

多标准宏块预测与边界强度计算的 VLSI 架构及应用

陶昱良,何卫锋,王 琴,毛志刚,李涌伟,郑吉君
(上海交通大学 微电子学院,上海 200240)

摘 要: 提出一种支持 H. 264 High Profile 4. 1 和 AVS JiZhun Profile 6. 0 的多标准宏块预测与边界滤波强度计算的 VLSI 架构,该架构根据解码器的算法特点,实现了 H. 264 和 AVS 标准中控制占优的帧内模式预测、帧间运动矢量预测以及边界滤波强度计算算法,能应用于当前的可重构多媒体系统. 对该架构进行实现后,采用 TSMC 65 nm 工艺综合,工作频率可达到 312 MHz,解码一个 H. 264 和 AVS 宏块最大分别消耗 351 和 189 个时钟周期,能够满足 H. 264 和 AVS 高清(1080p)实时处理的需求.

关键词: 多标准;帧内模式预测;运动矢量预测;边界滤波强度计算;高清实时解码

中图分类号: TN47

文献标识码: A

文章编号: 1000-7180(2011)06-0026-05

A VLSI Architecture and Application for Multi-Standard Macro-block Prediction and Boundary Strength Calculation

TAO Yu-liang, HE Wei-feng, WANG Qin, MAO Zhi-gang, LI Yong-wei, ZHENG Ji-jun

(School of Microelectronics, Shanghai Jiaotong University, Shanghai 200240, China)

Abstract: In this paper, a VLSI architecture is proposed for macro-block prediction and boundary strength calculation which supports H. 264 (High Profile 4. 1) and AVS (JiZhun Profile 6. 0) video decoder. The control-dominant algorithms, intra mode prediction, motion vector prediction and boundary strength calculation in H. 264 and AVS are implemented by this architecture that can be applied to the current reconfigurable multi-media systems. After design implementation, the work frequency can be up to 312MHz at TSMC 65nm technology library. The number of maximum cycles to decode one H. 264 and AVS macro-block is 351 and 189 respectively, which meets the requirement of real-time processing for H. 264 and AVS High Definition (1080p).

Key words: multi-standard; intra prediction mode; motion vector prediction; boundary strength calculation; real-time HD decoding

1 引言

H. 264 和 AVS 利用视频图像在时域和空域上的相关性,使得压缩率远远大于其他视频标准,因此也带来了更为复杂的解码过程. H. 264 和 AVS 的解码算法可以分为运算占优和控制占优两类. 运算占优的算法占到整个解码器比重的 70% 以上^[1],具有运算量大、规整且易于并行实现的特点. 考虑到为了适应不同的视频标准,这类算法可以在可重构运算阵列上实现,以达到很高的加速比^[1]. 而可重构计

算阵列仅适合运算量大、算法规整的加速, H. 264 和 AVS 中控制占优的算法不适合映射到阵列上实现^[2],因此只能通过 ASIC 实现以满足高清视频实时解码的需求.

在 H. 264 和 AVS 中控制占优的算法中,熵解码算法较为特殊,一般采用单独的 ASIC 进行实现^[2-3]. 宏块预测中的帧内模式预测和帧间运动矢量预测,以及边界滤波计算都具有利用宏块空间相关性进行预测的特点,并且边界滤波强度的计算依赖于运动矢量预测的结果,因此,将这三者结合设计可

收稿日期: 2010-09-08; 修回日期: 2010-11-02

基金项目: 国家“八六三”计划资助项目(2009AA011705)

以有效的复用硬件,提高性能并减少存储开销.当前对 H. 264 和 AVS 控制占优算法进行 ASIC 实现有较多实例,但是大多数无法同时兼容两种标准.即使对多种标准进行了复用实现,也没有考虑将具有共性和依赖关系的算法结合设计,以达到最大程度复用硬件和提高效率的目的.文献[4-5]分别对 H. 264 和 AVS 的帧间运动矢量预测进行了实现,但不能同时兼容 H. 264 和 AVS 两种标准.虽然文献[6]中实现了 H. 264 和 AVS 两种标准的运动矢量预测,但是当前仍没有一个有效的 VLSI 结构将两种标准的三种算法结合设计.

本文将帧内模式预测、帧间运动矢量预测和边界滤波强度计算三种具有共性和依赖关系的算法结合实现,能兼容 H. 264 和 AVS 两种解码标准,并且该 VLSI 结构特别适合当前的多媒体可重构 SoC 系统^[2],能有效复用硬件设计,提高解码效率,满足高清视频实时处理的需求.

