

DOI:10.3969/j.issn.1001-4551.2013.06.028

基于可扩展标准单元的半定制电路设计方法*

李碧琛, 沈海斌*, 郑丹丹, 严晓浪

(浙江大学 超大规模集成电路研究所, 浙江 杭州 310027)

摘要: 针对芯片设计中关键路径优化不足的问题,提出了基于可扩展标准单元的半定制电路设计方法。采用逻辑功效模型分析了关键路径,根据分析结果,通过使用Cadance工具的开发语言编程设计了具有完备驱动能力的扩展单元集,通过使用hspice工具仿真扩展单元的时序,通过使用逻辑功效优化算法将关键路径中延时较大的标准单元替换成具有理想驱动能力的扩展单元从而减少了单元延时,以获得最短路径延时,实现了设计流程自动化,获得了良好的工艺可移植性和设计灵活性;以处理器芯片为实验电路,采用台积电的4种工艺,分别使用扩展单元和标准单元完成了芯片的物理设计和时序仿真。实验结果表明,主流工艺下基于可扩展标准单元的半定制设计方法可以充分优化关键路径和缩短延时,有效地提升电路的主频。

关键词: 芯片设计; 可扩展标准单元; 关键路径优化; 逻辑功效模型; 半定制设计

中图分类号: TN4; TP3; TH39 文献标志码: A

文章编号: 1001-4551(2013)06-0754-05

Semi-custom design method based on standard cell extension

LI Bi-chen, SHEN Hai-bin, ZHENG Dan-dan, YAN Xiao-lang

(Institute of VLSI Design, Zhejiang University, Hangzhou 310027, China)

Abstract: Aiming at problem of insufficient optimization for critical path in VLSI design, a semi-custom design method based on standard cell extension was proposed. Critical path was analyzed on the basis of logical effort model. Extended cells with perfect driving capability were constructed by development language of Cadance according to analysis results, and timing information of extended cell might be simulated by hspice. Standard cells in optimal critical path with long timing-delay could be replaced by extended cells with ideal drive ability in order to achieve shortest timing-delay both of the cell itself and critical path, meanwhile design automation and flexibility could be realized and the very method had advantageous portability for mainstream technology. Physical design and timing simulation by four kinds of TSMC technology for a certain process chip were accomplished respectively with standard cell and extended cell method. The experimental results indicate that the semi-custom design methodology based on standard cell extension could shorten the critical path delay and improve circuit frequency effectively in mainstream technology design.

Key words: VLSI design; extended standard cell; optimization for critical path; logical effort model; semi-custom design

0 引言

随着半导体技术的飞速发展,具有大量数据处理能力的高性能处理器芯片在嵌入式领域有着广泛的应用需求。处理器芯片支撑着整个嵌入式系统的运作,其性能好坏直接决定系统的整体性能。因此,在技术和市场需求的双重推动下,设计出具有高性能的处理

芯片已经成为一种必然趋势。评价处理器芯片性能的主要指标为处理器的主频,而在设计工作中关键路径的延时决定了电路主频,是性能提升的瓶颈^[1]。通过优化关键路径提高电路性能已成为当今的研究热点。

设计高性能芯片的常用方法有两种^[2]:全定制方法和基于标准单元的半定制方法。全定制方法通过精确设计CMOS管的宽长比和布局布线,减少了寄生

收稿日期: 2012-12-12

基金项目: 国家科技重大专项“核高基重大专项”资助项目(2009ZX01030-001-002)

作者简介: 李碧琛(1987-),男,黑龙江哈尔滨人,主要从事嵌入式处理器与SOC设计方面的研究. E-mail: libc@vlsi.zju.edu.cn

通信联系人: 沈海斌,男,博士,教授,博士生导师. E-mail: shb@vlsi.zju.edu.cn

参数和串扰对电路性能的影响,极大地提高了晶体管的开关速度。然而,集成电路规模和复杂度的大幅提高以及市场竞争的日益激烈等客观形势,对产品研发的周期和成本提出了严峻的挑战。尽管全定制设计方法可以实现最短路径延时和最优性能,但需要投入大量人力和时间成本^[3-4],而基于标准单元的半定制设计方法兼顾了产品的性能和研发时间。该方法依托于 Foundry 提供的标准单元库和 EDA (Electronic Design Automation) 工具,可以实现设计自动化。但是该方法受限于标准单元库中单元有限的数量和驱动能力,对关键路径优化不足,无法实现最短延时。

