



(12) 发明专利申请

(10) 申请公布号 CN 101980157 A

(43) 申请公布日 2011. 02. 23

(21) 申请号 201010506082. 9

(22) 申请日 2010. 10. 13

(71) 申请人 中兴通讯股份有限公司  
地址 518057 广东省深圳市南山区科技南路  
55 号

(72) 发明人 董金光

(74) 专利代理机构 北京康信知识产权代理有限  
责任公司 11240  
代理人 余刚 吴孟秋

(51) Int. Cl.  
G06F 9/445(2006. 01)

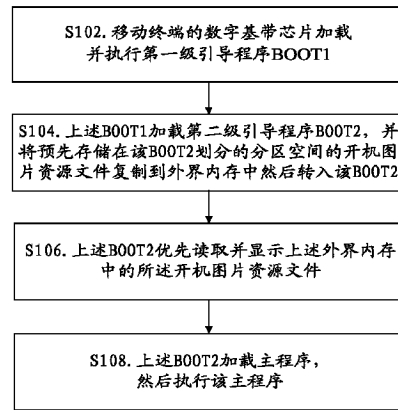
权利要求书 2 页 说明书 6 页 附图 4 页

(54) 发明名称

移动终端开机启动方法及移动终端

(57) 摘要

本发明公开了一种移动终端开机启动方法,包括:移动终端的数字基带芯片加载并执行第一级引导程序 BOOT1 ;BOOT1 加载第二级引导程序 BOOT2,并将预先存储在该BOOT2划分的分区空间的开机图片资源文件复制到外界内存中,然后转入BOOT2 ;BOOT2 优先读取并显示上述外界内存中的上述开机图片资源文件 ;BOOT2 加载主程序,然后执行该主程序。本发明还公开一种移动终端。通过本发明加快了从用户按下开机键后的开机图片显示,加快了人机交互的显示和操作从而有效缩短开机启动时间,增强了用户体验效果。



1. 一种移动终端开机启动方法,其特征在于,包括:  
所述移动终端的数字基带芯片加载并执行第一级引导程序 BOOT1 ;  
所述 BOOT1 加载第二级引导程序 BOOT2,并将预先存储在所述 BOOT2 划分的分区空间的开机图片资源文件复制到外界内存中,然后转入所述 BOOT2 ;  
所述 BOOT2 优先读取并显示所述外界内存中的所述开机图片资源文件 ;  
所述 BOOT2 加载主程序,然后执行所述主程序。
2. 根据权利要求 1 所述的方法,其特征在于,所述 BOOT2 加载主程序然后执行所述主程序,包括:  
预先将所述主程序分别编译链接为多个独立的文件,其中,所述多个独立的文件之一为包括操作系统内核的操作系统映像文件 ThreadXOS. bin ;  
所述 BOOT2 加载所述 ThreadXOS. bin 并开始执行 ;  
所述 ThreadXOS. bin 根据需要加载所述多个独立的文件中的部分或全部文件。
3. 根据权利要求 2 所述的方法,其特征在于,所述多个独立的文件还包括:应用程序文件 App. bin 和资源文件 Resouce. bin。
4. 根据权利要求 3 所述的方法,其特征在于,所述 ThreadXOS. bin 根据需要加载所述多个独立的文件中的部分或全部文件包括:  
所述 ThreadXOS. bin 根据预置优先级加载并执行所述 App. bin 中的一个或多个应用程序 ;  
所述 ThreadXOS. bin 加载并运行需要用到的所述 Resouce. bin 中的部分或全部资源。
5. 根据权利要求 1 至 4 中任一项所述的方法,其特征在于,所述第一级引导程序采用跳除坏块的坏块管理方案。
6. 一种移动终端,其特征在于,包括:  
数字基带芯片,用于加载并执行第一级引导程序 BOOT1 ;  
第一加载模块,用于实现所述 BOOT1 加载第二级引导程序 BOOT2,并将预先存储在所述 BOOT2 划分的分区空间的开机图片资源文件复制到外界内存中,然后转入所述 BOOT2 ;  
图片文件显示模块,用于实现所述 BOOT2 优先读取并显示所述外界内存中的所述开机图片资源文件 ;  
第二加载模块,用于实现所述 BOOT2 加载主程序,然后执行所述主程序。
7. 根据权利要求 6 所述的移动终端,其特征在于,所述第二加载模块包括:  
编译子模块,用于预先将所述主程序分别编译链接为多个独立的文件,其中,所述多个独立的文件之一为包括操作系统内核的操作系统映像文件 ThreadXOS. bin ;  
执行子模块,用于实现所述 BOOT2 加载所述 ThreadXOS. bin 并开始执行,以及实现所述 ThreadXOS. bin 根据需要加载所述多个独立的文件中的部分或全部文件。
8. 根据权利要求 7 所述的移动终端,其特征在于,所述多个独立的文件还包括以下至少之一:应用程序文件 App. bin、资源文件 Resouce. bin。
9. 根据权利要求 7 或 8 所述的移动终端,其特征在于,  
所述执行子模块还用于实现所述 ThreadXOS. bin 根据预置优先级加载并执行所述 App. bin 中的一个或多个应用程序以及实现所述 ThreadXOS. bin 加载并运行需要用到的所述 Resouce. bin 中的部分或全部资源。

