应用情景检测的安全任务内电压调度算法

陈 杰，易本顺

(武汉大学电子信息学院 武汉 430079)

【摘要】针对移动嵌入式实时系统的低功耗设计，提出一种改进的应用情景检测的安全任务内电压调度算法。该算法利用任务程序少数参数的值域定义情景并在任务处理过程中进行检测，因而在处理在线的情况下，可较为精确地预测后续处理的部分路径，优化地调度处理电压。通过所提出的情景检测点设置算法，在任务程序中找到检测参数情景最合适的点，给出算法的实现方法。通过仿真实验证明该算法能有效地降低任务处理能耗。

关 键 词 动态电压调整； 任务内电压调度； 移动嵌入式实时系统； 情景检测

中图分类号 TP316 **文献标识码** A doi:10.3969/j.issn.1001-0548.2011.03.025

Safe Intra-Task Voltage Scheduling Algorithm Using Scenario Detection

CHEN Jie and YI Ben-shun

(School of Electronic Information, Wuhan University Wuhan 430079)

Abstract In this paper, an improved safe intra-task voltage scheduling algorithm using scenario detection is proposed for the energy-efficient designs of the mobile embedded real-time systems. The ranges of few parameters in the program of the task are used to describe scenarios which are detected in the process of the task execution, and some of the remaining execution paths can accurately be predicted in the on-line mode which can optimize the voltage scheduling. Furthermore, the most reasonable points in the program to detect scenarios are searched out by the proposed scenario detection point positioning algorithm. The implementing method of the algorithm is presented and the effectiveness on reducing energy consumption of the task execution is validated through the simulation studies.

Key words dynamic voltage scaling; intra-task voltage scheduling; mobile embedded real-time systems; scenario detection

随着移动电话、PDA、数码相机等移动嵌入式实时系统发展的日新月异，逐渐增多的大数据量处理与其电池供电、能量有限的矛盾也日益突出，促使研究者们探索延长电池工作时间的方法。动态电压调节技术及电压可调处理器的出现为降低处理器能耗提供了一种有效方法[1-2]，它们可针对处理器不同计算量的任务分配不同的电压等级，在满足任务处理实时性(也即安全性)的同时，减少处理器空闲时间、降低能耗，能获得很好的节能效果[3-4]。近年来，研究者们提出了任务内的电压调度策略，在任务程序中设置电压调整点(VSP)对电压进行实时调整，算法简单易行而节能效果更优。

根据剩余处理路径预测策略的不同，任务内的电压调度算法可分为基于剩余最坏情况处理路径(RWEP)[5]、基于剩余平均情况处理路径(RAEP)[6]及基于剩余最优化情况处理路径(ROEP)[7]共3类。因该3类基本算法在实际应用中存在节能效果不佳或实时性难以确保的缺陷，文献[8]提出了3种改进的基于RAEP的电压调度算法及其实时性确保方法，并提出应用任务程序的数据流和控制流信息标识VSPs，并将VSPs尽可能前移以优化调度任务处理电压。虽显著地降低了任务处理能耗，却引入了大量的预处理开销。文献[9]提出了基于RWEP的情景感知任务内电压优化调度技术(简称为SA-RWEP)，根据人为经验设置情景检测点(SDP)，对少数变量进行前摄检测、提前感知及利用空闲时隙，引入的预处理开销小、节能效果明显，性能更优。

基于文献[9]，本文提出一种改进的SD-RAEP。首先，引入基于RAEP的轮廓感知安全任务内电压调度算法(简称为PAS-RAEP)；然后，陈述在待处理的任务代码中选择参数及依其定义情景的方法；最后，提出SDP设置算法，以便在代码合理位置检测情景

收稿日期：2009－07－14； 修回日期：2011－03－30

作者简介：陈 杰(1982－)男，博士生，主要从事嵌入式实时系统功耗管理技术方面的研究.

