

# 嵌入式系统低功耗软件技术研究

姚 伟

(北京工业大学 计算机学院, 北京 100124)

**摘 要:** 嵌入式系统低功耗设计中有硬件技术无法涉足的空间, 可通过低功耗软件技术实现降低系统功耗的目的。针对液晶显示器(LCD)电气特性, 从软件角度, 综合运用动态电源管理技术和动态电压管理技术, 根据处理器负载变化趋势和对空闲模式计时的思想, 给出了降低液晶显示器功耗的算法和策略; 利用优化编译技术中的操作替换和指令排序方法, 分析和研究图形图像处理中常见的矩阵变换算法, 给出了低功耗策略和验证节能73.9%。并就低功耗软件技术算法和策略给出了结论和提出了下一步研究的方向。

**关键词:** 低功耗; 动态电源管理; 动态电压调节

**中图分类号:** TP302

**文献标识码:** A

**文章编号:** 1673-629X(2011)01-0112-04

## Study of Embedded System Low-Power Software Technology

YAO Wei

(Dept. of Computer, Beijing University of Technology, Beijing 100124, China)

**Abstract:** There is a research field for low power technology that can't be mentioned only by the hardware design, so then using low-power software technology can reduce system power consumption. From the software perspective, according to changes in processor load and idle mode, and taking full advantage of LCD electrical characteristics, the algorithm, using dynamic power management and dynamic voltage management combined, is put forward to reduce LCD power consumption. Furthermore, Operation Replacement and Optimization Instruction Sequence are used to analysis and research of matrix transformation algorithm frequently using in graphics and image processing and give the low power strategy decreasing 73.9% energy. According to low-power algorithms and strategies, it gives the conclusion and further research direction.

**Key words:** low-power; DPM; DVS

## 0 引 言

伴随嵌入式设备广泛应用, 低功耗技术愈显重要, 从软件角度研究低功耗问题也变得更加突出。用软件技术降低功耗, 一般从优化编译、操作系统和应用程序入手, 文中以降低 LCD 和矩阵运算功耗为例, 介绍低功耗软件技术的策略和方法。

## 1 研究意义与现状

### 1.1 研究意义

近几年, 嵌入式系统(embedded system)已经成为电子信息产业中最具增长力的一个分支<sup>[1]</sup>。随着手机、PDA、GPS、机顶盒等新兴产品的大量应用, 嵌入式系统的市场虽然正在以每年30%的速度递增, 而功耗问题严重制约嵌入式设备的进一步的发展、应用和推

广。不管是在航天、航海, 还是在工业、家庭、商业贸易上的应用, 都是靠电池供电, 然而电池技术的发展远远跟不上嵌入式技术的发展, 所以降低功耗, 延长电池更换周期变成了迫切的问题。

因此, 嵌入式低功耗软件核心技术的意义深远:

1) 降低嵌入式设备制造成本, 高功耗的设备肯定需要高成本部件来降低温度, 以保持性能稳定。

2) 提高系统的可靠性, 功耗密度的增加会对系统的可靠性和性能影响加剧, 一般温度每增加10度, 系统的失效率会增加一倍。反过来, 功耗降低可靠性增加。

3) 延长电池供电时间, 一般的嵌入式设备都要靠电池供电, 设备的功耗降低, 自然会使电池的工作时间延长。

4) 减轻嵌入式设备重量, 方便消费者使用, 电池在嵌入式设备重量中占有很大比例, 功耗降低使电池的重量降下来, 嵌入式设备重量相应的就会降下来, 消费者使用就会更轻便。

收稿日期: 2010-04-22; 修回日期: 2010-07-13

基金项目: 北京市优秀人才培养基金项目(20081D0501500169)

作者简介: 姚 伟(1984-), 男, 山东人, 硕士研究生, 研究方向为嵌入式软硬件设计。

5) 扩大嵌入式设备市场规模,随着电池寿命延长,设备重量降低和性能稳定,给消费者带来更多的便利和满足,势必会扩大使用范围,增大市场规模和占有率。

6) 节能环保天地广阔,全球变暖,冰川融化,煤炭石油消耗严重,而且不久的将来面临枯竭的危机,这些都迫切要求我们节约能源,保护环境,低功耗技术恰恰能够发挥重要的作用,而且随着大功率电子设备突飞猛进的增长将会产生更为深远的影响力。