10. 根据权利要求 6 所述的移动终端,其特征在于,所述 B00T1 采用跳除坏块的坏块管理方案。

## 移动终端开机启动方法及移动终端

### 技术领域

[0001] 本发明涉及通信领域,具体而言,涉及一种移动终端开机方法及移动终端。

### 背景技术

[0002] 目前,移动终端(例如 TD 终端)大多采用的 LeadCore(联芯科技)的 A2000PH 平台,其支持多种类型的存储媒质,主要包括或非闪存(NOR logic FLASH,简称 NOR 或 NOR FLASH)和伪静态随机访问存储器(Pseudostatic Random Access Memory 简称 PSRAM)的 MCP(Multi-Chip)、与非闪存(NAND logic FLASH,以下简称 NAND 或 NAND FLASH)和同步动态随机方位存储器(Synchronous Dynamic Random Access Memory,以下简称 SDRAM)的 MCP。由于 Nand Flash 逐渐成为非易失存储市场的主流,同时考虑成本因素,A2000+H 采用 NAND+SDRAM 的存储方案:NAND 中存放代码、数据、文件系统等,代码搬移到 SDRAM 上运行。

[0003] 在调试过程中,发明人发现,基于 A2000PH 平台的手机启动慢的主要原因是从 NAND FLASH 中加载内核到内存特别慢,几乎占了启动使用的 99%的时间。

[0004] 相关技术中,移动终端一般采用两级 boot(引导)加载的方式,首先由固化在数字基带芯片(Digital BaseBand,简称为 DBB)中的 BOOTROM CODE 程序将第一级 BOOT1 程序加载至 DBB 中的系统内存空间(简称为 RAM),并将控制权交给 BOOT1,交由其执行,然后 BOOT1 开始进行简单的初始化配置 SDRAM(外部内存),并加载第二级 BOOT2 程序。待第二级引导程序加载至 SDRAM 中后,BOOT1 将控制权交给 BOOT2 并开始执行,此时由 BOOT2 加载并显示开机显示图片,同时完成主程序的拷贝,并跳转到主程序开始执行。

[0005] 由上可知,基于 A2000PH 平台的手机在加载完主程序之前,不能显示相应人机交互操作界面,譬如键盘、LCD 触摸以及界面显示,只能在引导程序加载完主程序并将控制权交给主程序之后才能进行手机的人机交互操作,使用户感觉手机的启动时间太长,从实际测试统计来看,终端从用户按下开机键,到显示 Idle 界面大约需要 25 秒钟以上的时间,严重影响手机用户的用户体验,对于初次的用户使用者来说,可能会以为是手机出现开机故障。

### 发明内容

[0006] 针对相关技术中开机启动时间长,用户体验效果较差等问题,本发明提供了一种移动终端开机方法及移动终端,以解决上述问题至少之一。

[0007] 根据本发明的一个方面,本发明提供了一种移动终端开机启动方法,包括:移动终端的数字基带芯片加载并执行第一级引导程序 BOOT1;该 BOOT1 加载第二级引导程序 BOOT2,并将预先存储在上述 BOOT2 划分的分区空间的开机图片资源文件复制到外界内存中,然后转入该 BOOT2;该 BOOT2 优先读取并显示所述外界内存中的开机图片资源文件;BOOT2 加载主程序,然后执行主程序。

[0008] 根据本发明的另一方面,本发明还提供一种移动终端,包括:数字基带芯片,用于加载并执行第一级引导程序 BOOT1;第一加载模块,用于实现上述 BOOT1 加载第二级引导程

序 BOOT2, 并将预先存储在该 BOOT2 划分的分区空间的开机图片资源文件复制到外界内存中, 然后转入上述 BOOT2; 图片文件显示模块, 用于实现该 BOOT2 优先读取并显示所述外界内存中的所述开机图片资源文件; 第二加载模块, 用于实现上述 BOOT2 加载主程序, 然后执行该主程序。

[0009] 本发明通过修改两级引导程序的加载启动方式, 将原先的开机显示图片从主程序空间中移除, 将其放入第二级引导程序的存储空间, 加快了从用户按下开机键后的开机图片显示, 加快了人机交互的显示和操作从而有效缩短开机启动时间, 增强了用户体验效果。

[0010] 本发明的其它特征和优点将在随后的说明书中阐述, 并且, 部分地从说明书中变得显而易见, 或者通过实施本发明而了解。本发明的目的和其他优点可通过在所写的说明书、权利要求书、以及附图中所特别指出的结构来实现和获得。

### 附图说明

[0011] 此处所说明的附图用来提供对本发明的进一步理解, 构成本申请的一部分, 本发明的示意性实施例及其说明用于解释本发明, 并不构成对本发明的不当限定。在附图中:

[0012] 图 1 为根据本发明实施例的移动终端开机方法流程图;