的发生。算法节能效果明显、具有很好的实时性。

1　能耗模型与任务模型

处理器能耗由动态能耗和漏电能耗两部分组成。降低处理器工作电压会线性地降低工作频率，导致任务处理时间延长，而其能耗和功耗分别以大于工作电压二次方和三次方的幅值降低。因此，根据任务参数，在任务处理实时性得到满足的前提下，降低处理器工作电压、减少空闲时隙，不仅可以提高处理器利用率，而且可以显著地降低能耗[2]。

任务程序代码由基本结构顺序结构、分支结构(if…else…)及循环结构(while…do…)构成(其他结构都可转换为这3种基本结构[9])，因此，任务的处理量会随着结构条件表达式中参数取值的不同而发生变化，并且在完成该任务前无法预知与估计，可用任务程序对应的控制流图(CFG)对任务的处理过程进行描述[6]，运用任务内的电压调度算法在线选择基本块处理电压(或频率)，任务的处理能耗则为处理路径上各个基本块的处理能耗与基本块处理电压之间转换能量开销的总和，该转换能量开销较小，本文认为其值近似为0。

2　SD-RAEP算法

2.1　PAS-RAEP算法

SD-RAEP算法将具有最大加权概率的处理路径作为参考路径，并依此预测基本块的剩余平均处理时钟周期数为：

$$\delta_{\mathrm{a}}(b_i)=c(b_i)+\begin{cases}\delta_{\mathrm{a}}(b_j) & \delta_{\mathrm{a}}(b_j)p_j\geqslant\delta_{\mathrm{a}}(b_k)p_k\\ \delta_{\mathrm{a}}(b_k) & \text{其他}\end{cases}\tag{1}$$

式中，b_j 与 b_k 为基本块 b_i 的直接后继基本块；p_j 与 p_k 为完成 b_i 后处理 b_j 或 b_k 的概率；$c(b_i)$ 为完成 b_i 所需时钟周期数。参考路径的选择兼顾了剩余路径的概率及处理时钟周期数。循环结构循环首标及循环体内基本块的参考路径及 δ_{a} 的确定方式与基于RAEP的电压调度算法的确定方式[6]类似。本文将基于上述机制的电压调度算法简称为WP-RAEP。

为保证任务处理的实时性，PAS-RAEP算法[8]还计算了基本块的剩余安全处理时钟周期数 δ_{s}，根据 $(\delta_{\mathrm{a}},\delta_{\mathrm{s}})$ 值对，计算与直接前向基本块之间VSP的频率更新率 r，实现了任务内安全的电压调度、最小化处理能耗及VSPs。假定参考路径由基本块 $b_m,\cdots,b_n,b_{\mathrm{exit}}$ 构成，那么 $\delta_s(b_m),\cdots,\delta_s(b_n)$ 为：

$$\delta_s(b_l)=\begin{cases}\dfrac{D-\mathrm{lst}(b_m,\cdots,b_n)}{d(b_m,\cdots,b_n)-\mathrm{lst}(b_m,\cdots,b_n)}\\ (c(b_m)+\cdots+c(b_n)) & \text{当}l=m\\ \delta_s(b_p)-c(b_p) & \text{其他}\end{cases}\tag{2}$$

式中，b_p 为 b_l 的直接前向基本块；D 为任务截止期；$d(b_m,\cdots,b_n)$ 表示基本块组 $b_m,\cdots,b_n$ 的截止期；$\mathrm{lst}(b_m,\cdots,b_n)$ 表示处理 $b_m,\cdots,b_n$ 的最晚开始时间。由该定义知，$d(b_m,\cdots,b_n)$ 与 b_n 的截止期 $d(b_n)$、$\mathrm{lst}(b_m,\cdots,b_n)$ 与 b_m 的最晚开始时间 $\mathrm{lst}(b_m)$ 相等，基本块的截止期与最晚开始时间分别为：

$$d(b_i)=D-\frac{\delta_{\mathrm{w}}(b_i)-c(b_i)}{f_{\max}}\tag{3}$$

$$\mathrm{lst}(b_i)=\min\left(D-\frac{\delta_{\mathrm{a}}(b_i)}{f_{\mathrm{a}}(b_i)},d(b_i)-\frac{c(b_i)}{f_{\max}}\right)\tag{4}$$