2 算法分析

宏块预测为图像样点重建提供必要的的数据,包括帧内模式预测和帧间运动矢量预测.边界滤波强度计算目的是为去块滤波提供滤波强度值. H. 264 和 AVS 解码器中这三部分算法不仅在流程上十分相似,而且都具有基于宏块划分和利用相邻块在空间上的相关性进行预测的特点,因此可以设计一套复用的硬件结构.

2.1 宏块划分方式

H. 264 和 AVS 的宏块预测与边界滤波强度计算均采用基于 16×16 宏块划分的方法.在宏块预测中,对于帧间图像宏块, 16×16 、 16×8 、 8×16 和 8×8 是 H. 264 和 AVS 公有的划分方法. H. 264 在宏块进行 8×8 的方式划分时,还可以对 8×8 子宏块进行 8×4 、 4×8 和 4×4 的进一步划分.而对于帧内图像宏块, H. 264 存在 16×16 、 8×8 和 4×4 的划分方式, AVS 只存在 8×8 的划分方式. H. 264 对边界强度计算的处理是基于 4×4 块的划分方式,分别计算每个 4×4 子块垂直和水平边界的滤波强度,而 AVS 则是按照 8×8 块的划分方式,计算四个 8×8 块的垂直与水平边界的滤波强度值.

因此,不管是宏块预测还是边界滤波强度计算, AVS 宏块划分方式都是 H. 264 宏块划分方式的子集,如果统一对宏块采用 4×4 块的划分方式,这样可以适应两种标准的帧内模式预测、运动矢量预测以及边界滤波强度计算的处理需求.

2.2 空间相关性的预测

H. 264 和 AVS 宏块预测与边界滤波强度计算的一个共同点就是都利用了相邻块在空间上的相关性预测当前块的信息.利用宏块的相关性进行预测,需要保存当前宏块左边和上边宏块的信息.当图像按照光栅扫描的顺序逐行进行宏块预测时,对当前行宏块的预测需要使用上一行宏块已经预测得到的信息,因此,需要使用一个 Buffer 缓存上一行宏块的信息,我们称之为 Line Buffer. Line Buffer 中相邻块信息的存储与更新示意图如图 1 所示.

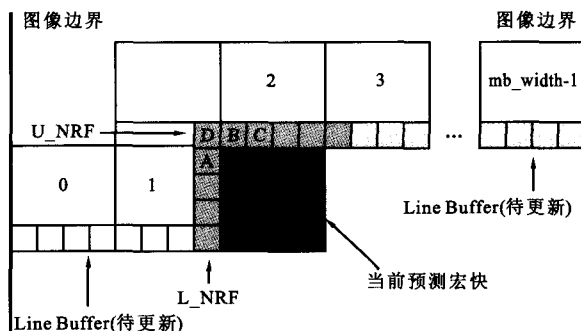


图1 Line Buffer 存储与更新示意图

在图 1 中,对于当前被预测的子块 E,帧内模式预测利用左块 A 和上块 B 预测得到 E 块的预测模式.边界滤波强度计算利用左块 A 和上块 B 的运动矢量、参考索引等信息分别计算 E 块左垂直边和上水平边的滤波强度值.而运动矢量预测的空域模式除了需要左块 A 和上块 B 的信息外,还需要右上块 C 和左上块 D 来预测当前块 E 的运动矢量以及参考索引信息.我们对 H. 264 和 AVS 的宏块采用 4×4 的划分方式,因此 Line Buffer 只需要保存上一行宏块中最下面一行 4×4 块的信息即可.在预测过程中,除了需要 Line Buffer 保存上一行的 4×4 块信息,还需要保存左边宏块的最右边四个 4×4 块信息,图 1 中 L_NRF 即为该组相邻块寄存器.为了支持 1920×1080 高清分辨率,Line Buffer 需要 $1920/4=480$ 个存储单元,每个单元存储帧内预测模式、运动矢量、参考索引以及用于 H. 264 边界强度计算的非零系数标志(block_non_zero_flag)等信息.

