Davinci 平台下大页面 NAND Flash 上的系统构建问题

唐建兵,吴仲光,肖 炀
(四川大学 计算机学院,四川 成都 610065)

摘 要:以 TI 公司新一代 Davinci TMS320DM6446 嵌入式处理器为例,详细描述了在 NAND Flash 上构建嵌入式 Linux 操作系统的一般方法。结合 Sumsang 公司的 K9K8G08U0A 大页面 NAND Flash,给出了引导程序的移植、烧写的具体设计及实现过程,提供基于大页面 NAND Flash 作为唯一外部存储设备的稳定、可靠的解决方案 。为新一代 Davinci 系列处理器的配置、使用提供了参考。

关键词:Davinci;NAND Flash;大页面;引导程序;Linux
中图分类号:TP368.33 **文献标识码**:A **文章编号**:1674-7720(2011)02-0103-03

Key issues of system construction on large page NAND Flash based on Davinci

Tang Jianbing, Wu Zhongguang, Xiao Yang
(Department of Computer, Sichuan University, Chengdu 610065, China)

Abstract: Take TI's next-generation Davinci TMS320DM6446 embedded processor as an example, the general method of build embedded Linux operating system on NAND Flash is described in detail. Combined with Samsung's K9K8G08U0A large page NAND Flash, the specific design and implementation process of boot transplant and programming is given, stable and reliable solution based on large page NAND Flash as only external storage device is given. It provide reference for configuration and use of the new generation processors of Davinci family.

Key words: Davinci; NAND Flash; large page; Boot Loader; Linux

达芬奇(Davinci)系列嵌入式处理器是 TI 公司的具有高速处理能力的新一代嵌入式设备[1],它同时具备了 DSP 和精简指令级计算机技术的优点。它集成了一个高性能的 DSP 核心与一个 ARM9 内核,被广泛应用于嵌入式图片、视频处理中[2]。在 Davinci 平台下,通常以 ARM 核为基础构建嵌入式操作系统,但是目前经常采用一片 NOR Flash 加上一片 NAND Flash 作为外部存储设备,并且通常都是 256 B/页或者 512 B/页的小页面 NAND Flash,本文旨在只以一片 2 KB/页的大页面 NAND Flash(Samsung K9K8G08U0A)作为外部存储设备、Davinci TMS320DM6446 作为处理器的硬件结构下,阐述构建稳定可靠的系统需要解决的问题。

1 问题概述

EMIF 是用来连接 Flash、SRAM 等多种存储设备的外设端口。TMS320DM6446 的 EMIF 端口支持每路 32 MB 总共 4 路可寻址的片选空间,支持 8 bit 以及 16 bit 的数据总线宽度,具有可编程的建立、选通以及保持时间,还具备 NAND Flash ECC 校验数据生成功能等[5],因此可以方便灵活地与外部 NAND Flash 芯片通信。在本文的硬件系统中,即采用 TMS320DM6446 的 EMIF 的 CS2 空间与 Samsung K9K8G08U0A NAND Flash 相连。

由于本系统只有一片 NAND Flash 作为外部的存储设备,因此所有的引导程序、操作系统内核以及根文件系统均需要存储在这上面,系统也就需要从 NAND Flash 启动。TMS320DM6446 具有多种启动方式,具体由哪种方式启动,由系统复位时引脚 BTSEL[1:0]电平决定,当 BTSEL[1:0]被置为"01"时,TMS320DM6446 的 ARM 核从 EMIFA 的 EM_CS2 存储空间开始执行(地址为 0x0200 0000),这种情况下 EMIF 连接的是具有线性地址的非易失存储器,通常是 NOR Flash。当 BTSEL[1:0]不为"01"时,TMS-