目前,国内外对采用标准单元设计中的关键路径优化方法研究比较少,文献[5]根据 AIG (And-Inverter Graph) 和 BDD (Binary Decision Diagrams),将与非门和反相器组成具有特定复杂逻辑功能的单元,将生成的单元扩充到标准单元库中进行电路设计。由于与非门和反相器的良好特性,复杂逻辑单元不仅能够实现指定功能,还具有较短的单元延时和较小的面积,因此使用复杂逻辑单元设计的电路在主频和面积上均有较好地提升。但该方法的复杂逻辑单元需要专门手工全定制设计,且每种单元只能实现单一功能,设计成本很高,不适合量产。文献[6]提出了高性能标准单元库设计,针对常用单元,使用或非门和反相器设计出具有多种驱动能力的单元集并给出了单元建库的验证方法,以一定的面积代价换取了电路性能的提高。但该方法需要针对不同工艺重新设计单元库,而且由于或非门的特性,其设计出的单元并不能明显改善电路性能。文献[7]提出了基于逻辑功效模型的关键路径优化方法,将两个同类型标准单元扩展成一个新单元并且具有多种驱动能力。同时根据路径逻辑功效分析结果,在逻辑综合阶段选择将延时较大单元替换成具有最佳驱动能力的扩展单元。该方法对电路主频的提升较明显,但由于扩展单元是由同类标准单元并联拼接组成,不但消耗了很多面积资源,而且扩展单元输入电容为两个单元之和,单元延时会成比例增加,对于单元本身性能的提升效果并不理想。

在逻辑功效模型优化关键路径方法的基础上,本研究提出基于可扩展标准单元的半定制设计方法。由于大部分标准单元以反相器作为输出端,该方法将标准单元和面积很小的反相器组合成扩展单元,构建具有更完备、更大驱动能力的扩展单元集合。该方法的优势是,可以根据设计需求对标准单元任意扩展,既提升了单元本身性能,又以很小的面积代价提高了电路整体的主频,解决了关键路径优化不足的问题。可扩展标准单元的全部设计流程由 EDA 工具自动化

实现,大大减少了设计时间和成本。基于可扩展标准单元的半定制设计方法由以下几个步骤组成:

- (1) 关键路径的逻辑功效模型分析;
- (2) 根据分析结果扩展标准单元并验证;
- (3) 将扩展单元加入设计流程中并优化关键路径。

1 逻辑功效模型和关键路径优化

1.1 关键路径的逻辑功效分析

在芯片设计中,关键路径的延时是影响芯片性能的主要因素,引入逻辑功效理论^[8]分析关键路径,路径中单个门延时可用线性模型的基本延时公式表示为:

$$t_p = t_{p0}(p + gf/\gamma) \quad (1)$$

式中: t_{p0} —反相器的本征延时; p —该门本征延时与基本反相器本征延时之比; g —逻辑功效,表示为复合门和基本反相器提供相同输出电流时的输入电容比; f —等效扇出; γ —门本征电容和输入电容之比。

关键路径延时为各门延时之和,表示为:

$$t_p = \sum_{j=1}^N t_{pj} = t_{p0} \times \sum_{j=1}^N (p_j + \frac{f_j \times g_j}{\gamma}) \quad j=1, 2, \dots, N \quad (2)$$

式中: t_{p0} , p_j , γ , g_j —常数; $f_j \times g_j$ 定义为门功效。

根据逻辑功效理论,当关键路径中每一级门单元门功效相等时路径延时最短。本研究遍历标准单元库中单元驱动能力种类,通过调整每一级单元的驱动能力,实现各级门功效相等从而得到最短路径延时^[9]。

关键路径实例如图1所示。cell1 在标准单元库中的驱动能力有 X1, X2 和 X4, 为保证路径每一级单元逻辑功效相等, cell1 需要选择的驱动能力为 X3, 而标准单元库中只能选择驱动能力为 X2 或者 X4 的单元替代,无法实现最短延时。与之类似,若 cell2 的下一级 cell3 需要的驱动能力为 X6, 但标准单元库中 cell3 最大驱动能力为 X4, 驱动能力不足也会产生很大的单元延时,无法实现路径最优化。对以上两种情形,研究者根据设计需求对标准单元库进行扩展,可实现更完备、驱动能力更大的单元集合,使关键路径延时最短。