2.3 运动矢量预测中运算的复用

H. 264 和 AVS 在运动矢量预测上存在一些细节上的差异^[6],但在总体流程和处理方式上十分相似.根据处理方式的不同,运动矢量预测可以空域模式和时域模式两大类,其中空域模式包括 P_Skip 模式、一般空域模式以及 AVS 特有对称模式,而时域模式利用时域上的相关性进行预测,即直接时域模式.运动矢量预测除了控制十分复杂外,还存在大量

复杂的计算,如式(1)~(5)所示.

$$MVX_y = \text{Sign}(mvX_y) \times ((\text{Abs}(mvX_y) \times \text{BlockDistE} \times (512 / \text{BlockDistX}) + 256) \gg 9) \quad (1)$$

$$mvBw_y = -((mvFw_y \times \text{BlockDistBw} \times (512 / \text{BlockDistFw}) + 256) \gg 9) \quad (2)$$

$$mvFw_y = \pm(((16384 / \text{BlockDistRef}) \times (1 \pm mvRef_y \times \text{BlockDistFw}) - 1) \gg 14) \quad (3)$$

$$\text{DistScaleFactor} = \text{Clip3}(-1024, 1023, (tb * tx + 32) \gg 6) \quad (4)$$

$$mvL0 = (\text{DistScaleFactor} \times mvCol + 128) \gg 8 \quad (5)$$

上述公式中,其中式(1)~(3)分别为 AVS 空域模式、对称模式和时域模式的计算表达式,而式(4)、(5)是 H. 264 时域中计算运动矢量的表达式.很容易看出这些公式中存在很多相同的运算,如除法和乘加移位运算,因此针对上述公式可以设计出复用的运算单元.

3 VLSI 架构

通过上述的算法分析,我们提出了如图 2 所示的 VLSI 架构.该架构不仅能兼容 H. 264 和 AVS 两种解码标准的宏块预测和边界滤波强度计算算法,还能适用于当前的可重构多媒体系统^[2].

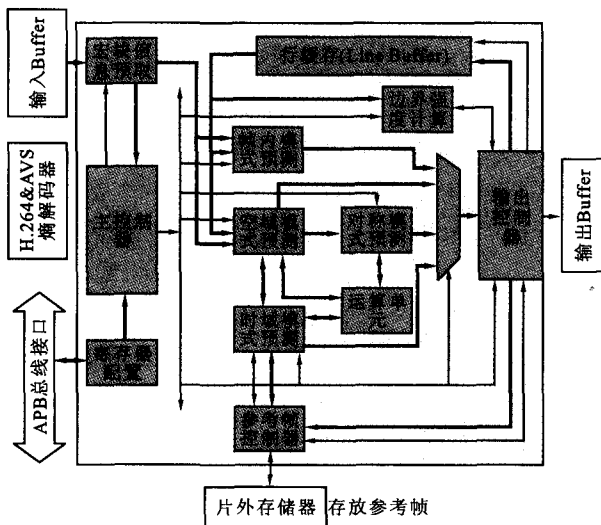


图 2 宏块预测与边界滤波强度计算 VLSI 架构

3.1 系统结构说明

该系统结构主要分为四个部分,包括输入输出控制,寄存器配置、主控制器和预测实现.

输入输出控制包括宏块信息预取模块和输出控制模块.预取模块从输入 Buffer 中读取熵解码得到的宏块信息,输出控制模块完成两部分任务:一是在宏块预测过程中更新相邻块信息;二是在宏块预测完成后更新 Line Buffer 信息以及将结果输出到输

出 Buffer 中供后续解码单元处理.

寄存器配置模块提供 APB 总线接口对系统进行寄存器读写,配置解码关键参数,如状态机使能、码流类型(H. 264 或 AVS)、图像高度、图像宽度、图像编码类型以及 MBAFF 等信息.寄存器配置使得硬件设计更为灵活,并且能够简化硬件设计.

主控制器通过系统有限状态机控制整个系统的工作过程,包括控制输入输出、宏块参数解析以及控制子模块状态机等.

预测实现完成宏块预测与边界滤波强度计算,包括帧内模式预测模块、空域模式预测模块、对称模式预测模块、时域模式预测模块、运算单元、BS 计算模块以及 Line Buffer 模块和参考帧控制器模块.帧内模式预测模块完成帧内模式预测,包括亮度和色度的帧内模式预测.空域模式预测模块完成 P_Skip 模式、一般空域模式以及 AVS 对称模式前运动矢量的预测.时域模式预测模块处理直接时域模式下的运动矢量预测.所有模式中需要复杂运算的操作都在运算单元模块中完成,该模块的详细设计将在 3.2 节描述.BS 计算模块在帧内模式预测或者运动矢量预测完成后计算宏块的边界滤波强度.参考帧控制器模块主要用于控制参考帧图像的存取,为了有效的存取参考帧,该模块包括了一个 64×64 比特的 FIFO 和 64×32 比特的 Cache,以支持对外部 SRAM 的猝发写入和读取.猝发写入和读取的个数可以由寄存器配置成 8、16 和 32 以满足不同系统的需求.

