

引用格式:陈成,王宜怀,钱涵佳,等. 基于 NB-IoT 的嵌入式远程软件更新系统设计[J]. 微电子学与计算机,2019,36(4):12-16. [CHEN C,WANG Y H,QIAN H J ,et al. Design of embedded device remote software update system based on NB-IoT [J]. Microelectronics & Computer,2019,36(4):12-16.]

基于 NB-IoT 的嵌入式远程软件更新系统设计

陈 成,王宜怀,钱涵佳,彭 涛

(苏州大学 计算机科学与技术学院,江苏 苏州 215000)

摘 要: 针对嵌入式设备软件更新繁琐、代价高昂以及 NB-IoT 上下行传输数据量少的问题,设计了一种基于 NB-IoT 的远程更新系统.并在硬件和软件方面进行优化,以节约更新所需的网络开销、缩短更新时间.在硬件层面上,对终端节点的 Flash 进行合理划分,设计出 Bootloader 区和驱动固化区等,以减少代码更新区域.在软件层面上,利用文本比较算法 Needleman/Wunsch,避免重复代码数据的传输.最后对远程更新系统进行测试,测试结果表明该系统更新代码量少,更新时间短,且具有较好的成功率.

关键词: NB-IoT;KL36;远程软件更新;Bootloader;Needleman/Wunsch

中图分类号: TN791

文献标识码: A

文章编号: 1000-7180(2019)04-0012-05

Design of Embedded Device Remote Software Update System Based on NB-IoT

CHEN Cheng,WANG Yi-huai,QIAN Han-jia,PENG Tao

(School of Computer and Technology, Soochow University, Suzhou 215000,China)

Abstract: In the face of the complexity and high cost of software update for embedded devices and NB-IoT has less uplink and downlink transmission data, a remote software update system was designed. And optimize the hardware and software to save the network overhead required to update and shorten the update time. The system makes a reasonable division of the terminal nodes Flash, design the bootloader area and the drive curing area, and reduces the code update area at the hardware level. And use the text comparison algorithm Needleman/Wunsch in the server software to avoid duplicate code data transmission at the software level. Finally, the remote update system is tested. The test results show that the system has less update code, shorter update time, and better success rate.

Key words: NB-IoT; KL36; remote software update; bootloader; needleman/wunsch

1 引言

随着物联网技术的快速发展,特别是 GPRS、Zigbee、NB-IoT 等远传技术在单片机上的大量应用,远程代码更新技术愈发显得必要.一般修改单片机程序往往需要对单片机产品进行回收才能实现,这对那些数量较多,分布较广的单片机产品而言就显得回收成本高昂和繁琐. NB-IoT 技术是

3GPP 标准定义的一种低功耗广域网解决方案,可支持人与物及物与物之间的连接^[1]. 因其具有大连接、深覆盖、低成本、低功耗等特点,NB-IoT 在智能抄表、智能停车、车辆跟踪、物流监控、智慧农业、智能设备等方面具有极大优势^[1,2],应用前景十分广阔,因此本系统选用 NB-IoT 作为远程更新的通信方式.

本文设计并实现一套基于 NB-IoT 的远程程序

更新系统,选择 KL36 和 BC95 为硬件平台来实现单片机远程程序更新操作.在深入分析机器码格式、Bootloader 机制等技术要点的基础上,优化单片机 Flash 的划分并引入文本比较算法 Needleman/Wunsch 来减少通讯流程中的网络开销.