## 1.2 国内外研究现状

鉴于低功耗技术的意义深远和重要性日趋明显,低功耗技术研究(Low-Power Design)变成了首当其冲的关键问题,开始主要研究工作是硬件工程师注重通过底层的电路设计来降低功耗,例如低功耗器件逻辑,互连功耗优化、漏电流控制、布局封装,后来系统工程师又从体系结构考虑降低功耗,直到最近嵌入式工程师发现由于工艺等多方面因素的限制,仅仅从前两个方面考虑,已经没太多可以挖掘的潜力,由此开始转向软件低功耗核心技术的研究、开发和应用<sup>[2-5]</sup>。

尤其是近十年来软件的低功耗技术已经开始升温,大量的研究和实用说明,利用软件实现低功耗是完全可行的,值得深入研究。目前最受欢迎的软件低功耗手段是操作系统,编译器,应用软件四个方面实现。

世界各地的著名科研院所和有影响力的高科技公司都投入了巨大的人力、物力和财力,对低功耗技术进行研究和推广,例如:国外许多大学成立了低功耗技术小组、PARAPET 研究组,以及有关嵌入式移动计算方面的低功耗研究;国内有中科院的从事多线程低功耗编译优化技术研究的许多专家,及其他大学的一些教授和博士等都在致力于低功耗软件技术的研究和推广。然而,总的讲国内在低功耗技术研究方面无论是在开发人员的认知,还是投入的物力或人力都明显满足不了市场的需求。因此,对我们来说努力加快研究和积极的推广应用,无论对国家的经济、社会、军事、教育、文体医疗,还是对人们的日常生活都显的尤为重要和迫切。

## 2 优化编译

优化编译降低系统功耗的原则是在不降低或不明显降低程序执行效率的条件下,尽可能降低峰值功率、总能量消耗及保持功耗与性能之间的良好性价比,最终达到嵌入式设备运行良好和功耗最低的双赢目标。优化编译技术包括:均衡优化指令功能均、降低执行频率、提高执行速度、缩短执行时间、减少数据传输和片外总线的驱动次数等。仅通过对应用程序的指令功能

均衡优化和降低执行频率就有可能节省 50% 的功耗<sup>[6]</sup>。

### 2.1 减少冗余代码

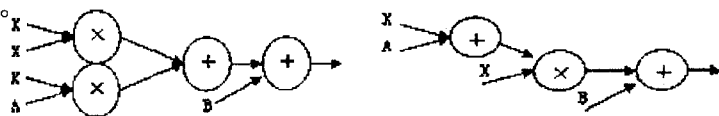
研究发现,应用程序即使经过性能优化,也仍存在 95.6% 的 load 指令和 42% 的 store 指令的冗余。片上存储器占处理器能耗的 30%,显然减少冗余对降低功耗影响重大。

### 2.2 降低 I/O 次数

处理器能耗中,驱动 I/O 也占用重要部分,因此,利用编码方法对 I/O 数据进行压缩优化来减少 I/O 频率,可以有效降低功耗。

### 2.3 操作替换

处理器工作电路中指令翻转引起电路变化产生功耗,因此会出现一些操作花费的功耗多于其他操作。如乘法操作比加法和移位更耗电;操作寄存器比操作内存要节省 25% 的功耗。选择合适的操作方式对降低功耗发挥着不可忽视的作用。将计算  $X^2 + AX + B$  经操作替换变成  $(A + X) \times X + B$ ,减少了操作次数,如图



1 所示。

图 1 操作替换

### 2.4 指令排序

优化指令排序能降低处理器运行功耗,处理器功耗公式:

$$P = I \times V \times N \times T \quad (1)$$

其中  $I$  是平均电流,  $V$  和  $T$  在电池供电的设备中已知,  $N$  是程序运行周期数。显然,充分利用嵌入式处理器的并行处理能力,通过优化指令排序,减少指令执行周期,能够实现降低系统功耗。