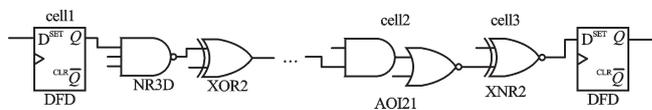


图1 关键路径实例

1.2 关键路径优化算法

对图1的关键路径建立逻辑功效模型,当每一级门功效相等时,即 $f_1 g_1 = f_2 g_2 = f_3 g_3 = \dots = f_n g_n$ 。路径总逻辑功效为:

$$G = \prod_{i=1}^N g_i = g_1 \times g_2 \times \dots \times g_n, i=1, 2, 3, \dots, N \quad (3)$$

路径有效扇出为:

$$F = \frac{C_L}{C_{g_1}} \quad (4)$$

式中: C_L —路径最后一级负载, C_{g_1} —路径第一级输入电容。

路径上单个逻辑门的分支功效 b 定义为:

$$b = \frac{C_{on-path} + C_{off-path}}{C_{on-path}} \quad (5)$$

式中: $C_{on-path}$ —该门沿路径上的负载电容, $C_{off-path}$ —离开路径上的连线电容。

如果路径无分支,则分支功效为 1。路径总分支功效为:

$$B = \prod_1^N b_i = b_1 \times b_2 \times \dots \times b_N, i = 1, 2, 3, \dots, N \quad (6)$$

路径总有效扇出为:

$$F = \prod_1^N \frac{f_i}{b_i} = \frac{f_1 \times f_2 \times \dots \times f_N}{b_1 \times b_2 \times \dots \times b_N} = \frac{\prod_1^N f_i}{B} \quad (7)$$

路径总功效为:

$$H = \prod_1^N h_i = \prod_1^N g_i \times f_i = GFB, i = 1, 2, \dots, N \quad (8)$$

使路径延时最小的门努力为 $h = \sqrt[N]{H}$, 此时路径总延时为:

$$t_p = \sum_{j=1}^N t_{p,j} = t_{p0} \times \left(\sum_{j=1}^N P_j + \frac{N(\sqrt[N]{H})}{\gamma} \right), j = 1, 2, \dots, N \quad (9)$$

对于关键路径,优化算法如下:求路径上所有单元的总功效延时 H ,得到路径延时最短时每一级单元的最优门功效 h_{best} 。对于路径上的每一个单元,遍历扩展单元库中同类型的单元,如果满足“替换单元的门功效和最优门功效 h_{best} 更接近”则执行替换操作。

其中: `get_critical_path` 函数返回设计中的关键路径, `get_input_capacitance`、`get_output_capacitance`、`get_branch_effort`、`get_logical_effort` 函数分别返回关键路径的输入电容、负载电容、分支功效和逻辑功效; `get_cell_fanout` 函数返回路径中单元扇出; `get_same_type_cell(lib, cell)` 函数返回 lib 中同类 cell 单元; `get_tmp_cell_fanout(cell, tmp_cell)` 函数将 cell 用 tmp_cell 替换,返回 tmp_cell 的扇出后取消替换操作;以上函数均由设计者自己定义。`size_cell(cell, tmp_cell)` 执行对 tmp_cell 对 cell 的替换操作,数学函数 `square(H, n)` 返回 H 的 n 次方根, `abs(x)` 返回 x 的绝对值:

```
Path_optimization(design, lib)
critical_path=get_critical_path(design);
input_cap=get_input_capacitance(critical_path);
output_load=get_output_capacitance(critical_path);
F=output_load/input_load;
B=1;G=1;n=0;
```

```
foreach cell in critical_path do
g=get_logical_effort(cell);b=get_branch_effort(cell);
G=G*g;B=B*b;n=n+1;
enddo
H=F*G*B;
h_best=square(H,n);
foreach cell in critical_path do
g=get_logical_effort(cell);
f_best[cell]=h_best/g;
f[cell]=get_cell_fanout(cell);
lib_cells=get_same_type_cell(lib,cell);
foreach tmp_cell in lib_cells do
f=get_tmp_cell_fanout(cell,tmp_cell);
if[abs(f-f_best[cell])<abs(f[cell]-f_best[cell])] then
size_cell(cell,tmp_cell)
endif
enddo
enddo
```