[0013] 图 2 为根据本发明优选实施例的存储空间示意图;

[0014] 图 3 为根据本发明优选实施例的引导程序空间的使用示意图;

[0015] 图 4 为本发明优选实施例的主程序拆分后存储空间示意图;

[0016] 图 5 为根据本发明优选实施例的加载第一级引导程序 (BOOT1) 流程示意图;

[0017] 图 6 为根据本发明优选实施例的加载第二级引导程序 (BOOT2) 的流程示意图;

[0018] 图 7 为根据本发明优选实施例的主程序运行示意图;

[0019] 图 8 为根据本发明实施例一种移动终端的结构框图;

[0020] 图 9 为根据本发明优选实施例移动终端的第二加载模块结构框图;

### 具体实施方式

[0021] 下文中将参考附图并结合实施例来详细说明本发明。需要说明的是, 在不冲突的情况下, 本申请中的实施例及实施例中的特征可以相互组合。

[0022] 目前, 随着 3G 业务的不断出现以及手机功能的增强, 手机的代码量越来越大, 给手机的开机启动时间带来了巨大的挑战,

[0023] 相关技术中, 基于 A2000PH 平台的手机启动慢的主要原因是从 NAND FLASH 中加载内核到内存特别慢, 几乎占了启动使用的 99% 的时间, 而加载完内核后的启动是挺快的。因此, 要缩短启动时间, 可以通过缩小内核来实现。也就是说, 在不删除必要组件的前提下必须减小内核文件体积。本发明实施例即是基于上述设计思想。具体如下:

[0024] 图 1 为根据本发明实施例的一种移动终端开机方法流程图, 如图 1 所示, 该方法包括:

[0025] 步骤 S102, 移动终端的数字基带芯片加载并执行第一级引导程序 (BOOT1);

[0026] 步骤 S104, 上述 BOOT1 加载第二级引导程序 (BOOT2), 并将预先存储在该 BOOT2 划分的分区空间的开机图片资源文件复制到外界内存中然后转入该 BOOT2;

[0027] 优选地, 上述 BOOT1 采用跳除坏块的坏块管理方案。

[0028] 步骤 S106, 上述 BOOT2 优先读取并显示上述外界内存中的上述开机图片资源文件;

[0029] 步骤 S108, 上述 BOOT2 加载主程序, 然后执行该主程序。

[0030] 本发明上述实施例, 将原先的开机显示图片从主程序空间中移除, 将其放入第二级引导程序 BOOT2 的存储空间, 加快了从用户按下开机键后的开机图片显示, 在提供了较好的用户体验的同时, 亦提高了手机的启动速度。

[0031] 优选地, 上述 BOOT2 加载主程序并开始执行, 可以包括以下处理:

[0032] 预先将上述主程序分别编译链接为多个独立的文件, 其中, 多个独立的文件之一为包括操作系统内核的操作系统映像文件 ThreadXOS. bin; 上述 BOOT2 加载上述 ThreadXOS. bin 并开始执行; 上述 ThreadXOS. bin 根据需要加载上述多个独立的文件中的部分或全部文件。通过将主程序分为多个独立的文件, 避免了 BOOT2 一次加载的程序太大, 从而可以提高 BOOT2 加载主程序的速度。

[0033] 优选地, 上述多个独立的文件还包括: 应用程序文件 App. bin 和资源文件 Resouce. bin。

[0034] 相应地, 在上述实施例优选实施过程中, 上述 ThreadXOS. bin 根据预置优先级加载并执行上述 App. bin 中的一个或多个应用程序, 这样可以先加载优先级高的应用程序, 从而提高用户体验; 以及 ThreadXOS. bin 加载并运行需要用到的上述 Resouce. bin 中的部分或全部资源。

[0035] 从上述实施例可以看出, 主程序通过编译、链接及加载技术, 独立编译成三个执行文件, 减少了第二级引导程序加载主程序的时间, 从而达到缩短开机启动时间之目的。

[0036] 本发明移动终端开机启动方法的一个优选实施例如下:

[0037] 如图 2 所示, 本实施例中, 移动终端的存储空间包括: 终端的存储空间分为: DBB 片上内存 SRAM、数据 Cache、指令 Cache 以及片上引导程序 BOOTROM, 外部非易失性存储器 NAND FLASH 和外部主内存 SDRAM。

[0038] 如图 3 所示, 本实施例中引导程序空间的使用如下: NANDFLASH 制造厂商保证第一块的非坏块特性, 故可以可靠的存放第一级引导程序 BOOT1, 第二级引导程序 BOOT2 存放在 Block 0 至 9 连续的两个好块中, 用于开机显示的图片 Start Picture 存放在 BOOT2 之后的连续的两个好块中, 由于前面 10 个物理块的操作在开机使用时, 都是读 FLASH 的操作, 所以采用简单的坏块管理方案 - 跳块处理, 这也免除了由于考虑写操作产生动态坏块而采用逻辑寻址和映射的方式的坏块处理方式。采用这种方式可以较快的加载并显示开机图片, 大约可以从原来的 4 秒钟减少到现在的 1.5 秒钟。具体包括:

