

医疗监护系统中 WSN 节点的软件设计

明光照, 张大龙, 李 鸥

(解放军信息工程大学信息工程学院, 郑州 450002)

摘要: 提出医疗监护系统中无线传感器网络节点软件设计的一些关键技术, 包括基于 TinyOs 系统的 nesC 语言开发技术、特殊应用场合的 MAC 层设计技术, 以及节点的节能技术。通过实验手段, 验证了采用该软件设计技术的医疗监护系统节点具有较高的数据交互可靠性和低功耗特性。

关键词: 无线传感器网络; nesC 语言; 低功耗

Software Design of WSN Node in Health Care System

MING Guang-zhao, ZHANG Da-long, LI Ou

(Institute of Information Engineering, PLA Information Engineering University, Zhengzhou 450002)

【Abstract】 This paper gives some key technologies in software design for Wireless Sensor Networks(WSN) node in health care system, including technology of nesC language based on TinyOs, design technology of MAC in special applied situation, and technology of energy conservation. It validates the health care system node which uses these design technologies has high reliability of data communication and low power with laboratory works.

【Key words】 Wireless Sensor Networks(WSN); nesC language; low power

1 概述

在医院住院部建立基于无线传感器网络的医疗监护系统使经常需要测量生理指标的人员(比如慢性病人或者老年患者等)能够在护理室随意运动的状态下测量某些常规指标;使医护人员可以对患者进行实时的跟踪和监控,长期地收集人的生理数据,可以及早发现病症,提高人们的生活质量;更重要的是能进行现代化信息管理,提高工作效率;另外将该系统接入因特网,使远离医院等医护机构的病员也随时能够得到必要的医疗监护,并且在必要的时候得到远程医生的咨询指导,比如我国农村的缺医少药地区。

如图 1 所示,系统由无线传感器网络 sink 节点、医疗传感器节点、485 网线和监护设备组成。sink 节点位置固定不变,而各种医疗传感器节点如血糖测量节点、血压测量节点、血氧和脉搏测量节点则是每个病人随身佩戴的。本系统中医疗传感器节点之间没有通信,只在医疗传感器节点和 sink 节点间进行数据传输,传递各种人体生理指标,比如体温、血压、脉搏、血糖、血氧等信息, sink 节点收集的信息通过 485 网线传递给监护设备。该设备用于显示和存储收集到的所有病人的各项生理指标,该设备被设计成一种手持设备,并拥有多种通信模块接口,用于和主机服务器进行通信。在该应用下的无线传感器网络节点的软件设计是整个系统设计的关键部分。