式中，$f_{\max}$ 为最大处理频率；$\delta_{\mathrm{w}}(b_i)$ 为 b_i 的剩余最坏情况处理时钟周期数；$f_{\mathrm{a}}(b_i)$ 为WP-RAEP下 b_i 的处理频率。当任务处理路径经过边 (b_i,b_j) 之间的VSP时，调整处理频率为：

$$f(b_j)=f(b_i)r=f(b_i)\frac{\max(\delta_{\mathrm{a}}(b_j),\delta_{\mathrm{s}}(b_j))}{\max(\delta_{\mathrm{a}}(b_i),\delta_{\mathrm{s}}(b_i))-c(b_i)}\tag{5}$$

2.2　参数选择和情景定义

在任务程序代码中，一些参数取不同值时任务工作量变化较大，这些参数一般只能取少数几个值，可以在程序中执行输入数据或加入赋值语句的方式进行设定，且在余下部分(或全部)的任务处理过程中保持不变。基于该事实，可以利用这些参数的值域划分应用情景，在任务处理时根据所发生的情景较为精确地预测任务处理的部分路径，剔除多余路径，优化调度余下基本块电压、降低任务处理能耗[9]。

$$\mathrm{IC}_v(S)=0\tag{6}$$

$$\mathrm{IC}_v(B_1;B_2)=\begin{cases}\mathrm{IC}_v(B_2) & \text{当}v\text{值在}B_2\text{中发生了变化}\\ \mathrm{IC}_v(B_1)+\mathrm{IC}_v(B_2) & \text{其他}\end{cases}\tag{7}$$

$$\mathrm{IC}_v(\text{if }B\text{ then }B_1\text{ else }B_2)=\begin{cases}\max(\mathrm{WCEC}_{B_1},\mathrm{WCEC}_{B_2})-\min(\mathrm{WCEC}_{B_1}-\mathrm{IC}_v(B_1),\mathrm{WCEC}_{B_2}-\mathrm{IC}_v(B_2))\\ \qquad\qquad\text{当将}v\text{与常数的比较为}B\text{的条件，且值在}B_1\text{和}B_2\text{中未变化}\\ \max(\mathrm{IC}_v(B_1),\mathrm{IC}_v(B_2)) \quad \text{其他}\end{cases}\tag{8}$$

$$\mathrm{IC}_v(\text{while } B \text{ do } B_1)=\begin{cases} n_{\max}\mathrm{WCEC}_{B_1}-n_{\min}(\mathrm{WCEC}_{B_1}-\mathrm{IC}_v(B_1)), & \text{当}v\text{作为 }B\text{的条件，且}v\text{值在}B_1\text{中未变化} \\ 0 & \text{当 } v \text{ 值在 } B_1\text{中发生了变化} \\ n_{\max}\mathrm{IC}_v(B_1) & \text{其他} \end{cases} \tag{9}$$

根据各个参数取不同值时任务工作量的最大变化(简称为影响因子或IC)的大小，选择若干参数定义情景以在处理任务时进行检测。在该过程中，一般不必计算所有参数的IC，而只需选择在各个结构条件表达式中出现的参数进行计算即可。在计算参数 v 的 IC_v 时，以从后向前的方式遍历程序抽象句法树，按式(6)～式(9)规则计算各个结构的 IC_v，累加得到 v 对任务的影响因子，该计算过程基于最坏情况处理时钟周期数(WCEC)。式(6)～式(9)中，S 表示非控制型语句，v 对其影响因子为0；B 为条件表达式，B_1 及 B_2 为语句段；$n_{\min}$ 和 $n_{\max}$ 分别表示循环迭代的最小和最大次数，式(7)～式(9)为3种基本结构IC的计算规则。可选择IC相对较大的参数定义情景：在任务代码中，收集与每个所选参数进行比较的常数及相应比较符，利用其划分所选参数的取值区间，每个区间对应任务处理的一种情景。