320DM6446 内部的 ROM BOOT LOADER（简称 RBL）开始运行，RBL 再根据 BTSEL[1:0]的不同值决定从何处加载用户的引导程序 UBL（USER BOOT LOADER）。当 BTSEL[1:0]为“00”时，RBL 将从连接到 EMIF 的 CS2 空间的 NAND Flash 中加载 UBL。由于 RBL 的加载过程是将 UBL 拷贝到 ARM 的内部 RAM 中，因此对于 UBL 的大小限制在 14 KB 以内，但是在嵌入式环境常用的 U-BOOT、ViVi 等的大小都远超过这个限制，因此需要多级加载，一级引导程序主要做系统的初始化，然后将二级引导程序（在本系统中采用 U-BOOT，本文后面提到的 UBL 均指一级引导程序）从 NAND Flash 中读取到 RAM 中，然后启动它，由 U-BOOT 负责操作系统的引导[4]。于是整个 NAND Flash 上系统构建的关键问题包括如何移植 UBL，以使其能够正常初始化系统，正常加载二级引导程序 U-BOOT 到 RAM 中，U-BOOT 的移植使其满足大页面 NAND Flash 的读写要求以及裸机时引导程序的烧写。

2 UBL 移植

UBL 为 TI 公司提供的对于 Davinci 系列处理器通过内部的 ROM BOOT LOADER 启动时的一级引导程序。其工作流程如图 1 所示。

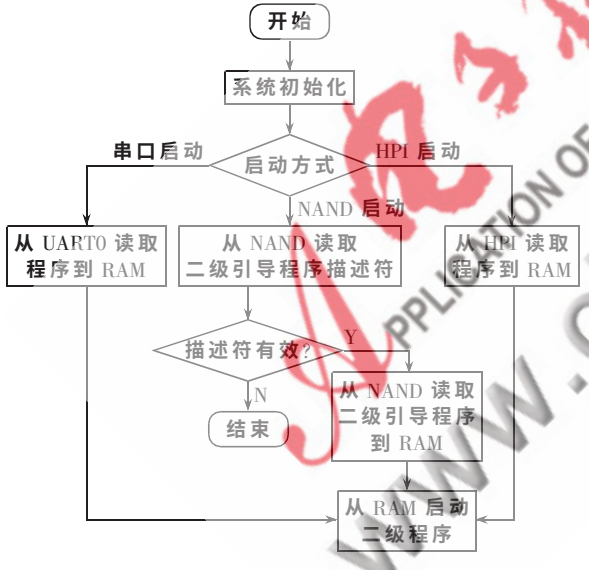

图 1 UBL 流程图

UBL 的移植主要针对本系统中硬件板的结构修改系统初始化过程以及增加对 Samsung K9K8G08U0A NAND Flash 的支持，以下分别阐述。

2.1 系统初始化

2.1.1 设置 CPU、DDR 工作频率

TMS320DM6446 具有两路 PLL，其中 PLL1 通过分频供系统的主时钟及大部分外设的时钟，PLL2 供 DDR2 使用。DSP 时钟频率为 $SYSCLK1=27\ \text{MHz}\times(PLL1\_PLLM+1)$，使用固定一分频，本系统中 DSP 工作在正常频率 594 MHz，因此需设置 $PLL1\_PLLM=21$，即设置寄存器 0x1C4 0910 为 21。

本系统使用两片 K4T1G164QQ-HCE6 DDR2 SDRAM 作为系统内存，该芯片为 DDR2 667 芯片，时钟频率为 333 MHz。TMS320DM6446 中 DDR2 使用 PLL2 的 PLLDIV2 分频作为时钟频率，计算公式为 $(27\ \text{MHz}\times(PLL2\_PLLM+1))/(PLL2\_PLLDIV2\rightarrow RATIO+1)$。因此设置 $PLL2\_PLLM=23$，$PLL2\_PLLDIV2\rightarrow RATIO=1$，PLL2_PLLDIV2 的第 15 位为分频允许位，应置为 1，所以 PLL2_PLLDIV2 为 0x8001，即寄存器 0x1C4 0D10=23，0x1C4 0D1C=0x8001。