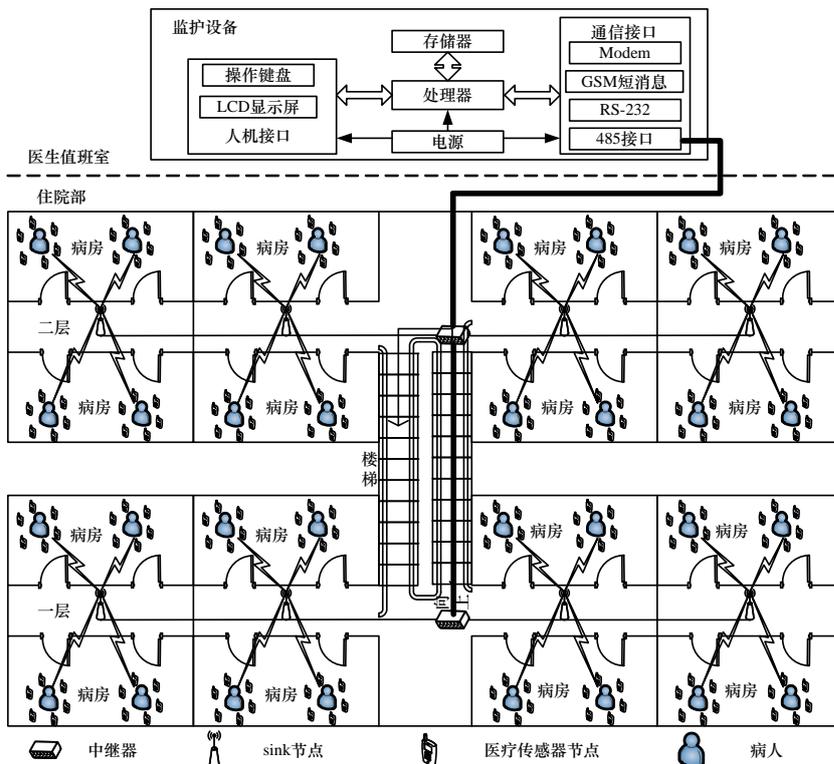


图 1 系统结构

基金项目: 国家发改委 CNGI 基金资助项目“IPv6 无线传感器网络节点”(CNGI-04-10-1D)

作者简介: 明光照(1982—),男,硕士研究生,主研方向:无线传感器网络;张大龙,博士研究生;李 鸥,教授、博士、博士生导师

收稿日期: 2009-09-06 **E-mail:** 8poilkjmn68@163.com

2 节点软件结构

2.1 基于TinyOS的无线传感器网络应用程序开发技术

在无线传感器网络^[1]中,单个传感器节点内存、能量资源有限,处理器、通信设备速度低,又需要进行较复杂的任务调度与管理,系统需要一个微型化的操作系统。为此,UCBerkeley 分校的研究人员设计了 TinyOS 操作系统。TinyOS 操作系统采用的轻量级线程技术、两层调度方式、事件驱动模式、主动消息通信技术及组件化编程等有效地提高了传感器节点 CPU 的使用率,有助于省电操作并简化了应用的开发。TinyOS 系统、库和基于 TinyOS 的应用基本上都是用 nesC^[2]语言开发的。

nesC 是一种适用于开发基于结构化组件应用的新型语言。nesC 是对 C 的扩展,扩展的目的在于将 C 的模块化思想与 TinyOS 的基于事件驱动结合起来。nesC 语言中的 2 个要素是组件和接口,组件又包括配置(configurations)和模块(modules),它们都可以提供和使用接口,组件和接口都以相应的文件形式存在,即存在配置文件、模块文件和接口文件。

2.2 节点软件结构

医疗监护系统节点需要具备以下功能:(1)节点 MAC 层算法。(2)载波侦听和数据包收发功能。其中数据包收发又包括:数据包收发流程功能,收发器初始化功能,处理器和收

发器间 SPI 通信流程功能, SPI 控制功能。(3)随机数产生功能,用于产生指定时间区域内的随机发送点和随机退避时间。(4)串口打印功能,把发送和接收到的数据包信息打印出来用于后续处理。(5)电量指示灯和呼叫护士按键的驱动(仅医疗传感器节点具备)。(6)不同传感器数据采集功能(仅医疗传感器节点具备)。(7)节点通过 485 网线向监护设备传输数据包(仅 sink 节点具备)。

基于 Tinyos 开发技术,需要把上述功能层次化和模块化,把功能分为上层功能和下层功能,不同功能用不同模块表示,模块间用接口连接,接口具体功能的实现在下层模块中完成,上层模块通过接口来调用下层模块的功能。模块的连接关系用配件文件表示。如图 2 所示为项目组设计出的医疗传感器节点基本功能结构图。显示了程序中各组件和接口及其相互连接调用关系。其中实线框表示是模块文件,虚线框表示是配置文件。箭头左边的为上层模块,箭头右边的为下层模块。其中,ACMACM 为节点 MAC 层算法模块,CC1100RadioM 为数据包收发流程模块,CC1100ControlM 为收发器 CC1100 初始化模块,HPLCC1100M 为处理器和收发器间 SPI 通信流程模块,RandomLFSR 为随机数产生模块,KeyLampM 为电量指示灯和呼叫护士按键驱动模块,HPLUARTM 为串口打印模块,其余的为 TinyOs 系统自带模块。