2.3 SDP设置算法

根据所选定的参数和定义的情景，便可在任务程序中的适当位置设置SDP。SDP需检测任务处理过程中所发生的情景，由几行计算和判断语句完成。激活的情景应得到及时的感知，因此，在任务第一个基本块之前应设置SDP，检测程序输入数据所激活的情景，而在所选参数的值可确定时也应设置SDP。SDP设置算法的伪代码借用了文献[8]中变量前趋点及先行点的概念。设置SDP算法如下：

```
SDP_Setup( ) {
  SetSDP(0);
  for v_i ∈ V {
    S(v_i) = φ;
    S(v) = Find_Conditions_Line(v_i);
    for s_j ∈ S(v_i) {
      P(s_j, v_i) = Find_MDP(s_j, v_i);
      L(s_j, v_i) = Look_Ahead(P(v));
      Transform(L(s_j, v_i), s_j);
      SetSDP(L(s_j, v_i));
    }
  }
}
Find_MDP(s_j, v_i) {
  P = Find_Data_Predecessor(s_j, v_i);
  for p ∈ P {
    V'(p) = Find_Variables(p);
    if (V'(p) ≠ φ) {
      P = P − {p};
      for v_k ∈ V'(p)
        P = P ∪ Find_MDP(p, v_k);
    }
  }
  return P;
}
```

算法功能由函数SDP_Setup和Find_MDP实现。SDP_Setup完成SDP设置点的迭代搜索、确定及SDP设置全过程。首先调用SetSDP在任务第一个基本块之前设置SDP；然后调用Find_Conditions_Line搜索程序各个结构的条件表达式中包含先前选定参数 v_i，且以 v_i 的取值情况作为条件成立判据之一的行，函数返回其行号集 $S(v_i)$，将其中的元素称为基准点，针对基准点 s_j，应用Find_MDP函数，迭代搜索 s_j 处 v_i 的前趋点集 $P(s_j,v_i)$ 和先行点集 $L(s_j,v_i)$；最后在先行点处设置SDP。Find_MDP完成搜索参数多级前趋点集。首先搜索输入行处输入参数的前趋点集 P，称此时的输入参数为中间变量，前趋点 p 为中间变量前趋点；然后对 p 中等式右边的变量进行识别，构成变量集 $V'(p)$，如果其非空，则调用Find_MDP对 $V'(p)$ 中的变量继续进行迭代搜索，函数返回最终搜索结果。

对于存在一个或多个SDP的点，判断各个SDP所涉及的参数，如果参数 v_i 的每个相关情景发生时，任务的预测剩余处理路径与不进行情景检测时的预测路径相同，则在该点取消对 v_i 的情景检测。完成SDP的设置后，采用PAS-RAEP算法对参考路径基本块进行电压调度并处理，任务处理偏离参考路径时则重新确定参考路径并更改处理频率。当任务处理到达SDP设置处时，利用中间变量前趋点处的表达式(由Transform函数关联)计算 v_i 值后判断其哪一个情景被激活，并在相应基准点移除部分代码或判据，如所移除的代码中包含其他基准点，移除代码的同时也应移除这些基准点对应的SDP，之后仍需重新

确定参考路径及进行频率更新。

3　仿真实验与分析

通过仿真实验对SD-RAEP算法性能进行分析，算法各步骤在SUIF2平台上实现。假定处理器最小处理频率为20 MHz，最大可达66 MHz(如采用ARM7TDMI内核的微处理器S3C44B0X)，且在该范围内，频率可连续转换，转换开销为70 μs。分支结构任选的一条有向边的概率值由标准差为1、均值为0.5的正态分布计算得到，选择参考路径及基于RAEP的电压调度时，考虑情景检测所引入的开销，而将任务的截止期设置为最坏情况处理时间的1.5倍。通过MP3解码多媒体应用[10]，仿真实验验证了在移动嵌入式实时系统的低功耗设计中，应用SD-RAEP算法能显著地降低系统能量开销，保证任务处理的实时性，且比SA- RWEP性能更优。