### 2.5 举例

运用处理器的并行处理能力,通过优化指令排序,减少指令执行周期,能实现降低系统功耗。图形图像处理中矩阵  $A[M][N]$  常有转换运算:

$$k \times A[i] + 2 \times A[i] \times A[j] \quad (2)$$

以加运算代替乘运算式(2) 转换:

$$(A[i] + A[i]) \times (R + A[j]) \quad (3)$$

其中  $k$  为实数,  $i, j$  是行和列 ( $1 \leq i \leq M, 1 \leq j \leq N$ )。式(3) 比式(2) 中减少 2 个乘运算,增加一个加运算 ( $R = k/2$ ),显然更节省功耗。式(3) 分解为  $a = A[i] + A[i], b = R + A[j], c = a \times b$ ,以  $A[i], A[j]$  代表矩阵中第  $i, j$  行元素参与运算。

分解后对其优化指令排列前后汇编代码和功耗如表 1 所示。

表 1 优化指令排序前后汇编代码及功耗对比

优化排序前代码	功耗 mA	优化排序后代码	功耗 mA
Move A[i], r0	90 mA	Move A[i], r0 A[i], r1	120 mA
Move A[i], r1	90 mA	Add r0, r1 R, r2 A[j], r3	150 mA
Add r0, r1	100 mA	Add r2, r3	100 mA
Move r1, a	90 mA	Mpy r1, r3, r4	160 mA
Move R, r2	90 mA	Move r4, c	90 mA
Move A[j], r3	90 mA		
Add r2, r3	100 mA		
Move r3, r4	90 mA		
Move a, r5	90 mA		
Mpy r4, r5, r6	160 mA		
Move r6, c	90 mA		

由表 1 可知优化后执行周期少 6 个、寄存器少 2 个,内存和寄存器操作次数分别由 7 次减至 5 次和 11 次减至 8 次。

由式(1)和参考文献[7]中所建立的模型,优化前后功耗比:

$$\begin{aligned} \frac{P_1}{P_2} &= \frac{I_1 \times V \times N_1 \times T}{I_2 \times V \times N_2 \times T} = \frac{I_1 \times N_1}{I_2 \times N_2} \\ &= \frac{1080 \times 11}{(120 + 150 + 100 + 160 + 90) \times 5} \\ &= \frac{11880}{3100} \end{aligned}$$

可见功耗降低了 73.9%。算法实现可先建立数据依赖图(Data Dependence Graph, DDG),再采用以串列为基础的排序机制<sup>[3]</sup>,程序中要兼顾减少程序执行周期、少用寄存器和少操作内存。因此优化指令排序和操作替换可以减少执行周期和对寄存器与内存的操作次数,从而既提高了系统性能,又大大降低系统功耗。

### 3 操作系统

#### 3.1 动态电源管理(DPM)

动态电源管理在降低嵌入式系统功耗发挥着不可替代的作用,属于操作系统级别的管理方式。动态电源管理的本质是一种电源管理的机制或策略,能够在系统运行时,动态地管理电源。嵌入式系统一般都有几种工作模式如:工作模式,空闲模式,睡眠模式,不同的系统对应的模式划分会有差异,但本质不变。针对不同模式间的转换,产生出不同转换策略,如超时策略、预测策略和随机策略<sup>[8,9]</sup>。

#### 3.2 动态电压调节(DVS)

根据 CMOS 电路电气特性,分析系统功耗有三部

分:动态功耗、静态功耗和短路功耗,其中动态功耗比重占 70% ~ 90%,随着发展静态功耗正在成指数增长,逐渐成为仅次于动态功耗,而短路功耗所占较小。重点研究系统的动态功耗和静态功耗,根据公式:  $P = f \times C \times V^2 \times N + I \times V$ ,其中  $f$  为频率、 $C$  为电容、 $V$  为电压、 $N$  为程序运行引起状态门平均转变数、 $I$  为电流。由此可知,DVS 的实质是因系统功率正比于频率、与电压成二次方关系,所以根据当前运行任务的需求,在保证性能前提下,动态的调整处理器工作电压和频率,以此来降低系统功耗。DVS 又分实时 DVS 任务调度和非实时 DVS 任务调度<sup>[10]</sup>。