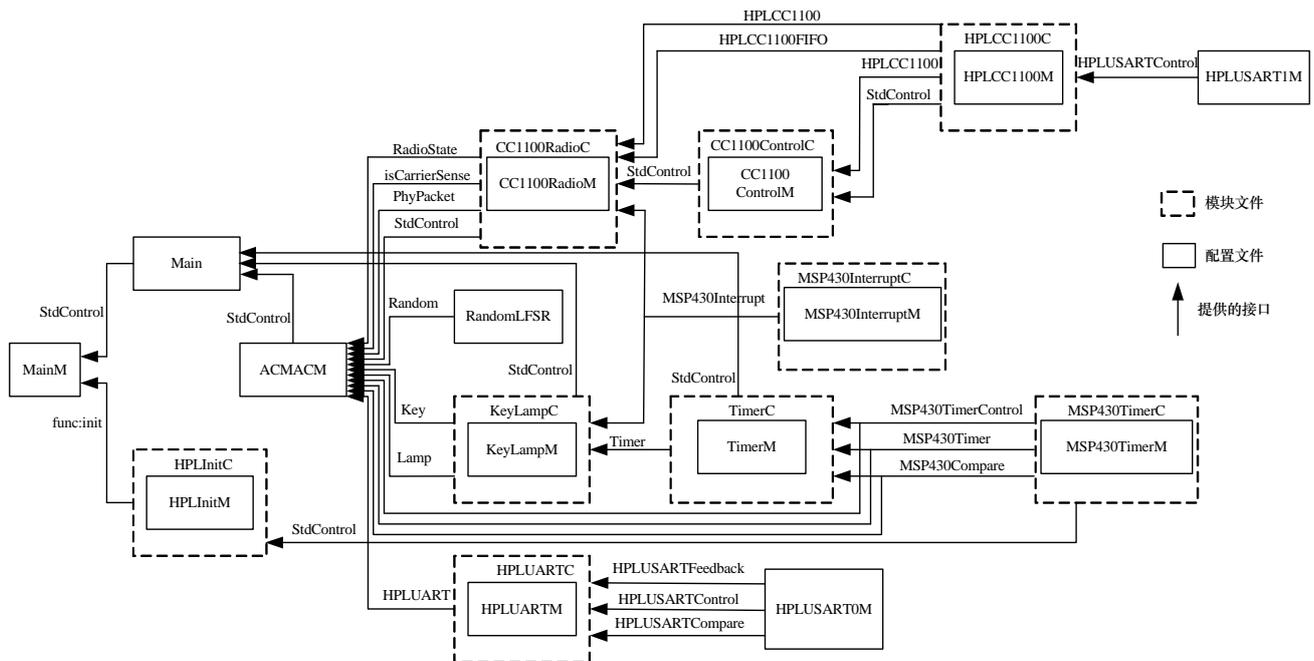


图 2 节点软件结构

3 节点MAC层方案

在一定区域范围内,系统模型如下:有 1~2 个位置固定的 sink 节点,有 80 多个医疗传感器节点,分别编号为 1~80;医疗传感器节点间没有通信,各医疗传感器节点和 sink 节点间有小业务量的数据包收发;医疗传感器节点用电池供电,且要求工作半年以上,sink 节点由 485 网线连接,并由电源供电。

基于此模型的 MAC 层设计需要达到移动节点低功耗、数据包传输可靠性高、时延小的目的,从以下 4 个方面入手:

(1)网络性质。分为全网同步、局部同步和异步网络。由于本系统 MAC 协议总体比较简单,如果采用同步网络,协议复杂度的增加大于其带来的好处,因此选择异步网络。

(2)信道接入方式。本系统采用的是基于 CSMA 竞争机制的信道接入方式。CSMA 又分为 1 坚持 CSMA、p 坚持 CSMA 和非坚持 CSMA。采用坚持 CSMA 与低功耗设计目的相矛盾,且当多个医疗传感器节点同时检测到信道闲时容易产生碰撞,使数据包传输可靠性降低。所以本系统采用随机数退避的非坚持 CSMA。