2 系统总体介绍

本文实现一套基于 NB-IoT 的远程程序更新系统,该系统可通过服务器端软件更新与之相连的终端程序. NB-IoT 远程程序更新系统涉及到两个部分:终端和服务端.系统的总体模型如图 1 所示,

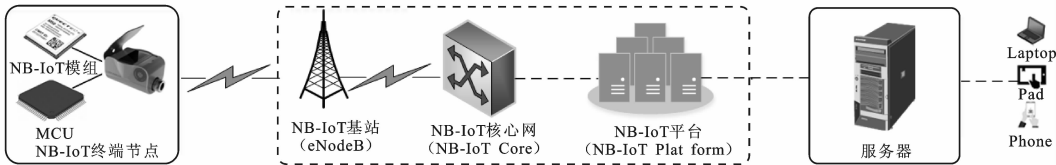


图 1 系统模型

在本系统中终端节点使用的是系统时钟频率可达 48 MHz 的低功耗芯片 MKL36Z64,该主控芯片是属于 ARM Cortex M0+内核,其拥有 64 KB 的 Flash、8 KB 的 RAM 和 48 个通用 IO^[3]. NB-IoT 通信模组选择的是 BC95,支持串口 AT 指令,数据传输方式支持 UDP 和 COAP 协议,同时具有高性能、低功耗等特点^[4].

3 终端节点 Bootloader 设计

NB-IoT 终端节点设计包含硬件设计和软件设

其中 NB-IoT 基站、核心网、平台由移动运营商负责维护.终端节点可通过 NB-IoT 通信模组直接与基站连接,在附着核心网成功后即可与服务器端更新软件进行数据交互以此完成更新代码的传递.远程代码更新工作流程为:服务器端软件解析程序机器码文件(Hex 文件),提取出有效数据并通过 NB-IoT 通信协议将命令或数据发送至指定的终端节点,终端节点接收到命令或数据后对片上 Flash 进行相应的复制、插入操作,最后分别接收服务器端校验和更新命令以实现更新代码的校验和远程更新操作.

计.其中硬件由 KL36 芯片最小系统、BC95 通信模组外围电路及其他电路组成,KL36 作为终端节点的主控芯片主要负责数据采集、运算和控制等操作.软件设计由 Bootloader 程序和真正的用户应用程序组成.Bootloader 是指终端芯片上电后运行的一段代码、其在用户应用程序运行之前执行,主要功能是 Flash 合理划分、各种硬件驱动初始化、更新操作判断,最终将终端节点带到合适的状态,为用户应用程序的执行准备好软硬件环境.

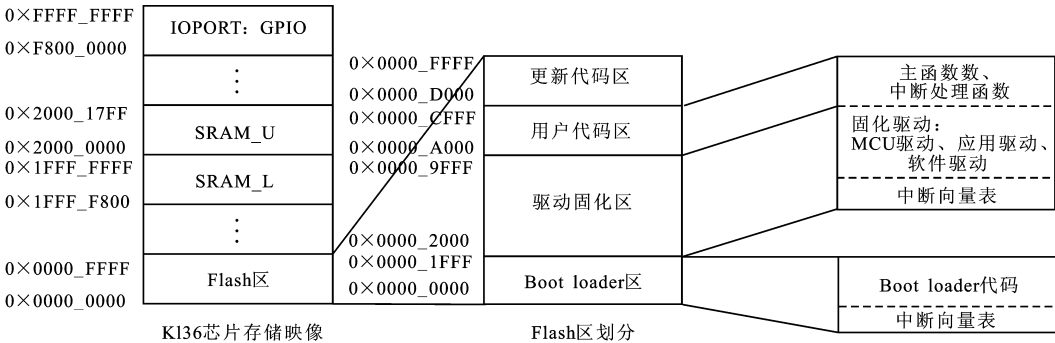


图 2 KL36 芯片 Flash 划分图

参考文献[5]提出一种基于 SD 卡的 GPRS 远程代码更新方式,但使用该方法需要在硬件层面增加一块 SD 卡,增加硬件成本的同时软硬件的可移植性也不高.参考文献[6]提出一种基于片上 Flash 分区方式的 Zigbee 远程代码更新方式,但使用该方法须将 Flash 一分为三,应用程序 Flash 所占区域

少于原 Flash 大小的一半,严重限制应用程序代码的规模.本文基于以上不足,提出一种新型的 Flash 分区方法,如图 2 所示.