表1　MP3解码器中IC值较大的参数

参数	IC
nch	1.202.984
block_type	1.053.728
mode_extension	104.418
mixed_block_flag	66.128

计算MP3解码器中参数的IC，其值大于100的参数已在表1中列出。由于第二个参数与第三个参数的IC值相差较大，因此，可以选择前两个参数定义情景。表2对SD-RAEP与SA-RWEP在平均节省能量及情景检测引入开销(以时钟周期数计)两项指标上进行了性能比较，解码器输入处理文件为随意在因特网下载的立体声歌曲，平均节省能量以基于RWEP的电压调度算法结果[5]为参考且表中所列为20次实验结果的平均值。同时，仿真实验也对只选最大IC值参数及选择两个以上参数定义情景的情况进行了比较与分析。

从表2中可知，SD-RAEP较SA-RWEP节省更多能量。基于RWEP的电压调度算法虽能较好地满足处理的实时性要求，然而，由于任务的处理很可能不会沿着最长路径且处理器频率范围有限，因此，基于该方法的电压调度可能会留有较多的空闲时隙，节能效果一般；而基于RAEP是以最大加权概率的处理路径作为参考路径的轮廓感知安全任务内电压调度算法，既考虑了剩余处理路径的概率，又考虑了该路径的时钟周期数，不会产生空闲时隙，比基于RWEP的电压调度算法的节能效果要好很多，而处理的实时性也能得到保证。另外，SA-RWEP根据人为经验设置SDP，虽然可以有效地控制SDP数量，算法相对简单，任务的预处理阶段的开销也较小，然而，缺乏依据的SDP设置方法会造成情景检测过早或不及时，使得情景发生后却无法感知或延迟感知，电压调度优化效果及节能效果有限或无效果。应用本文提出的SDP设置算法能最早地精确感知情景的发生并及时准确地做出路径预测及相应代码与CFG图简化处理，适时调整处理电压，最大程度地节省能量。通过在所有参数情景激活点设置SDP，算法也能感知参数各个情景状态的变化，而加入的较多的情景检测代码也引入了更多的额外开销，如表2所示。然而，当提前检测到的任务剩余工作量的预测值变化较大时，其对节能效果的影响较小。从表中也可以看出，增加参数mode_extension定义情景为应用SD-RAEP的最佳方案，节能性最优；而如增加参数mixed_block_flag，反而会降低节能效果。

表2　SD-RAEP与SA-RWEP的性能比较

定义情景所用参数	情景个数	平均节省能量/(%)		引入开销	
		SA-RWEP	SD-RAEP	SA-RWEP	SD-RAEP
nch	2	13.92	39.33	31	144
nch, block_type	4	14.45	41.42	46	175
nch, block_type, mode_extension	10	15.95	42.81	104	239
nch, block_type, mixed_block_flag	6	14.83	41.15	74	211
nch, block_type, mode_extension, mixed_block_flag	15	15.80	41.96	188	334

4　结　论

在研究了目前存在的任务内电压调度策略后，本文提出基于情景检测的安全任务内电压调度算法，以降低移动嵌入式实时系统的任务处理能耗。仿真实验证实应用该算法能在任务处理过程中及时感知发生的情景，优化地调度处理电压，明显地降低处理能耗而实时性也能得到保证，从而实现系统低功耗运行的目的。

参　考　文　献

[1] YUAN L, QU G. Analysis of energy reduction on dynamic voltage scaling-enabled systems[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2005, 24(12): 1827-1837.