(3)节能机制。引入休眠机制是无线传感器网络目前广泛应用的有效节能手段。但为确保节点在收发业务到来时及时激活,降低休眠对网络时延、吞吐等性能的影响,节点休眠多长时间及何时激活是其主要问题。发送节点可将业务是否到达作为自身激活的控制依据,但接收节点如何能够在发送节点业务到达时保持及时的激活则成为休眠策略的关键。由

于 sink 节点不需要考虑其节能问题, 可令其一直处于激活状态, 一旦有医疗传感器节点发送数据包, 它能马上接收, 并根据业务需求, 选择立刻回复 ACK 信号或不回复。而移动节点采用在发送完成后在一个很短的时间内保持激活, 确保如果有 sink 节点回复 ACK 信号, 一定能接收到。

(4)冲突避免机制。该机制从以下 2 个方面考虑: 1)简化握手机制, 采用 DATA-ACK 握手机制。2)采用 CSMA 竞争机制。3)离散化发送数据包的时间。移动节点从发送周期 N ms 中随机选择时间点来发送数据包。假设周期为 3 000 ms, 而移动节点从发送数据包开始到接收完 ACK 所需时间为 16 ms, 那么 80 个移动节点发送数据包时, 经计算会产生冲突的概率降低为原先概率的 18%。

图 3 为项目组设计的移动节点 MAC 层算法流程, sink 节点算法较为简单, 这里不再叙述。

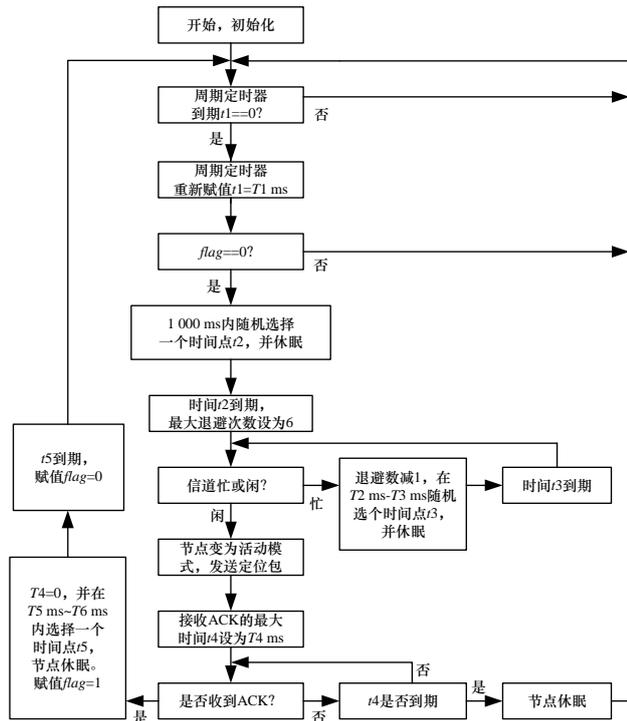


图 3 医疗传感器节点流程

4 医疗传感器节点低功耗软件实现

节点低功耗软件的实现要考虑以下几个方面: 节点晶振的工作方式, 节点处理器模块的能耗优化, 节点射频模块的能耗优化。

4.1 节点晶振的工作方式

用户根据不同的需求, 系统可以使用不同时钟信号 ACLK, MCLK, SMCLK 来选择低频或者高频。这 3 种不同频率的时钟输出给不同的模块, 从而合理地利用系统电源。当节点休眠时, SMCLK, MCLK 均选用 DCO 振荡器, ACLK 选择外接低频晶振。这样高频晶振空闲, 便可以将其关掉以节省功耗, 并把 ACLK 时钟输出给系统时钟模块; 当需要进行数据收发时, 先打开外接高频晶振, 再让 SMCLK 和 MCLK 都选择高频晶振, ACLK 仍选择外接低频晶振, 并把 SMCLK 时钟输出给高速 SPI 通信模块, 系统时钟模块此时改用 MCLK, 这样做是因为低频时钟用于计时, 产生的时钟中断比高频时钟产生的时钟中断长, 这样会影响 SPI 的高速通信, 造成 SPI 通信延时等问题。所以只有休眠时系统时钟模块才选择 ACLK 时钟。