本文将终端芯片 Flash 分为四个部分:Bootolader 区、驱动固化区、用户代码区和更新代码区.Bootloader 区是由 Bootloader 代码和 Bootloader 中断向量表

两部分组成,驱动固化区包括应用程序固化的驱动和应用程序中断向量表,用户代码区和更新代码区则分别包含正在运行的和待更新的主函数和中断处理函数.对终端节点而言更新程序往往只需修改主函数或中断处理函数,芯片的各个驱动是不需要变动的.因此本系统每次更新的代码仅为应用程序主函数和中断处理函数,这样可避免重复更新驱动固化区的各个驱动代码,减少更新过程中的代码量.

根据图 2 中的描述,用户应用程序中断向量表需重定位.原中断向量表起始地址寄存器 SCB→VTOR 指向地址 0x0000,即 Bootloader 程序的中断向量表地址.现需将其重定位至用户应用程序中断向量表起始地址 0x02000,具体可通过语句 SCB→VTOR=0x2000 来设置^[7].芯片 Flash 的具体划分可在链接文件中进行配置,链接文件可控制工程程序各部分在地址空间上的布局,即可将不同属性的内容放入不同的段中(代码和数据的基本单位是“段”).链接器按照链接文件中各段的起始地址和长度将各个段数据放入指定 Flash 区域,完成链接操作^[8].

同时需按照 Bootloader 程序和应用程序所占物理空间大小的不同和图 2 中 Flash 的划分机制合理划分 Bootloader 程序和应用程序中链接文件的配置.

Bootloader 程序链接文件 Flash 划分如表 1 所示.

表 1 Bootloader 代码 Flash 划分

段名	起始地址	长度	说明
m_interrupts	0x00000000	0x0100	中断向量表
m_flash_config	0x00000400	0x0010	Flash 配置域
m_text	0x00000410	0x1BF0	代码、常量区
m_data	0x1FFFF800	0x2000	数据区(RAM)

应用程序链接文件 Flash 划分如表 2 所示.

表 2 应用程序代码 Flash 划分

段名	起始地址	长度	说明
m_interrupts	0x00002000	0x0100	中断向量表
m_flash_config	0x00002400	0x0010	Flash 配置域
m_text	0x00002410	0x7BF0	驱动固化区
m_fun_text	0x0000A000	0x6000	用户代码区
m_data	0x1FFFF800	0x2000	数据区(RAM)

Bootloader 程序的起始地址从 0x0000 开始,可执行代码起始地址为 0x0410.Bootloader 程序最大占 Flash 大小为 8 K,主要负责程序更新操作判断,若需更新则将更新代码区 Flash 数据复制到用户代码区.应用程序的中断向量表的起始地址设置为 0x2000,具体的程序代码分别存储在驱动固化区和

用户代码区.驱动固化区存放无需修改的驱动代码,如 GPIO 驱动、UART 驱动等,程序更新时无需变动.用户代码区存放主函数、中断处理函数代码,这两块函数代码与具体项目紧密相关,也是代码更新的变动所在.其中驱动固化区占有 30KB 的 Flash,可包含 KL36 芯片所有的基础硬件驱动.

4 服务器端更新软件设计

4.1 文本有效数据提取

Hex 文件是可以烧写到单片机中,被单片机执行的一种文件格式.本系统服务器端更新软件从 Hex 文件中提取有文本有效数据.Hex 的全称是 Intel HEX,Intel HEX 文件以行为单位,每一行包含一个 Hex 记录,这些记录由对应的机器码十六进制编码数字组成.每行 Hex 记录由 6 个部分组成^[9],如表 3 所示.Hex 文件共有 6 种记录类型,分别为数据记录、文件结束记录、扩展段地址记录、开始段地址记录、扩展线性地址记录和链接开始地址记录.数据段中的数据为一行 Hex 记录中的有效数据,校验和用于检验 Hex 记录的正确性,算法为:校验和=0x100-校验和之前所有 16 进制数据累加和.