以上程序算法在逻辑综合和 STA (static timing analyse) 阶段使用,通过分析关键路径中所有单元的门功效,搜索每一级单元在扩展单元库中拥有不同驱动能力的同类单元,找出门功效最接近 h_{best} 的单元,从而使得关键路径每级门功效接近相等以实现最短延时和最优性能。而可扩展标准单元库的设计是关键路径优化算法实现的基础。

2 可扩展标准单元库设计

在一般标准单元库中,大部分标准单元是由 CMOS 组合逻辑和反相器驱动输出端组成,所以反相器的驱动能力就是标准单元的驱动能力。扩展单元通过改变单元反相器的宽长比来改变反相器的驱动能力,扩充为具有不同驱动能力的扩展单元。根据逻辑功效理论,单元输入电容不变负载变化时均有唯一尺寸反相器使单元延时最小。反相器的相对尺寸可由逻辑功效计算得出。可将标准单元和反相器拼接得到扩展单元。可以根据设计需要灵活地将已有单元扩展成所需驱动能力的单元,配合关键路径优化算法实现最短延时。

2.1 扩展单元的版图实现

扩展单元版图实现是扩展单元库设计中最重要的一步。本研究以 tsmc65 nm GP 工艺为例,使用 Cadence virtuoso 工具,将工艺库中标准单元和反相器任意拼接,得到不同驱动能力扩展单元。整体流程通过 skill 语言编程自动化实现。skill 是 Cadence 公司提供的二次开发操作语言,可以实现 virtuoso 所有图形界面的操作。使用 skill 语言编程,自动化实现扩展单元的流程如图 2 所示。

在上述流程中,函数 `dbNewCellView` 创建了一个新的单元视图,函数 `dbCreateInst` 创建单元实例,函数

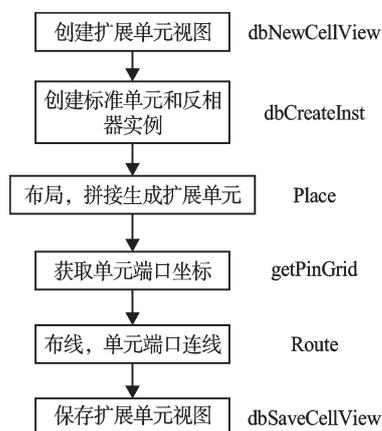


图2 扩展单元自动化实现流程

Place通过设置参数,并将创建单元和反相器拼接从而生成扩展单元。函数getPinGrid获取扩展单元端口坐标。函数Route实现布线,dbSaveCellView保存扩展单元的单元视图。

为了实现设计自动化,本研究将标准单元设计成高度相等但宽度不等。为了便于工具综合,扩展单元也保持相同的高度。反相器输出端单元扩展实现如图3所示,将cell_y和反相器INV拼接。假设cell_y有A1、A2、A3、ZN 4个端口;A为cell_y的反相器信号输入点,INV有I、ZN两个端口。将cell_y放置在原点,方向为R0,将INV紧贴在cell_y的右侧,方向为R180,Y轴镜像,扩展成新的单元cell_y1;将cell_y的A和反相器I端相连,将cell_y的ZN端和反相器ZN端相连;原cell_y的端口A1、A2、A3、ZN为扩展单元cell_y1的端口。

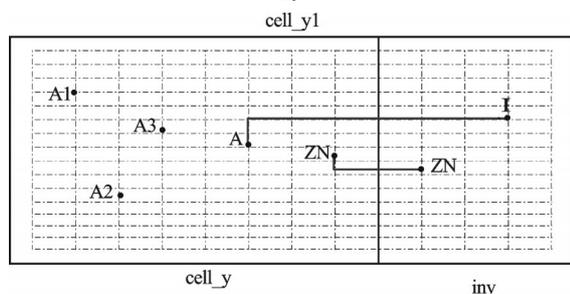


图3 反相器输出端单元扩展实现

由于要处理大量的单元生成扩展单元库,本研究在版图实现中引入基于布线通道的自动布线方法。依据工艺文件中metal2的最小间距定义布线通道,如图3所示:虚线表示布线通道,标准单元设计使得端口处在metal2的布线通道上,问题抽象到有限空间内两点之间互联问题。采用李氏算法^[10],使用skill语言编程实现,自动布线流程图如图4所示。