2.1.2 配置 EMIF 接口

根据 Samsung K9K8G08U0A NAND Flash 的读、写时序要求，TMS320DM6446 的 EMIF 用于与 NAND Flash 连接时，配置寄存器各字段值需满足如下要求：

$R_{SETUP}\geqslant tCLR(m)/tCYC-1=0$

$RSTORBE\geqslant \max((tREA(m)+tSU)/tCYC, tRP(m)/tcyc)-1=1.5$

$R\_SETUP+R\_STROBE\geqslant (tCEA(m)+tSU)/tcyc-1=2$

$R\_HOLD\geqslant (tH-tCHZ(m))/tcyc-1=-4$

$R\_SETUP+R\_STROBE+R\_HOLD\geqslant tRC(m)/tcyc-3=-0.5$

$TA\geqslant \max((tCHZ(m))/tcyc, (tRHZ(m)-(R\_HOLD+1)tcyc)/(tcyc))-1\geqslant 2$

$W\_SETUP\geqslant \max(tCLS(m)/tcyc, tALS(m)/tcyc, (tCS(m)/tcyc)-1=1$

$W\_STROBE\geqslant tWP(m)/tcyc-1=0.2$

$W\_SETUP+W\_STROBE\geqslant tDS(m)/tcyc-1=0.2$

$W\_HOLD\geqslant \max((tCLH(m))/(tcyc), (tALH(m))/tcyc, (tCH(m))/(tcyc), (tDH(m))/(tcyc))-1=-0.5$

$W\_SETUP+W\_STROBE+W\_HOLD\geqslant tWC(m)/(tcyc)-3=-0.5$

其中 tSU 是 EMIF 数据建立时间，取值 5 ns，tH 数据保持时间取 0，EMFI 时钟为系统 6 分频，所以 $tcyc=1/(27\times(21+1)/6)\approx 10$ ns，根据 EMIF 连接 NAND 的取值要求，设置 EMIF CS2 的配置寄存器值为 0x842429c。

2.2 支持 Samsung K9K8G08U0A NAND Flash

UBL 通过数据结构 struct _NAND_DEV_STRUCT_ 来表示一个型号的 NAND Flash，具有 devID、numBlocks、pagesPerBlock、bytesPerPage 几个字段。通过 struct _NAND_DEV_STRUCT_ 类型的数组 gNandDevInfo[]来记录所有支持的 NAND Flash。UBL 在从 NAND Flash 读取数据之前，首先通过读取设备号命令 0x90 得到 NAND Flash 的设备号，然后从数组 gNandDevInfo[]中查找具有相同设备号的记录，从而得到 NAND Flash 的详细信息，以确定 NAND Flash 的读方式。

因此，需要 UBL 支持特定的 NAND Flash，只需要将其信息添加到数组 gNandDevInfo[]中即可。本系统中用到的 Samsung K9K8G08U0A NAND Flash 设备号为 0xD3，具有 8 192 个存储块，每个块具有 64 个页面，每页具有 2 048 B 数据存储区域以及 64 B 的 Spare 区域，在数组 gNandDevInfo[]中添加{0xD3, 8192, 64, 2048+64}即可。

3 U-BOOT 移植

本系统中使用的 U-BOOT 引导程序由 TI 公司提供的

支持 Davinci 平台以及 NAND Flash 启动的 U-BOOT1.1.3 移植而来。

3.1 NAND Flash 读写时序

U-BOOT1.1.3 不支持 2 KB/page 的大页面 Flash，因此移植过程主要是增加 NAND Flash 的读写、擦除。2 KB 页面 NAND Flash 与普通读写擦除最主要的区别在于地址构成不同，本系统中用到的 Samsung K9K8G08U0A NAND Flash 总共存储空间 1 GB=2^{30}，每页大小为 2 KB=2^{11}，因此总的地址长度 30 bit，从 A0~A29，页地址长度为 11 bit，从 A0~A10，本系统采用 8 bit 的地址数据宽度连接 NAND Flash，页地址和块地址需要分不同的地址周期，因此 NAND 的地址需要 5 个周期送出，前两个周期为页地址，依次为地址的 A0~A7、A8~A10，后三个周期为块地址，依次为地址的 A11~A18、A19~A26、A27~A29，页地址和块地址的最后一个周期不足 8 位，不足的高位均为 0。