表 3 Hex 文件记录行语义

	字段 1	字段 2	字段 3	字段 4	字段 5	字段 6
名称	开始符	数据长度	地址	记录类型	数据段	校验和
长度	1 字节	2 字节	2 字节	1 字节	n 字节	1 字节
内容	开始标志“:”		数据记录有效	记录类型:00~05	取决于记录类型	所有字节之和的补码

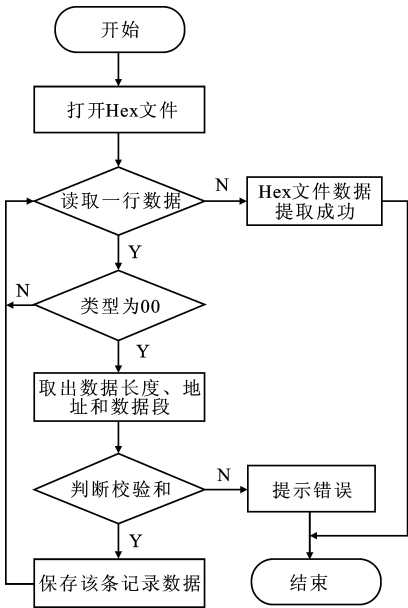


图 3 Hex 文件有效数据提取

图 3 为 Hex 文件有效数据提取流程图,本系统使用结构体变量储存每行数据,包括数据长度、地址和数据等.每次读取 Hex 文件一行数据,取出所有记录类型为 00 的数据并存入结构体中.

4.2 文本比较算法 Needleman/Wunsch

当代码更新变动不大时可以发现更新前后 Hex 文件有较多相同代码段.本文针对终端节点代码更新变动较少的特点,结合文本比较算法 Needleman/Wunsch,优化降低更新过程中所需的网络传输量.文本比较算法 Needleman/Wunsch 是

表 4 通信帧格式

名称	帧头	帧长	总帧数	当前帧数	命令	复制/插入个数	Flash 数据	校验和	帧尾
字节数	2	2	2	2	1	1	n	2	2

$$c[i,j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ c[i-1,j-1]+1 & \text{if } i,j > 0 \text{ and } x_i = y_j \\ \max(c[i,j-1],c[i-1,j]) & \text{if } i,j > 0 \text{ and } x_i \neq y_j \end{cases}$$

最后可根据矩阵 c 使用动态规划算法自底向上求得两段文本之间的最大公共子序列.当 $c[i,j] = c[i-1,j-1]+1$ 时,表示 X_{i-1} 与 Y_{j-1} 的最长公共子序列加上 x_i 得到的子序列即为 X_i 与 Y_j 的最长公共子序列.

要想将文本比较算法 Needleman/Wunsch 用于本远程更新系统,需对提取出的 Hex 有效数据进行一定的优化处理,本系统将 Hex 文件每一行有效数据当作一个序列.当比较 x_i 和 y_j 时,即比较新旧 Hex 文件行数据中的有效数据.若有效数据相同则 x_i 和 y_j 是序列 X 和 Y 的公共子序列.将新旧两个 Hex 文件进行文本比较算法 Needleman/Wunsch 处理后,可以知道本次远程更新中不变的代码有哪些,需要修改的代码有哪些.无需更改的只需告知终端节点该代码段原 Flash 起始地址、长度和现 Flash 起始地址,即可从现有 Flash 中获取代码数据.需修改的则需要告知终端节点该代码段需插入的 Flash 起始地址、长度和数据,然后进行 Flash 数据插入操作.

因此使用文本比较算法 Needleman/Wunsch 可避免传输 Flash 中已有的、无需变化的代码段,仅需传递待变动的代码.对于终端节点更新代码变动不大的情况下,本算法的应用可大大减少远程更新代码网络开销.

5 通信及程序更新

5.1 服务器端与终端节点通信

服务器端更新软件提取出 Hex 文件有效数据

使用动态规划的思想找出两段文本之间的最大公共子序列.因此使用本算法需在服务器端更新软件导入新旧两份 Hex 文件来作比较.