为了便于在不同工艺状况下实现扩展单元版图自动化,本研究在skill程序中定义了大量子函数,主函数只需调用子函数即可,并将与工艺相关参数如金属最小线宽和间距、通孔大小等单独定义在一个初始

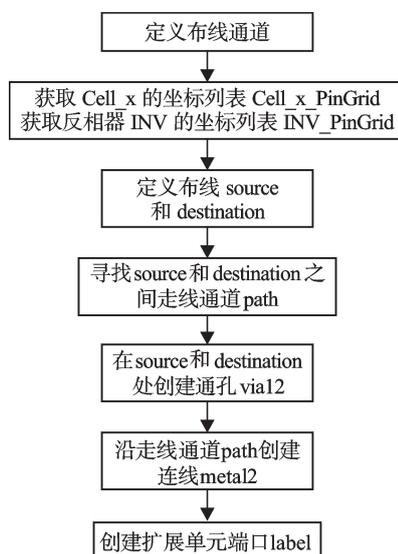


图4 自动布线流程图

化函数中。只需修改初始化函数即可将设计方法移植到不同工艺上。

2.2 扩展单元物理验证和特征化

为确保功能的正确性,需要对扩展单元进行物理验证。验证包括3个方面:设计规则检查(DRC, design rule check)、天线效应检查(Antenna rule check)、版图一致性检查(LVS, layout vs schematic)。自动化生成的扩展单元均能通过物理验证,因此可对扩展单元进行特征化。特征化即提取单元时序信息,生成后端设计所需的库文件,包括输入/输出延时、输出转换时间(output transition)、输入端口电容、面积和功耗等特征参数的提取。

由于扩展单元使用metal2和通孔via12连接端口,会引入寄生参数,而寄生电阻电容会影响负载电容充电时间,进而影响延时。为了提取完整准确的特征化信息,必须对版图进行RC寄生参数提取。针对rcworst、rcbest、cbest、cworst、typical 5种corner,本研究使用synopsys的StarRCXT工具分别提取,得到包含寄生参数的spice网表。通过使用hspice工具仿真网表得到时序信息,即输入端到输出端延时(delay)和输出转换时间(output transition)。根据非线性延时模型(NLDM, non-linear delay model)可知,cell delay和output transition是输入转换时间(input transition)和输出负载(output load)的分段线性函数。本研究对输入端施加激励,输出端施加电容负载得到仿真波形,通过测量输入端到输出端的延时和输出转换时间,得到单元时序信息,从而生成扩展单元标准时序库文件^[11]。

3 实验结果与电路分析

由于处理器在嵌入式应用中发挥着重要作用,基

于上述理论分析和仿真验证,本研究以嵌入式CPU作为实验电路,对上文提出的电路设计方法进行实物验证。笔者分别设计了 tsmc55 nm GP、tsmc65 nm GP、tsmc90 nm 和 tsmc130 nm 工艺的扩展单元库,并实现同一嵌入式CPU的设计,然后采用逻辑功效优化算法对CPU的关键路径进行优化,并测试不同工艺下电路主频的提升。

3.1 嵌入式CPU的物理设计

以 tsmc65 nm GP 工艺为例,本研究在该单元库的支持下,使用综合工具 Module Compiler 分别实现基于标准单元库和扩展单元库的嵌入式CPU门级网表。为了准确评估基于扩展单元的设计方法对CPU主频的提升,在布图规划阶段使用较大的设计面积以忽略面积因素的影响。本研究中,笔者使用 Synopsys 的 ic compiler 工具对CPU的两个门级网表进行布局布线,并生成CPU版图;使用 Synopsys Star RCXT 工具分别对CPU版图进行寄生参数提取;使用 Synopsys 的 Primetime 工具将提取的寄生参数反标到门级网表上,对嵌入式CPU进行静态时序分析。根据时序分析结果使用逻辑功效模型优化算法对CPU关键路径进行