4.2 节点处理器模块的能耗优化

本系统以 MSP430^[3]为例, 介绍节点处理器模块的能耗优化。当系统时钟发生器基本建立之后, CPU 内状态寄存器 SR 中的 SCG1, SCG2, OscOFF 和 CPUOFF 位是重要的低功耗控制位。只要任意中断被响应, 上述控制位就被压入堆栈保护, 中断处理之后, 又可恢复先前的工作方式。控制位 SCG1, SCG2, OscOFF, CPUOFF 可由软件配置成 6 种不同工作模式: 1 种活动模式和 5 种低功耗模式。通过设置控制位 MSP430 可从活动模式进入到相应的低功耗模式, 而各种低功耗模式又可以通过中断方式回到活动模式。正好符合 TinyOS 系统的特点, 整个系统因事件驱动(各种中断)而运行, 如果没有事件发生, 则系统进入设定的休眠状态, 从而达到节能的目的。

4.3 节点射频模块的能耗优化

本系统以 CC1100 芯片为例, 介绍射频模块的能耗优化。CC1100 有发送、接收、空闲、休眠等状态, 图 4 为 CC1100^[4]状态转换和相关的程序设计。

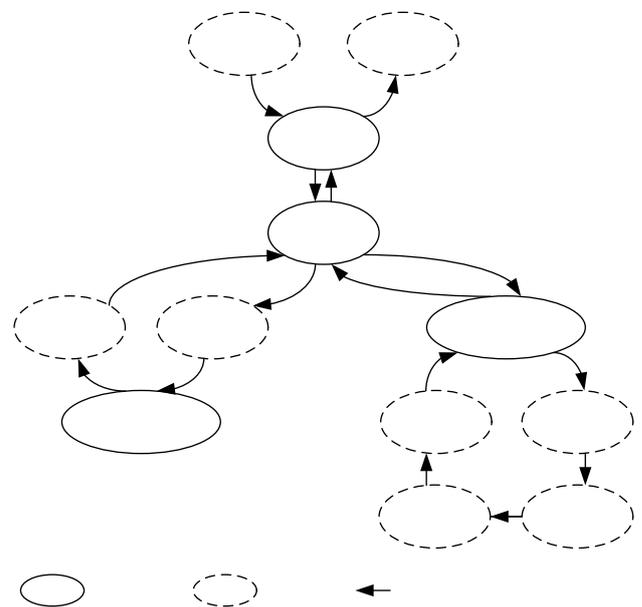


图 4 CC1100 状态转换图

程序流程如下: (1)先打开高频晶振。(2)把 CSn 管脚置低, 这样 CC1100 就会从休眠中激活。(3)MSP430 通过 SPI 向 CC1100 传输发送数据。(4)MSP430 通过 SPI 向 CC1100 发送控制指令 STX, 让 CC1100 进入发送状态。(5)当 CC1100 的 GDO2 管脚有上升延产生时, 表明数据发送完毕。(6)MSP430 通过 SPI 向 CC1100 发送控制指令 SFTX, 让 CC1100 回到空闲状态。(7)MSP430 通过 SPI 向 CC1100 发送控制指令 SRX, 让 CC1100 进入接收状态。(8)发送完成的相关程序处理, 包括发送数据串口打印、设置最大接收时间等。该步骤放在步骤(6)、步骤(7)之后是因为打开高频晶振以后, 会有一段稳定时延, 造成收发程序整体时延, 所以在发送完成后就需要马上切换到接收状态, 否则会由于时延耽误没有及时切换到接收状态, 漏掉了 ACK 的接收。(9)当 CC1100 的 GDO2 管脚有上升延产生时, 表明数据接收完毕。(10)MSP430 通过 SPI 从 CC1100 读取接收数据。(11)接收完成后相关程序处理。包括串口打印、设置休眠时间等。(12)MSP430 通过 SPI 向 CC1100 发送控制指令 SFRX, 让 CC1100 回到空闲状态。(13)MSP430 通过 SPI 向 CC1100 发送控制指令 STX, 让

