFT LCD 驱动源码](#)
7. [STM32 SD 卡移植 FATFS 文件系统源码](#)
8. [STM32 ADC 多通道源码](#)
9. [ARM Linux 在 EP7312 上的移植](#)

## Hardware:

1. [DSP 电源的典型设计](#)
2. [高频脉冲电源设计](#)
3. [电源的综合保护设计](#)
4. [任意波形电源的设计](#)
5. [高速 PCB 信号完整性分析及应用](#)
6. [DM642 高速图像采集系统的电磁干扰设计](#)