#### 3.3 举例

据统计,一般嵌入式系统功耗比是 CPU 占 12%、LCD 占 68%,其他占 12%,现以便携式 LCD 为例,综合运用 DPM 和 DVS 技术,选用 Linux 操作系统,实现 LCD 低功耗算法,达到实低功耗管理 LCD 的目标。Linux 包括四种工作模式:Turbo 模式、运行模式、空闲模式和睡眠模式,其中 Turbo 和 Run 模式都是系统正常工作模式,在此统称正常模式。

##### 3.3.1 LCD 功耗特性

$$\text{LCD 功耗: } P = f \times C \times V^2 \times N + I \times V \quad (4)$$

$$\text{人眼感知亮度: } H = P \times L \times Y \quad (5)$$

其中  $f$ : 频率、 $C$ : 电容、 $V$ : 电压、 $L$ : 背光亮度、 $Y$ : 图像信号亮度、 $P$ : 液晶屏的亮度传输率(常数)。背光亮度和逆变器功耗占据显示器功耗的大部分。电池供电设备  $V$  不变,又  $f$  正比  $V$  于,所以  $P$  是  $f$  的三次方关系。因此了解这些,由式(4)和式(5)找到了降低功耗的切入点。

##### 3.3.2 LCD 电源管理模式

如图 2 所示, LCD 电源管理模式包括二个单元监视单元和控制单元,监视单元负责监视处理器负载变化,传递指令给控制单元;控制单元控制 LCD 工作状态。

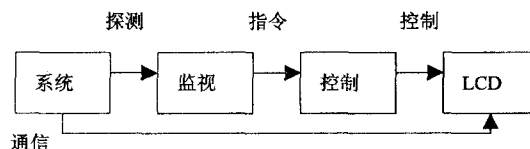


图 2 LCD 电源管理模式

##### 3.3.3 程序实现

Linux 内核有全局变量 `kstat` 中记录了 CPU 启动后运行在用户态和系统态的时钟滴答数 `ticks`,容易计算采样周期 100ms(经验值)内处理器处于负载工作的时间,二者相减得空闲时间,除去 100ms 得采样周期空闲比 `Idle_Time`。参考文献[8,9]中负载统计方法,根据三个连续采样周期内的空闲比的数据趋势和空闲模式计时器 `idle` 是否超时,判断处理器工作状态,做出控制

LCD 的决策。

```
int Get_Idle_Time(int * time) { 返回采样周期空闲比}
void Task_Detect() { Idle_Time[0~2] = Get_Idle_Time(int *
time);
If(Idle_Time[0] < Idle_Time[1] < Idle_Time[2]) { Contor_LCD
(1); } //空闲时间是增趋势
else { Contor_LCD(0); } //空闲时间是减趋势}
void Contor_LCD(int data) { if(idle 超时且 data==1) { LCD_In_
Sleep(); }
    else if(data==1) { LCD_In_Idle(); }
    else { void LCD_Down_Light(); } }
void LCD_In_Sleep() //LCD 进睡眠状态
{ if(LCD->State 不是睡眠) { 关闭背光灯、逆变器, 降低显示通
道时钟工作频率} }
Void LCD_Down_Light() //LCD 工作状态
{ 控制脉冲占空比动态减小 L、增大 Y }
void LCD_In_Idle() //LCD 进空闲状态
{ 降低 LCD f 和 L、使模数转换、锁相环电路等进入空闲状态 }
```

监视单元函数 Task\_Detect() 判断空闲比数增减, 向控制单元 Contor\_LCD() 传递 1 或 0。睡眠状态时关闭背光灯、逆变器, 降低显示通道时钟工作频率; 空闲状态时降低  $f$  和  $L$ , 使模数转换、锁相环电路等进入空闲状态。工作状态主要通过适度调整背光亮度  $L$  和图像信号亮度  $Y$  实现降低功耗。