Samsung K9K8G08U0A 的读过程如下：写 0x00 命令、分 5 个周期写地址、写 0x30 命令、读数据、根据读出的数据生成 ECC 校验数据、生成的 ECC 数据与读出的 ECC 数据比对以确定数据是否有误以及能否校正。

写过程如下：写 0x80 命令、分 5 个地址周期写地址、送出数据(包括 ECC 校验数据)、写 0x10 命令、读取状态直到 busy 信号无效、检查是否出现写错误。

擦除过程如下：写 0x60 命令、分三个地址周期写块地址、写 0xD0 命令、读取状态直到 busy 信号无效、检查是否出现擦除错误。

3.2 YAFFS2 文件系统烧写

YAFFS2 镜像烧写与 U-BOOT 下普通写 NAND Flash 区别在于 spare 区域的数据不需要程序根据数据存储区的数据生成，spare 区域的数据在制作 YAFFS2 镜像时，已经由镜像制作工具生成并写入了镜像文件。因此在 nand 命令的 write 中增加.yaffs2 选项，当使用 nand write.yaffs2 命令时，直接从指定地址中读出 2 048 B/页数据以及数据后紧跟的 64 B 的 spare 区域数据，并将其写入 NAND Flash 中。

U-BOOT 在 Flash 的读写过程中需要检查坏块情况，在开始读写每个块的时候首先检查该块第一页以及第二页的 spare 区域的第一个数据是否为 0xFF，如果不为 0xFF 则当前块为坏块，需要跳过它。

4 烧写程序

在 UBL 以及 U-BOOT 被固化进 NAND Flash 之前，系统处于裸机状态，无法正常引导操作系统，烧写程序的作用是在裸机状态下借助仿真器的作用，将 UBL 以及 U-BOOT 烧写到 NAND Flash 正确位置的。

前面已经提到，本系统采用的 Samsung K9K8G08U0A NAND Flash 具有 8 192 个存储块(block)。这 8 192 个块按照如下分配其使用方式：第 0 块在出厂时确保不是坏块，用作整个 NAND Flash 的坏块信息存储；第 1~3 块存储 UBL；第 4~7 块存储 U-BOOT；第 8 块存储 U-BOOT 环境变量；第 9~40 块存储 Linux 操作系统内核；第 41~8 191 块存储 YAFFS2 文件系统。

实际中，UBL 和 U-BOOT 都只需要占用一个存储块的存储空间，由于考虑到 NAND Flash 可能有坏块的存在，于是在设计烧写程序时，为 UBL 增加了 2 个冗余块，为 U-BOOT 增加了 3 个冗余块，以确保系统稳定可靠地从 NAND Flash 上启动。

烧写程序是系统在没有任何程序的裸机情况下执行的，烧写程序需要通过仿真器加载到系统目标板的 DDR2 中运行。由于加载程序时系统未执行任何程序，也就没有做任何初始化，DDR2 也处于不可用状态，系统将无法加载程序。TI 的 CCS 集成环境提供了 GEL 文件来解决这一问题，在仿真器连接目标板时会自动执行 GEL 文件中的 OnTargetConnect()函数，在该函数中，需要对系统做初始化。