[0039] ①, 第一级引导程序 (BOOT1) 一般很小, 只有几十 K 字节的大小, 本实施例中大小为 26Kbytes 左右, 占用 NAND FLASH 的第一块 (即图 2 中 Block0), 此处所述 Block0 块大小为 128K Bytes, 该块厂家保证其为非坏块;

[0040] ②, 第二级引导程序 (BOOT2) 占用手机从第二块 (即图 2 中 Block1) 开始的连续的两个好块;

[0041] ③, 开机显示图片 StartPicture 紧接着上述 BOOT2 开始的连续两个好块。

[0042] 本优选实施例中所述移动终端开机方法具体包括以下处理:

[0043] 首先将要显示的开机显示的图片资源文件 (大约 150K Bytes), 以下简称 Start

Picture, 从主程序分区中挪出, 放在第二级引导程序 BOOT2 划分的分区空间中 ;

[0044] (1) 由数字基带芯片 DBB 中的 BOOTROM CODE 加载并执行第一级 BOOT1 程序, BOOT1 的执行文件大小精简为大约 26KBytes。

[0045] (2) 第一级引导程序 (BOOT1) 加载第二级引导程序 (BOOT2) (其执行文件大小约 250K Bytes) 和复制存放在内存 SDRAM 中的 Start Picture (资源大小约为 150K Bytes), 然后程序跳转至 BOOT2 执行。优选的, 上述 BOOT1 采用跳除坏块的坏块管理方案。

[0046] 其中, 如图 5 所示, 本优选实施例中 BOOT1 模块启动过程包括 :

[0047] 步骤①, 移动终端的 ARM 核初始化以及 SDRAM 初始化, 然后执行步骤② ;

[0048] 步骤②, 加载第二级引导程序 (BOOT2) 并拷贝开机图片至 SDRAM 主内存, 然后执行步骤③ ;

[0049] 步骤③, 跳转至第二级引导程序被加载的起始地址开始执行。

[0050] (3) 第二级引导程序 BOOT2 一开始就从 SDRAM 里面将图片资源读取并进行开机图片的显示操作, 这样就能缩短从开机键按下至显示开机画面的时间。如图 6 所示, 在优选实施过程中, 第二级引导程序 BOOT2 模块启动过程包括 :

[0051] 步骤①, 初始化 LCD 等必要的外设组件接口, 然后执行步骤② ;

[0052] 步骤②, 从 SDRAM 中读取由第一级引导程序 BOOT1 拷贝好的 StartPicture 数据, 并显示开机图片至 LCD 屏, 然后执行步骤③ ;

[0053] 步骤③, 加载主程序内核 ThreadX0s. bin 文件至主内存 SDRAM, 然后跳转到步骤④ ;

[0054] 步骤④, 由第二级引导程序跳转至主程序内核 ThreadX0s. bin 开始执行。

[0055] (4) 通过编译、链接及分散加载的技术, 如图 4 所示, 将单个较大的主程序文件拆分为三个文件 : 操作系统内核以及必要的组件 ThreadX0s. bin、应用程序文件 APP. bin 以及资源文件 Resource. bin。将 ThreadX0s. bin 和 App. bin 存放在 NAND FLASH 的 BIN CODE 分区, 而 Resource. bin 存放在一个独立的分区, 将其作为资源文件访问, 而无需全部加载和运行。在第二级引导程序 BOOT2 中只需要加载 ThreadX0S. bin 并开始执行, 此时按键、UI 等交互界面可以正常使用 ;

[0056] 将主程序划分为多个独立的部分可以通过以下步骤实现 :

[0057] 步骤①, 调整编译、链接以及分散加载脚本 (Scatter) 以便适应多文件的生成和加载。具体包括 :

[0058] 根据目标存储器系统分散加载映像 (scatter-loading), Scatter-loading 文件是 ARM 的工具链里面的一个特性, 作为程序编译过程中给连接器使用的一个参数, 用来指定最终生成的目标映像文件运行时的分布状态。一个复杂的系统可能会把映像分割成几个部分。如图 5 所示, 系统中存在多种类型的存储器, 不同的代码部分根据执行性能优化的考虑分布于不同的地方。此时, 在 scatter-loading 文件里, 可以给编译出来的各个目标模块指定运行地址。

[0059] 步骤②, 调整系统的分区架构, 扩展一个分区用于手机资源文件的存放。

[0060] 步骤③, 修改下载工具以便适应 ThreadX0s. bin、APP. bin 和 Resource. bin 文件的下载。

[0061] 操作系统映像文件 ThreadX. bin 开始执行, 并初始化外设驱动, 至此阶段人机交

互已经可以正常进行。

[0062] 上述 ThreadX0S. bin, 开始加载 App. bin 进行必要的应用程序的执行, 例如网络应用等相关性的应用程序, 用户可以根据预置优先级预置那些应用优先级高, 然后系统在开机时根据优先级不同加载对应的应用程序。根据各个应用进程的需要访问 Resource. bin 中的图片资源文件和文件系统分区的文件。其中, 由于 Resource. bin 只是作为资源文件, 不需要完全加载到 SDRAM 中进行运行使用, 譬如里面会存放一些手机 UI 中使用的图片资源文件等。