注意,  $L$  太小会带来眼疲劳,  $Y$  增大也不应使饱和点像素的个数高于 5%。可以通过程序控制 LCD 微处理器 I<sup>2</sup>C 或 I/O 口关闭相应部件或降低频率。程序控制图像处理器, 应使输入信号场频略高于 40Hz, 从而既不使图像闪烁, 又降低 LCD 存储电容的刷新次数, 减缓帧缓冲内存的读写率及其总线的时钟频率, 达到降低功耗的目标。

#### 4 应用软件低功耗技术

不同于系统软件, 应用软件低功耗问题是常常被忽视。且至今, 没有一个严格的标准来判断应用软件的低功耗技术优劣, 但仍有一些被验证的能有效降低应用软件的功耗技术。

程序员很清楚对 RAM 的操作比对 Flash 的操作消耗更多功耗。应用程序中对子程序调用时会为了保存现场, 二次对 RAM 实现进堆栈(RAM)和退堆栈操作(RAM), 宏只是在编译时展开, 处理器顺序执行, 所以适度扩大 Flash 用宏代替子程序能明显降低功耗。相对于中断可以直接进入低功耗模式, 查询需要不停地访问 I/O 寄存器而消耗更多功耗, 用中断替换查询也是很好的低功耗策略。另外, 应用程序中在满足需

求的条件下, 尽量使用低精度的数据可以有效减少处理器指令翻转带来的功耗<sup>[11,12]</sup>。

#### 5 结束语

运用了处理器负载变化、动态电源管理、动态电压管理等技术, 就降低嵌入式外围设备功耗具有一定的代表性。利用优化编译技术降低矩阵运算的功耗对具体应用很有意义。

下一步研究则针对具体的处理器和 LCD, 由应用场景分析负载变化和 LCD 需求, 实现一套更优的降低功耗的技术。

#### 参考文献:

- [1] Wolf W. 嵌入式计算系统设计原理[M]. 孙玉芳, 等译. 北京: 机械工业出版社, 2002: 126-152.
- [2] Borkar S. Low Power design challenges for the decade (invited talk) [C]//In Proceedings of the 2001 conference on Asia South Pacific design automation. Yokohama, Japan: ACM Press, 2001: 293-296.
- [3] 赵荣彩, 唐志敏, 张兆庆, 等. 编译指导的多线程低功耗技术研究[J]. 计算机研究与发展, 2002, 39(12): 1572-1579.
- [4] Jung S, Kim K, Kang S. Low-Swing Clock Domino Logic Incorporating Dual Supply and Dual Threshold Voltages[C]//In Proceedings of the 39th conference on Design automation. New Orleans, Louisiana, USA: ACM Press, 2002: 467-472.
- [5] Amelifard B, Fallah F, Pedram M. Low-Power Fanout Optimization Using Multiple Threshold Voltage Inverters [C]//In Proceedings of the 2005 International Symposium on Low Power Electronics and Design. San Diego, California, USA: ACM Press, 2005: 95-98.
- [6] 赵荣彩, 唐志敏, 张兆庆, 等. 低功耗多线程编译优化技术[J]. 软件学报, 2002, 13(6): 1123-1129.
- [7] Gibbons P B, Muchnic S S. Efficient Instruction Scheduling for a Pipelined architecture[J]. SIGPLAN Notices, 2004, 39(4): 169-174.
- [8] 王力生, 夏志江. 嵌入式系统中软件优化的低功耗研究[J]. 单片机与嵌入式系统应用, 2006(4): 12-14.
- [9] 吴业进, 刘 锋. 嵌入式系统面向低功耗的协同设计[J]. 单片机及嵌入式系统应用, 2003(7): 14-16.
- [10] 杨雪金, 南余荣, 俞 立. 一种减少嵌入式处理器功耗的综合 DVS 方法[J]. 计算机工程与应用, 2005, 41(2): 42-50.
- [11] 李 允, 熊光泽, 程红蓉. 嵌入式系统的功耗管理技术研究[J]. 单片机与嵌入式系统应用, 2001(12): 21-25.
- [12] 王力生, 郭振轲. 基于 DVS 的实时多核嵌入式系统低功耗算法[J]. 计算机应用研究, 2009(1): 127-131.