文本比较算法 Needleman/Wunsch 思想如下:
 $X_i = \langle x_1, x_2, \dots, x_i \rangle$ 为 X 序列的前 i 个字符 ($1 \leq i \leq m$), $Y_j = \langle y_1, y_2, \dots, y_j \rangle$ 为 Y 序列的前 j 个字符 ($1 \leq j \leq n$), 序列 X_i 和 Y_j 的最长公共子序列的长度记录在 $c[i,j]$ 中.当 $i = 0$ 或 $j = 0$ 时, X_i 和 Y_j 的公共子序列为空,故 $c[i,j] = 0$. 其他情况下,可由如下关系求解^[10].

后,可通过 NB-IoT 通信方式将更新代码发送至指定的终端节点.本文设计了服务器端软件与终端节点之间的通信帧格式,来确保远程更新过程中数据的可靠性和安全性,如表 4 所示.根据命令字节的不同可分为开始命令帧、复制数据帧、插入数据帧、校验命令帧和更新命令帧五种帧.

开始命令帧,命令字节为 0.通信帧格式中仅总帧数和命令字节数据有效,可通知终端节点更新操作开始,终端节点进行硬件环境准备.

复制数据帧,命令字节为 1.终端节点可根据复制操作个数和数据内容项知道需进行复制操作的原 Flash 起始地址、长度和现 Flash 起始地址,即可根据终端节点现有的 Flash 内容进行复制操作,减少网络传输数据量.

插入数据帧,命令字节为 2.插入数据帧中数据内容项包含待插入的 Flash 起始地址、长度和数据,根据该内容可进行终端节点的 Flash 插入赋值操作.

校验命令帧,命令字节为 3.仅命令字节有效,根据总帧数和每次收到的通信帧中的当前帧数进行判断,若接收帧数完整则校验成功,否则终端节点上传告知服务器端软件未接收到的帧号.

更新命令帧,命令字节为 4.当校验命令帧执行成功后,方可向终端节点发送更新命令帧.终端节点接收到更新命令后,进行 Flash 更新标志设置,然后系统复位进入 Bootloader 程序中识别 Flash 更新标志,最后进行更新操作.

5.2 代码更新

终端节点和服务端软件按照 4.1 节中介绍的通信帧格式完成更新代码的传递后会进行终端节点

代码的更新. 按照图 2 中 Flash 区划分角度来观察, 更新代码区用于存放待更新的代码, 在接收到更新命令帧后, 系统复位转向 Bootloader 程序执行. 在 Bootloader 程序中先对用户代码区的扇区进行擦除, 读取更新代码区数据, 最后将代码数据写入用户代码区. 待更新代码替换完毕后, 重设主堆栈指针 MSP 为用户应用程序起始地址, 并将 PC 指针指向用户应用程序 Reset_Handler 函数, 以实现程序跳转功能. 跳转函数代码如下.

```
void bootloader_jumpToUser(vuint_32 userAddr)
{
    uint_32 resetAddr;
    volatile fun_reset_handler reset_handler; //函数指针
    resetAddr = * (vuint_32 *) (userAddr + 4); //Reset
中断
    if(userAddr != 0xffffffff){
        reset_handler = (fun_reset_handler)resetAddr;
        __set_MSP( * (vuint32_t *)userAddr); //设置主栈
        reset_handler(); //跳转用户程序
    }
}
```

6 实验验证与评估

在以 KL36 和 BC95 为硬件平台的基础上对本文设计的远程软件更新系统进行更新测试, 以验证软件更新的实时性、可靠性和网络开销.

更新场景如下: 修改终端节点信息上传时间间隔为 1 小时, 并添加采集终端节点温度功能、数据上传失败蜂鸣器报警功能.

本文同时对本系统更新不同大小的程序所需的网络开销和更新时间进行了对比测试, 实验结果如表 5. 其中原系统方案是指将 Hex 文件中所有有效数据传递给终端节点.