[0063] 需要说明的是, 虽然上述给出一种将主程序划分为三个独立部分的实施方式, 但并不限于此, 在实际应用中也可以有其它的实施方式, 例如, 可以将上述 App. bin 根据优先级划分为多个部分, 或者, 将 Resource. bin 根据资源的类型划分为多个部分。

[0064] 如图 7 所示, 在优选实施过程中, 主程序运行步骤包括:

[0065] 步骤①, 主应用程序内核 ThreadX0s. bin 初始化, 然后执行步骤②;

[0066] 步骤②, 初始化键盘等外设驱动接口, 然后执行步骤③;

[0067] 步骤③, 据应用程序重要性优先级加载 APP. bin 应用程序和加载在显示中会用到的 Resource. bin 的图片资源文件, 其中, 用户可以根据需要修改应用程序启动优先级, 从而达到灵活的启动配置需求。然后跳转到步骤④;

[0068] 步骤④, 执行网络应用、UI 等应用程序。

[0069] 通过这种方式有效的将系统执行文件减少至 10M Bytes 左右的大小, 减少了手机在开机引导加载的文件大小, 从而减少了启动时间, 有效的加快了手机的开机速度。紧接着加载应用程序, 并开始进行必要的应用程序的执行, 例如网络应用等相关性的应用程序。

[0070] 从以上描述可以得知, 本优选实施例主要包括如下两大部分:

[0071] 一、将两级引导程序启动流程进行调整和修改, 将开机图片资源由第二级引导程序 B00T2 加载调整为第一级引导程序 B00T1 进行加载。因为在第一级引导程序中的坏块管理方案采用简单的跳除坏块的管理方式, 所以加载的速度较快, 效率较高, 同时将开机启动使用图片在第二级引导程序 B00T2 一开始就执行, 加快了开机图片的显示, 从而非常有效的提高了用户体验程度。

[0072] 二、通过编译、链接及分散加载技术将整个主程序分别编译链接生成三个独立的执行文件, 那么由第二级引导程序 B00T2 进行加载的执行文件减小为原来的 1/6 左右, 即由第二级引导程序到主程序的启动时间减少为原来的大约 1/6 左右。例如原来启动需要 30 秒钟左右的时间, 现在只需要 5 秒钟左右。

[0073] 综合上述优选实施例, 用户从按下开机键到看到 UI 界面以及人机交互操作时间缩短至原来的大约 1/5 左右, 并且不影响手机的正常操作使用, 提供较好的用户体验的前提下, 提高了手机的启动速度。

[0074] 图 8 为根据本发明实施例一种移动终端的结构框图。如图 8 所示, 该移动终端, 包括:

[0075] 数字基带芯片 80, 用于加载并执行第一级引导程序 B00T1;

[0076] 第一加载模块 82, 用于实现上述 B00T1 加载第二级引导程序 B00T2, 并将预先存储在上述 B00T2 划分的分区空间的开机图片资源文件复制到外界内存中, 然后转入上述 B00T2; 优选实施过程中上述 B00T1 采用跳除坏块的坏块管理方案。



[0077] 图片文件显示模块 84,用于实现上述 BOOT2 优先读取并显示上述外界内存中的上述开机图片资源文件;

[0078] 第二加载模块 86,用于实现上述 BOOT2 加载主程序,然后执行上述主程序。

[0079] 优选地,如图 9 所示,该第二加载模块还可以包括:

[0080] 编译子模块 860,用于预先将上述主程序分别编译链接为多个独立的文件,其中,上述多个独立的文件之一为包括操作系统内核的操作系统映像文件 ThreadXOS. bin;

[0081] 执行子模块 862,用于实现上述 BOOT2 加载上述 ThreadXOS. bin 并开始执行,以及实现上述 ThreadXOS. bin 根据需要加载上述多个独立的文件中的部分或全部文件。

[0082] 在优选实施过程中,上述多个独立的文件还可以包括以下至少之一:应用程序文件 App. bin、资源文件 Resouce. bin。

[0083] 上述执行子模块 862 还用于实现上述 ThreadXOS. bin 根据预置优先级加载并执行上述 App. bin 中的一个或多个应用程序以及实现上述 ThreadXOS. bin 加载并运行需要用到的上述 Resouce. bin 中的部分或全部资源。

[0084] 采用上述移动终端将原先的开机显示图片从主程序空间中移除,放入第二级引导程序的存储空间,加快了从用户按下开机键后的开机图片显示,加快了人机交互的显示和操作从而有效缩短开机启动时间,增强了用户体验效果。