CC1100 进入休眠状态。(14)关闭高频晶振。

5 软件性能测试

5.1 可靠性性能测试

用 1 个区域 sink 节点和 80 个医疗传感器节点组成一个传感器网络。把 sink 节点发送和接收到的数据通过串口打印出来。数据包信息包括：数据包序号，通过它可以知道发送或接收数据包的总数和遗漏数据包个数；数据包类型号，可以知道是 data 数据包还是 ACK 数据包；节点序号，可以知道是哪个节点的数据包。再分别计算出错包率、丢包率。计算公式如下：

$$\text{错包率} = \frac{\text{错误分组总数}}{\text{错误分组总数} + \text{正确分组总数}}$$

$$\text{丢包率} = \frac{\text{成功发送的分组总数} - \text{正确接收的分组总数}}{\text{节点成功发送的分组总数}}$$

计算得到平均错包率约在 3‰左右，总的平均丢包率约在 5‰左右。

5.2 低功耗性能测试

将电表的 2 个表笔接在电池正极与医疗传感器节点电路板电源正极之间：让医疗传感器节点一直处于发送状态，测得发送时电流值为 20.90 mA；让移动节点一直处于接收状态，测得接收时电流值为 15.76 mA；让移动节点保持在休眠状态，测得休眠时电流值为 0.8 μA。而后让节点工作在休眠唤醒模式下，用示波器探头观察节点的射频晶振，可测得移动节点在一个周期 3 000 ms 中，处于发送状态的时间为 8 ms，处于接收状态的时间为 9 ms，其余时间处于休眠。这样可用下式得到节点在一个工作周期内的平均电流为 98.56 μA：

$$I_{avg} = \frac{I_{lpm3} \times t_{lpm3} + I_{send} \times t_{send} + I_{rev} \times t_{rev}}{t_{lpm3} + t_{send} + t_{rev}}$$

其中， I_{lpm3} 、 I_{send} 和 I_{rev} 分别是处理器在 $lpm3$ 模式、发送模式和接收模式状态下的电流。则使用一块容量为 900 mAh 的电池，考虑到系统长时间运行中电池的自放电因素，并为系统留有余量，假设电池可用电量为 60%，计算公式如下：

$$T = \frac{900 \times 1000 \times 60\%}{98.56 \times 24} = 228 (\text{天})$$

6 结束语

目前国际上对远程集中医疗的关注越来越强，而该应用下的无线传感器网络节点的软件设计是系统开发的主要部分。测试结果显示：利用本软件构架，节点的可靠性良好，总的错包与丢包率都较小；而且移动节点能耗极低，可用时长达 200 天以上。以上均达到了医疗监护系统的实际应用需求。

参考文献

- [1] 孙利民, 李建中. 无线传感器网络[M]. 北京: 清华大学出版社, 2005.
- [2] Gay D, Levis P, Culler D, et al. NesC 1.1 Language Reference Manual[EB/OL]. (2005-03-30). <http://www.tinyos.net/>.
- [3] 沈建华, 杨艳琴, 翟晓曙. MSP430 系列 16 位超低功耗单片机原理与应用[M]. 北京: 清华大学出版社, 2004.
- [4] 孙武剑. 基于 TinyOS 的 CC1100 驱动程序设计[J]. 计算机工程, 2008, 34(23): 95-97.

编辑 任吉慧

(上接第 252 页)

时，其运算时间应当包括数据的存储时间和读出时间，因此，本文提出的并行加法器的运算速度比超前进位加法器更快。

		Destination				
	c4	s1	s2	s3	s4	
a1	9.5ns/9.6ns	6.0ns	6.0ns/9.5ns	6.0ns/9.5ns	6.0ns/9.5ns	
a2	9.5ns/13.1ns		9.5ns	6.0ns/9.5ns	6.0ns/9.5ns	
a3	9.5ns/13.1ns			9.5ns	6.0ns/9.5ns	
a4	9.5ns/13.1ns				9.5ns	
b1	9.5ns/9.6ns	6.0ns	6.0ns/9.5ns	6.0ns/9.5ns	6.0ns/9.5ns	
b2	9.5ns/13.1ns		9.5ns	6.0ns/9.5ns	6.0ns/9.5ns	
b3	9.5ns/13.1ns			9.5ns	6.0ns/9.5ns	
b4	9.5ns/13.1ns				9.5ns	
c0	6.0ns	6.0ns	6.0ns	6.0ns	6.0ns	