表 5 更新测试对比表

组别	更新方案	更新代码字节(k)	更新次数	成功次数	平均更新时间(s)
组别 1	原系统方案	30.0	30	28	32.4
	本系统方案	1.1	30	30	2.7
组别 2	原系统方案	27.6	30	29	29.5
	本系统方案	1.2	30	29	2.9
组别 3	原系统方案	43.2	30	27	44.7
	本系统方案	2.7	30	29	4.1

测试结果表明: 使用本系统实验与目标设计一致, 更新所需网络开销和时间大大减少. 同时验证本系统所需的网络开销和更新时间与更新程序的大小无关, 而是与更新程序的变动量相关. 这对更新变动量较小的程序而言具有很大的优势, 且随着网络传输数据的减少, 更新成功率也大大提高, 保证了更新系统的可靠性.

7 结束语

本文设计一款基于 NB-IoT 的远程软件更新系统, 在终端节点部分对 Flash 进行合理划分, 将无需修改的驱动构件放置在驱动固化区, 代码更新时无需修改该区域代码. 在服务器端更新软件使用文本比较算法 Needleman/Wunsch 来优化更新过程中的网络开销, 避免相同代码的重复传输. 本文介绍的远程更新系统不仅仅适用于 NB-IoT 通信方式, 其他远程无线通信方式也可适用. 与传统的远程程序更新相比, 本系统可靠、可行、实时性强、网络开销少、更新时间短, 同时具有很好的稳定性与可移植性. 目前, 该方案已在 NB-IoT 远程水质监测系统中成功应用, 实践结果表明, 该设计方法稳定可靠、更新成功率高.

参考文献:

[1] 曲井致. NB-IoT 低速率窄带物联网通信技术现状及发展趋势[J]. 科技创新与应用, 2016(31):115-115.

[2] WANG Y P E, LIN X, Adhikary A, et al. A Primer on 3GPP Narrowband Internet of Things[J]. IEEE Communications Magazine, 2017, 55(3):117-123.

[3] JOSEPH YIU. ARM Cortex-M0 与 Cortex-M0+权威指南(第 2 版)[M]. 北京:清华大学出版社, 2018.

[4] Quectel_BC95_AT_Commands_Manual_V1. 6 [DB/OL]. [2017-07-03]. <http://www.quectel.com/cn/>.

[5] 顾会光, 王宜怀, 史新峰. 数据无损的远程代码更新的设计与研究[J]. 计算机工程与设计, 2015, 36(10): 2633-2639.

[6] 陈瑞杰, 王宜怀, 李会. 一种无线代码可靠更新系统的研究与设计[J]. 现代电子技术, 2016, 39(4):67-71.

[7] 陈发堂, 郭丽强. ARM+DSP 嵌入式系统 BootLoader 在 LTE 中的实现[J]. 电子技术应用, 2013, 39(5): 25-28.

[8] 汪春华, 白稳峰, 刘胤博, 等. 基于 CAN 总线 UDS 服务 BootLoader 应用开发[J]. 电子测量技术, 2017, 40(2):166-170.

March MDI_b 算法降低了约 24%，然后设计了 MBIST 电路,对 March DS 算法进行了仿真验证,结果表明该算法可以覆盖 SRAM 单存储单元动态故障,最终将设计的 MBIST 电路应用于 16 nm 存储器测试中,测试结果表明,本文提出的新的动态故障测试算法能够有效探测到 FinFET 工艺中可能出现的新型的单存储单元的动态故障.