[0085] 需要注意的是,上述移动终端中的各模块相互结合的优选工作方式具体可以参见上述方法实施例的描述,此处不再赘述。

[0086] 综合上述方法实施例和移动终端实施例,通过对基于 A2000PH 平台的 ThreadX 操作系统进行精简,并将 OS 执行文件、网络应用、应用程序通过 ADS 链接、编译和加载技术,分割成独立的多个文件,分为重要任务和非重要任务,进行分阶段加载和执行,解决了相关技术中基于 A2000PH 平台的 ThreadX 系统由于通常是经过二次引导启动,整个 BIN CODE 的映像文件加载完成后,才开始启动各个应用程序的运行(譬如执行代码、网络应用以及各种应用程序)而导致存在较长的启动时间的问题。从而大大提高了移动终端关键应用程序的启动速度。对于提高用户体验等用户感受的移动终端的快速启动有现实的应用背景。

[0087] 显然,本领域的技术人员应该明白,上述的本发明的各模块或各步骤可以用通用的计算装置来实现,它们可以集中在单个的计算装置上,或者分布在多个计算装置所组成的网络上,可选地,它们可以用计算装置可执行的程序代码来实现,从而,可以将它们存储在存储装置中由计算装置来执行,并且在某些情况下,可以以不同于此处的顺序执行所示出或描述的步骤,或者将它们分别制作成各个集成电路模块,或者将它们中的多个模块或步骤制作成单个集成电路模块来实现。这样,本发明不限制于任何特定的硬件和软件结合。

[0088] 以上仅为本发明的优选实施例而已,并不用于限制本发明,对于本领域的技术人员来说,本发明可以有各种更改和变化。凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