图 3 超前进位加法器时延结果界面

		Destination						
	c56	c66	c77	c88	c99	s0	s1	
a0					17.5ns/36.5ns	17.5ns	17.5ns/26.5ns	
a1				12.5ns/16.5ns	12.5ns/37.5ns		27.5ns	
a2	7.5ns			12.5ns/17.5ns	17.5ns/37.5ns			
a3		7.5ns/11.5ns	7.5ns	12.5ns/17.5ns	21.5ns/32.5ns			
b0					17.5ns/36.5ns	17.5ns	17.5ns/26.5ns	
b1				12.5ns/16.5ns	12.5ns/37.5ns		27.5ns	
b2	7.5ns			12.5ns/17.5ns	17.5ns/37.5ns			
b3		7.5ns/11.5ns	7.5ns	12.5ns/17.5ns	21.5ns/32.5ns			
c0					17.5ns/32.5ns	17.5ns	17.5ns/22.5ns	
c00					12.5ns/27.5ns	17.5ns	12.5ns/21.5ns	
c1				12.5ns/16.5ns	12.5ns/37.5ns		27.5ns	
c2	7.5ns			12.5ns/17.5ns	17.5ns/37.5ns			
c3		7.5ns	7.5ns	12.5ns/17.5ns	21.5ns/32.5ns			
c11					12.5ns/27.5ns	17.5ns	12.5ns/21.5ns	
c22					17.5ns/27.5ns		17.5ns	

图 4 本文提出的加法器时延结果界面

5 结束语

本文的并行加法器能完成 5 个加数的逻辑计算，通过扩展其 5 加数并行加法电路，可以解决 5 个多位二进制加数的相加问题。如果将该方案应用到新的微处理器设计中，能极大提高微处理器的运算速度，使其得到进一步优化，从而推动微处理器设计理念的变革。

参考文献

- [1] 王爱英. 计算机组成与结构[M]. 4 版. 北京: 清华大学出版社, 2007.
- [2] Sadrossadat S A, Amiri N K, Fakhraie S M. An Efficient Multi-operand Addition Structure[C]//Proc. of International Conference on Microelectronics. Cairo, Egypt: [s. n.], 2007: 73-76.
- [3] Kornerup P. Reviewing 4-to-2 Adders for Multi-operand Addition[C]//Proc. of IEEE International Conference on Application-Specific Systems, Architectures and Processors. California, America: [s. n.], 2002: 218-229.
- [4] 刘杰. 多二进制数同步超前进位相加符号和溢出性研究[J]. 昆明理工大学学报: 理工版, 2004, 29(5): 83-86.
- [5] Schulte M J, Chirca K. A Low-power Carry Skip Adder with Fast Saturation[C]//Proc. of the 15th IEEE International Conference on Application-specific Systems, Architectures and Processors. Galveston, TX, USA: [s. n.], 2004: 269-279.
- [6] Huang Yen-Mou, James B K. A High-speed Conditional Carry Select Adder Circuit with Asuccessively Incremented Carry Number Block Structure Forlow-voltage VLSI Implementation[J]. IEEE Trans. on Circuits and Systems, 2000, 47(10): 1074-1079.
- [7] Manzoul M A. Parallel CLA Algorithm for Fast Addition[C]//Proc. of Int'l Conf. on Parallel. Quebec, Canada: [s. n.], 2000: 55-58.
- [8] 李云锋, 赵金薇, 周汇, 等. 一种基 2 冗余符号数加法的改进算法[J]. 计算机工程, 2007, 33(24): 242-246.
- [9] 张志伟, 马鸿, 李立健, 等. VLIW 数字信号处理器 64 位可重构加法器的设计[J]. 计算机工程, 2007, 33(16): 29-31.

编辑 陈晖