[2] WANG A, CHANDRAKASAN A. Energy-efficient DSPs

for wireless sensor networks[J]. IEEE Signal Processing Magazine, 2002, 19(4): 68-78.
[3] YUAN T, EKICI E. Cross-layer collaborative in-network processing in multihop wireless sensor networks[J]. IEEE Transactions on Mobile Computing, 2007, 6(3): 297-310.
[4] 桂盛霖, 雷航. 能耗限制的松弛任务实时调度算法[J]. 电子科技大学学报, 2008, 37(4): 598-601.
GUI Sheng-lin, LEI Hang. On energy-constrained real-time scheduling for elastic tasks[J]. Journal of University of Electronic Science and Technology of China, 2008, 37(4): 598-601.
[5] SHIN D, KIM J, LEE S. Intra-task voltage scheduling for low-energy hard real-time applications[J]. IEEE Design & Test of Computers, 2001, 18(2): 20-30.
[6] SHIN D, KIM J. Intra-task voltage scheduling on DVS-enabled hard real-time systems[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2005, 24(10): 1530-1549.
[7] SEO J, KIM T, LEE J. Optimal intratask dynamic voltage-scaling technique and its practical extensions[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2006, 25(1): 47-57.
[8] SHIN D, KIM J. Optimizing intratask voltage scheduling using profile and data-flow information[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2007, 26(2): 369-385.
[9] GHEORGHITA S V, BASTEN T, CORPORAAL H. Intra-task scenario-aware voltage scheduling[C]// Proceedings of the 2005 International Conference on Compilers, Architectures and Synthesis for Embedded Systems. New York: ACM, 2005.
[10] LAGERSTROM K. Design and implementation of an MP3 decoder[EB/OL]. [2009-07-13]. http://www.kmlager.com.

编 辑 张 俊

(上接第378页)

[4] BLESSER B, LOCANTIHI B. The application of narrowband Dither operating at the Nyquist frequency indigital systems to provide improvedsSignal to noise ratio over conventional Dithering[J]. Audio Eng, 1987, 35(6): 446-454.
[5] ANNA D. A-D conversion with Dither signal-possiblities and limitations[J]. Measurement Science Review, 2001, 1(1): 75-78.
[6] WAGDY M F, NG W. Validity of uniform quantization error model for sinusoidal signals without and with Dither[J]. IEEE Transaction on Instrumentation and Measurement, 1989, 38(3): 718-722.
[7] SIRAGURSA E, GALTON I. A digitally enhanced 1.8 V 15 bit 40 MSample/s CMOS pipelined ADC[J]. IEEE Journal Solid-State Circuits, 2004, 39(12): 2126-2138.
[8] SHU Y S, SONG B S. A 15 bit linear 20 MSample/s pipelined ADC digitally calibrated with signal-dependent Dithering[J]. IEEE Journal of Solid-State Circuits, 2008, 43(2): 342-350.
[9] BJMSEN J, YRTERDA T. Behavioral modeling and simulation of high-speed analog-to-digital converters using systemC[C]//IEEE International Symposium on Circuit and Systems.[S.l.]: IEEE, 2003, 906-909.
[10]BILHAN E, ESTRADA P C. Behavioral model of pipeline ADC by using simulink[J]. Southwest Symposium on Mixed-Signal Design, 2001, 147-151.
[11] 陈廷乾, 许俊, 朱凯, 等. 高精度流水线A/D转换器误差分析与系统设计[J]. 微电子学, 2008, 38(1): 126-128.
CHEN Ting-qian, XU Jun, ZHU Kai, et al. Error analysis and system design of high-accuracy pipelined A/D converters [J]. Microelectronics, 2008, 38(1): 126-128.
[12] 程梦璋, 景为平. 新型流水线ADC的设计与分析[J]. 电子科技大学学报, 2008, 37(6): 930-933.
CHENG Meng-zhang, JING Wei-ping. Design and analysis of a novelpipelined ADC[J]. Journal of University of Electronic Science and Technology of China, 2008, 37(6): 930-933.

编 辑 张 俊