优化,对关键路径上延时较大的单元调整驱动能力,达到延时最优化。采用扩展单元的CPU设计版图如图5所示。

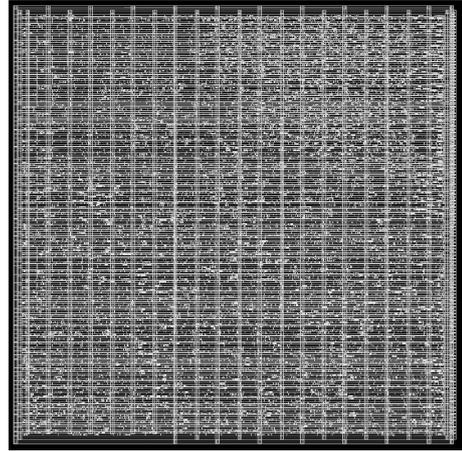


图5 tsmc65 nm GP工艺嵌入式CPU版图

3.2 CPU电路的仿真结果对比

对于同一个CPU代码,本研究在不同工艺下分别采用标准单元库和扩展单元库设计,根据 ic compiler 对单元面积的统计和 Primetime 的时序分析结果,实现的嵌入式CPU主频和面积数据如表1~4所示。

表1 tsmc55 nm GP工艺CPU性能数据

库类型	门面积总和/ μm^2	关键路径延时/ns	频率/MHz	主频提升/(%)	面积增加/(%)
标准单元库	33 597	1.144 2	874		
扩展单元库	3 284	1.039 5	962	10.07	7.99

表2 tsmc65 nm GP工艺CPU性能数据

库类型	门面积总和/ μm^2	关键路径延时/ns	频率/MHz	主频提升/(%)	面积增加/(%)
标准单元库	383 674	1.273 9	785		
扩展单元库	412 805	1.154 7	866	10.32	7.60

表3 tsmc90 nm工艺CPU性能数据

库类型	门面积总和/ μm^2	关键路径延时/ns	频率/MHz	主频提升/(%)	面积增加/(%)
标准单元库	758 214	1.661 1	602		
扩展单元库	812 805	1.497 0	668	10.96	7.19

表4 tsmc130 nm工艺CPU性能数据

库类型	门面积总和/ μm^2	关键路径延时/ns	频率/MHz	主频提升/(%)	面积增加/(%)
标准单元库	1 441 107	1.883 2	531		
扩展单元库	1 563 928	1.680 7	595	11.2	8.52

由表1~4可知,不同工艺下采用扩展单元库实现的嵌入式CPU比标准单元库实现的CPU主频提升了10%左右,面积只增加了8%左右。由于扩展单元是由标准单元和反相器拼接而成,扩展单元库实现的CPU面积会略微增大。实验结果证明基于可扩展标准单元的设计方法适用于所有工艺,有很好的移植性,研究者能够利用很小的面积开销显著地提升电路的主频性能。

4 结束语

依据逻辑功效理论优化芯片设计中的关键路径,

本研究提出了基于可扩展标准单元的新型半定制电路设计方法。经实验验证,该方法能够有效地提高电路的主频,实现设计自动化,大大减少研发成本,缩短研发周期,对于规模复杂有高负载的电路,这种方法更能体现其优势。通过该方法,可根据不同的设计需求对标准单元库进行任意扩展,从而实现驱动能力更大、更完备的单元集合,对设计高性能芯片、提高嵌入式系统性能有着重要的指导意义。

(下转第763页)

位算法,相比普通梯形图实现方法,减少了代码量,并占用更少的数据内存空间。此外,运动控制模块具有更好的可读性。

表2 绝对运动控制算法的普通梯形图和运动控制模块实现方法对比

实现方式	梯形图大小 /KB	数据内存空间 /B	可读性
运动控制模块	11	16	精简,可读性较好
梯形图	78	19	冗长,可读性一般

6 结束语

以DSP-PLC架构的电梯门机控制系统为基础,笔者提出了一种基于IEC 61131-3标准的运动控制模块的设计与研究方法,并对分别采用运动控制模块实现和梯形图实现的绝对定位算法进行了对比分析。实验结果表明:采用运动控制模块的方法具有实现简单、可读性好、生成代码量小等优点。

在下一步的研究中,可以设计更复杂的符合IEC61131-3标准的运动控制模块,从而方便工业上的运动控制过程。

参考文献(References):

- [1] 张丕鑫. 基于PLCopen的运动控制功能块研究与实现[D]. 大连:大连理工大学机械工程学院,2011.
- [2] 田永中. 开放式数控系统中G代码编译器的设计与研究[D]. 乌鲁木齐:新疆大学机械工程学院,2010.
- [3] 陈东. 嵌入式运动控制器高级语言设计与实现[D]. 济

南:山东大学机械工程学院,2005.