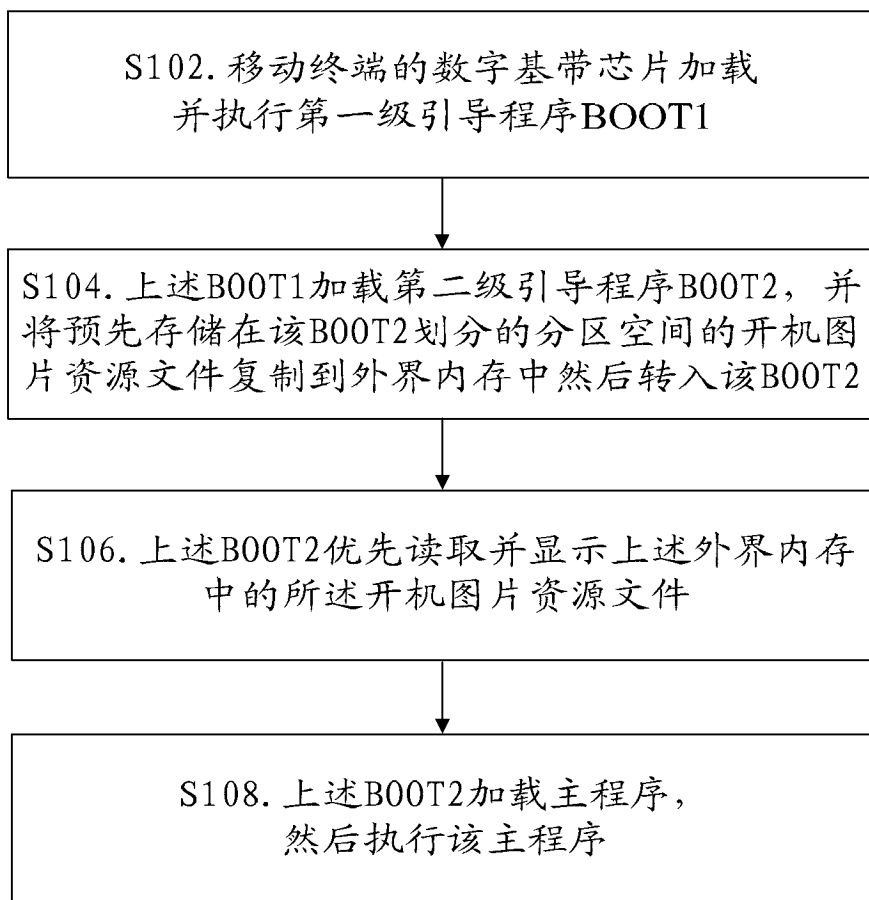


图 1

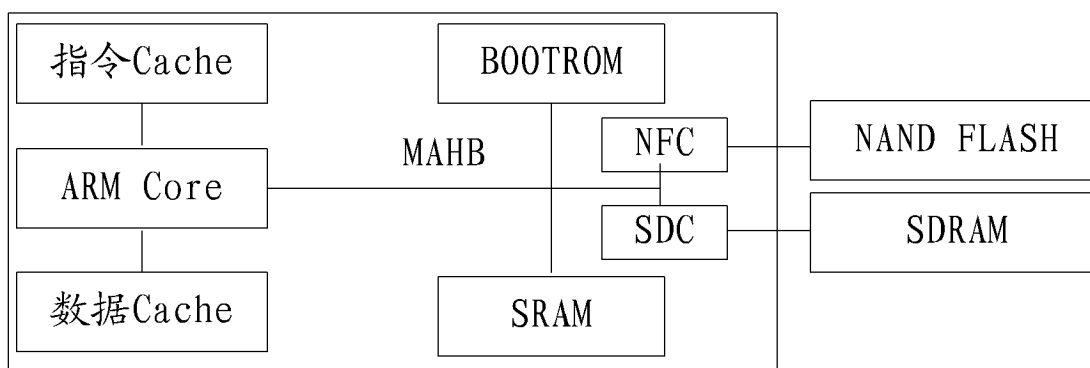


图 2

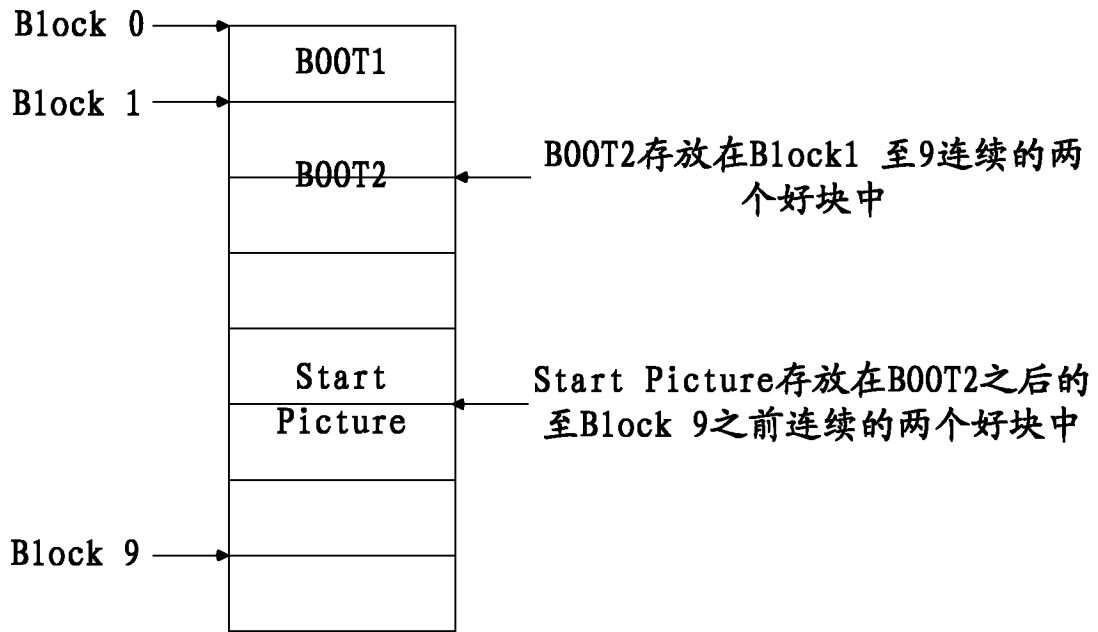


图 3

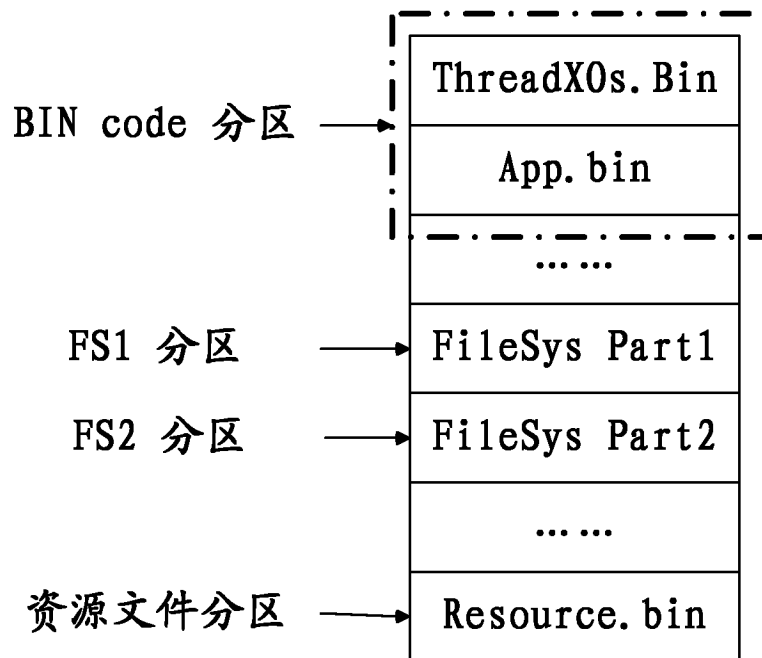


图 4