表 5 芯片测试结果 (typical, 25℃, VDD=0.88 V)

IP 名称与容量	Die0	Die1	Die2	Die3	Die4
SPW720B160M4B1	Pass	Pass	Pass	Pass	Pass
SPW1568B160M4B1	Pass	Pass	Fail	Pass	Pass
SPW9600B24M16B2	Pass	Pass	Pass	Pass	Pass
SPW1568B144M4B2	Pass	Pass	Pass	Pass	Pass
SPW1280B13M8B1	Pass	Pass	Pass	Pass	Pass
R1W22B142M1B1	Pass	Pass	Pass	Pass	Pass
R1W256B144M1B1	Pass	Pass	Pass	Pass	Pass
R1W1024B144M1B1	Fail	Pass	Pass	Pass	Pass
R1W32B144M2B1	Pass	Pass	Pass	Pass	Pass
R1W512B144M1B1	Pass	Pass	Pass	Pass	Pass
良率	90%	100%	90%	100%	100%

参考文献:

[1] HARUTYUNYAN G, TSHAGHARYAN G, VARDANIAN V, et al. Fault modeling and test algorithm creation strategy for FinFET-based memories[C]//Vlsi Test Symposium. IEEE, 2014;1-6.

[2] HAMDIOUI S, AL-ARS Z, GOOR A J V D. Testing Static and Dynamic Faults in Random Access Memories[C]// IEEE Vlsi Test Symposium. IEEE Computer Society, 2002;395.

[3] BENSO A, BOSIO A, CARLO S D, et al. Automatic March tests generation for static and dynamic faults in SRAMs [C]//Test Symposium, 2005. European. IEEE, 2005;122-127.

[4] HARUTUNYAN G, VARDANIAN V A, ZORIAN

Y. Minimal March Tests for Detection of Dynamic Faults in Random Access Memories [J]. Journal of Electronic Testing, 2007, 23(1):55-74.

[5] DU L, ZHAO H, YANG W, et al. Optimization of STI oxide recess uniformity for FinFET beyond 20nm [C]// Semiconductor Technology International Conference. IEEE, 2015;1-4.

[6] JOSEPH J, PATRIKAR R. Impact of Fin Width and Graded Channel Doping on the Performance of 22nm SOI FinFET [M]//VLSI Design and Test. Springer Berlin Heidelberg, 2013;153-159.

[7] HARTYUNYAN G, SHOUKOURIAN S, VARDANIAN V, et al. Extending fault periodicity table for testing faults in memories under 20nm [C]//IEEE East-West Design& Test Symposium. IEEE, 2015; 1-4.

[8] HAMDIOUI S, GAYDADJIEV G N, GOOR A J V. A Fault Primitive Based Analysis of Dynamic Memory Faults[C]//IEEE 14 Annual Workshop on Circuits, Systems and Signal Processing. Veldhoven, Netherlands. IEEE, 2003;84-89.

[9] 石磊, 王小力. 一种基于存储器故障原语的 March 测试算法研究[J]. 微电子学, 2009, 39(2);251-255.

[10] MARINESCU M. Simple and Efficient Algorithms for Functional RAM Testing[C]//International Test Conference 1982, Philadelphia, Pa, USA. DBLP, 1982; 236-239.

[11] BOSIO A, DILILLO L, GIRARD P, et al. Advanced Test Methods for SRAMs [M]. Springer US, 2010.

作者简介:

桑胜男 男,(1993-),硕士研究生. 研究方向为集成电路设计与测试. E-mail:ssnsang@163.com.

张立军 男,(1971-),博士,研究员,博士生导师. 研究方向为 SOC 设计及方法学.

(上接第 16 页)

[9] 王宜怀,朱仕浪,郭芸. 嵌入式技术基础与实践(第 4 版)[M]. 北京:清华大学出版社,2017.

[10] 姜鲜桃. 双序列比对 Needleman-Wunsch 算法研究 [D]. 呼和浩特:内蒙古农业大学,2017.

作者简介:

陈成 男,(1993-),硕士研究生. 研究方向为嵌入式系统

与物联网. E-mail:20165227004@stu.suda.edu.cn.

王宜怀 男,(1962-),博士,教授. 研究方向为嵌入式系统、传感网与智能控制技术.

钱涵佳 女,(1994-),硕士研究生. 研究方向为嵌入式系统与物联网.

彭涛 男,(1989-),博士研究生. 研究方向为物联网与图像处理.