- [4] 董红丹. 单片机数控系统软件及运动控制语言的设计与研究[D]. 西安:西安理工大学机电工程系,2007.
- [5] 夏露露. 基于运动控制器的运动控制语言的设计[D]. 济南:山东大学机械工程学院,2007.
- [6] 王明武,张士勇,陈应舒. IEC61131-3编程语言的现状与发展[J]. 陕西:陕西理工学院机械工程学院,2010.
- [7] SUNDER C, ZOITL A, MEHOFER F, et al. Advance use of PLCopen motion control library for autonomous servo drives in IEC61499 based automation and control systems [J]. *Elektrotechnik und Informationstechnik*, 2006, 123(5):191-196.
- [8] MASSEY N, ENGINEER S. PLCopen motion control: Reducing development time and cost with standardized motion programming [EB/OL]. [2008-05-19]. <http://www.e-driveonline.com/whitepapers/B&R2.pdf>.
- [9] 严义,朱旭燕. 基于DSP的PLC运动功能的研究[J]. 机电工程,2011,28(7):818-822.
- [10] 朱旭燕. 高性能DSP可编程逻辑控制平台的研究[D]. 杭州:杭州电子科技大学计算机学院,2010.
- [11] 郭书杰. 软件PLC梯形图编程系统的研究与实现[D]. 北京:中国科学院研究生院,2010.
- [12] 林惠标,焦志刚. 基于带抑止弧Petri网的PLC梯形图设计方法[J]. 机电工程技术,2011,40(4):73-75.
- [13] 卢冠军. 一种多核嵌入式PLC运动控制系统的研究[D]. 杭州:杭州电子科技大学计算机学院,2011.
- [14] 闫莎莎. 基于DSP+CPLD的多轴运动控制器平台设计及单轴伺服运动控制算法研究[D]. 杭州:浙江大学环境工程学院,2010.

[编辑:李辉]

(上接第758页)

参考文献(References):

- [1] 李黎,付宇卓,汪宁. SoC芯片中关键路径的优化方法研究[J]. 微电子学与计算机,2006,23(4):141-145.
- [2] 王琪. 半定制集成电路的设计[J]. 信息技术与标准化,2006(5):31-35.
- [3] BAI Fu-jun, GAO Zhi-qiang, XU Yi, et al. A Watermarking Technique for Hard IP Protection in Full-custom IC Design [C]//IEEE International Conference on Communications, Circuits and Systems Proceedings. kokura: [s.n.], 2007:1177-1180.
- [4] 刘志哲. 一种使用Advance MS的全定制加法器加速设计[J]. 微电子学,2010,40(4):566-569.
- [5] PILATO C, FERRANDI F. A Fast Heuristic for Extending Standard Cell Libraries with Regular Macro Cells [C]//IEEE Annual Symposium on VLSI. kefalonia: [s.n.], 2010:

23-28.

- [6] 卢俊. 高性能的标准单元库设计[J]. 航空计算技术,2007,37(3):85-93.
- [7] 曾宪恺,郑丹丹. 基于标准单元库扩展的快速乘法器设计[J]. 计算机应用研究,2012,29(5):1778-1781.
- [8] RABAEY J M, CHANDRAKASAN A. Digital Intergrated Circuits-A Design Perspective [M]. 2nd ed. Beijing. Press of Tsinghua University,2004.
- [9] 杨东. 基于逻辑功效模型的数字电路延迟估算与优化[J]. 微处理器,2010(5):21-27.
- [10] LEE C Y. An algorithm for path connections and its application [J]. *IEEE Transactions on Electronic Computer*, 1961, EC-10(3):346-365.
- [11] 李训根,罗小华. 标准单元工艺参数提取工具的设计及实现[J]. 微电子学与计算机,2007,24(6):12-15.

[编辑:李辉]