烧写程序在被加载到内存后，即可被执行来完成 UBL 以及 U-BOOT 的烧写，程序执行过程如图 2 所示。

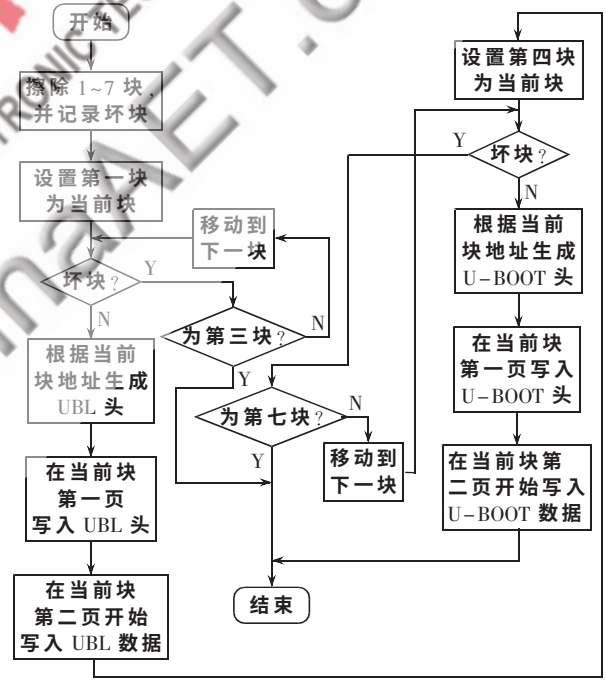

图 2 烧写程序流程图

数据写入时需要注意 UBL 是由 TMS320DM6446 内部的 ROM BOOT LOADER 读入到内部 RAM 中然后执行的，因此，烧写程序对于 ECC 校验数据的生成以及 ECC 数据在 spare 区域的存储位置必须要与 ROM BOOT LOADER 读取数据时的校验方式一致。ROM BOOT LOADER 采用 EMIF 的硬件 ECC 校验，每 512 B 的数据产生 4 B 的校验数据，并按照如下方式存储：spare 区域地址从 0x00 到 0x3F，其中 0x08-0x0B 存储第 1 个 512 B

数据的第3-0位ECC数据，0x18-0x1B存储第2个512 B数据的第3-0位ECC数据，0x28-0x2B存储第3个512 B数据的第3-0位ECC数据，0x38-0x3B存储第4个512 B数据的第3-0位ECC数据。因此，在烧写程序中也使用EMIF硬件ECC校验来生成校验数据，在每次写入数据达到512 B时，通过读寄存器NANDF1ECC（地址为0x0200 0070）来获得ECC值，最后在一页数据写入完毕后写入到spare区域的对应位置。

在UBL以及U_BOOT成功烧写到NAND Flash后，系统上电，U_BOOT成功执行，通过U_BOOT将Linux操作系统内核以及YAFFS2文件系统镜像烧写到NAND Flash，设置U_BOOT环境变量，再次引导系统，Linux系统正常启动。本系统中成功实现了从裸机到整个系统的构建，解决了对大页面NAND Flash的不支持，同时考虑了NAND Flash存在的坏块情况，系统在实际使用中运行稳定可靠。

参考文献

[1] TI Corporation.TMS320DM6446 Digital Media System-on-Chip[EB/OL].[2008-03-31].http://www.ti.com/lit/gpn/tms320dm6446.

[2] TI Corporation.TMS320DM644x DMSoC ARM Subsystem Reference Guide[EB/OL].[2009-03-31].http://www.ti.com/litv/pdf/sprue14b.

[3] TI Corporation.TMS320DM644x DMSoC Asynchronous External Memory Interface(EMIF) Reference Guide[EB/OL].[2009-02-24].http://www.ti.com/litv/pdf/sprue20c.

[4] 王化福，孙同景.从NAND Flash启动嵌入式操作系统[J].可编程控制器与工厂自动化，2009(5)：79-80.

[5] Samsung Corporation. K9XXG08UXA Flash Memory[S], 2006.

（收稿日期：2010-08-19）

作者简介：

唐建兵，男，1986年生，在读硕士研究生，主要研究方向：嵌入式系统、计算机网络与通信。

吴仲光，男，1953年生，副教授，硕士，主要研究方向：光机电一体化、嵌入式系统架构、计算机网络与通信。

肖炀，男，1982年生，硕士，主要研究方向：电路系统。