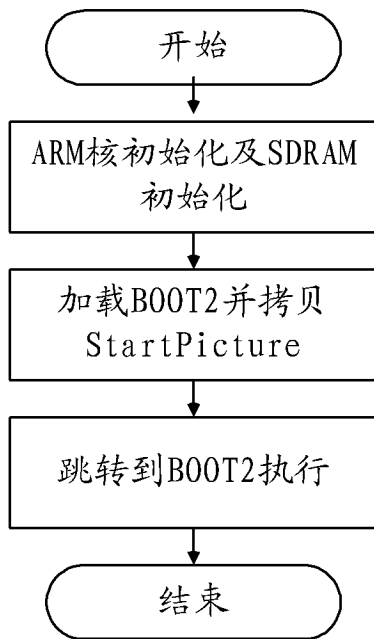


图 5

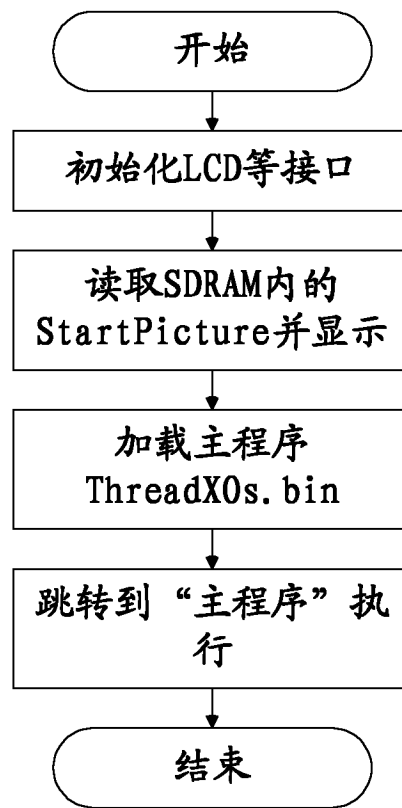


图 6

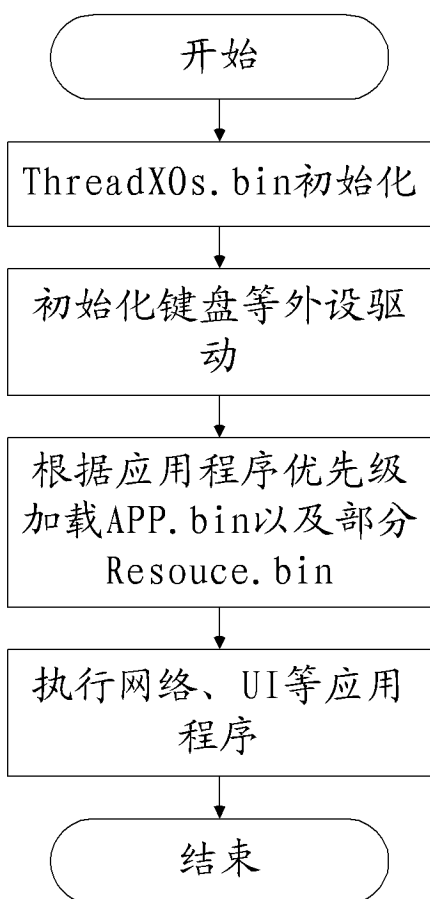


图7

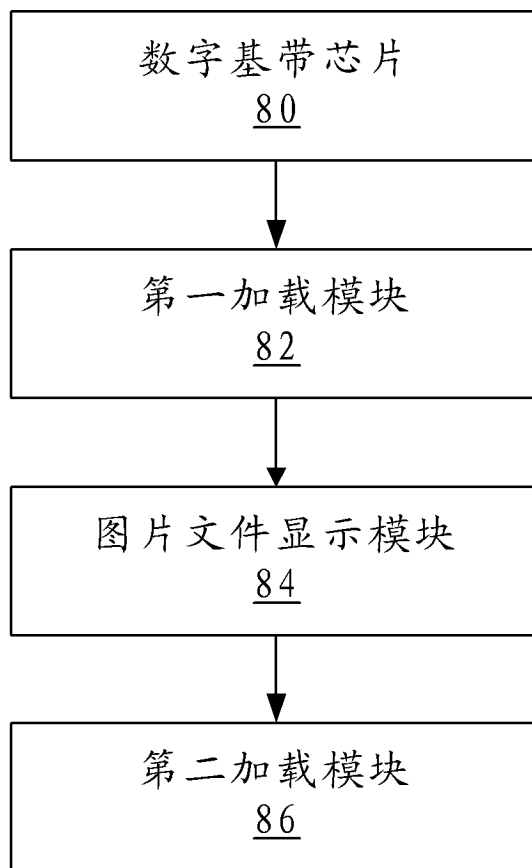


图8

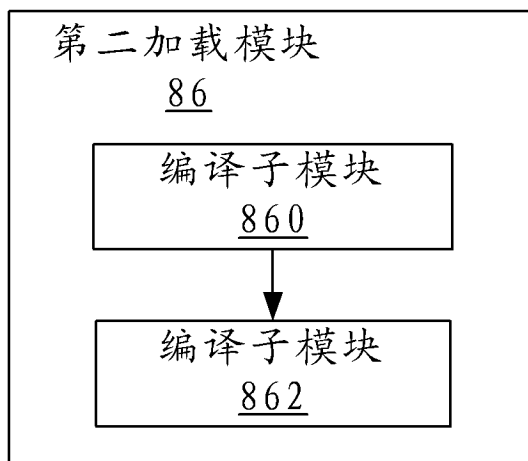


图9