3.2 运算单元的复用与流水线设计

2.3 节中分析了 H. 264 和 AVS 的运动矢量预测中存在的大量运算.如 AVS 运动矢量预测的空域模式需要利用式(1)对 12 个 MV 分量进行缩放运算,因此有必要设计一个专门的运算单元进行流水线操作,以提高预测的性能.

我们将 2.3 节中的式(1)~(5)拆分,设计成如图 3 所示的流水线结构.由于 H. 264 的时域模式 MV 的计算表达式较为复杂,DistScaleFactor 采用寄存器配置的方式获得,这样可以大大简化设计和最大程度的复用硬件.可以看到 AVS 时域模式的硬件结构完全可以兼容其他模式,因此将该结构作为复用的硬件结构.如图 3 所示,乘法器和除法器均使用 DesignWare 库的标准单元,其中 15 bit/9 bit 除法器采用两级流水线结构,两个 $15 \text{ bit} \times 10 \text{ bit}$ 和 $15 \text{ bit} \times 26 \text{ bit}$ 的乘法器均采用一级流水线结构.整个计算单元为 6 级流水线,在最多计算 12 个数据时需要消耗 18 个时钟周期.

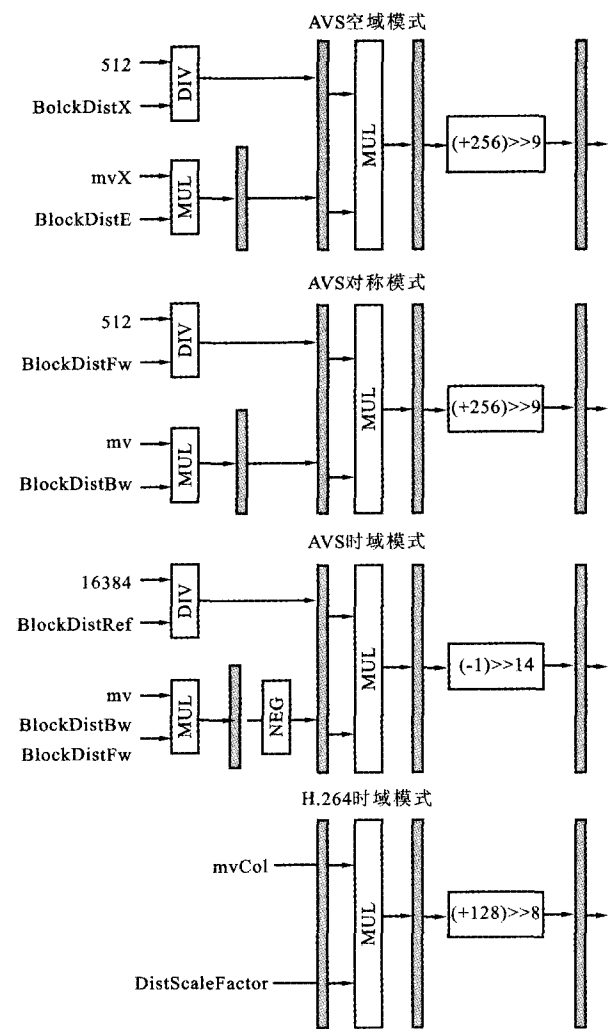


图 3 运算单元流水线设计

3.3 边界滤波强度计算单元

H. 264 和 AVS 的边界滤波强度计算在处理流程上几乎完全相同,不同点在于 H. 264 的滤波强度值范围可以从 0~4,而 AVS 的滤波强度值最大为 2. 边界滤波强度的计算过程存在大量分支判断,为了提高效率,我们采用文献[7]中的技术,并行地计算不同的分支路径,最后选择正确的输出结果. 对文献[7]中的结构进行改进,增加 is_h264 信号以得到 AVS 正确的边界滤波强度值,并且在图像边界(pic_bound 为 1)时将滤波强度值设为 0. 对于相邻块 P 和 Q 的 MV,先计算其差值的绝对值,然后将结果从第 3 bit 开始的所有高位做“按位与”操作来判断的 MV 的差值是否大于等于 4. 图 4 给出了边界滤波器强度计算单元的基本结构,该结构需要 3 个时钟周期的延时.

4 实现与结果比较

对上述 VLSI 结构进行实现后,利用 H. 264 和

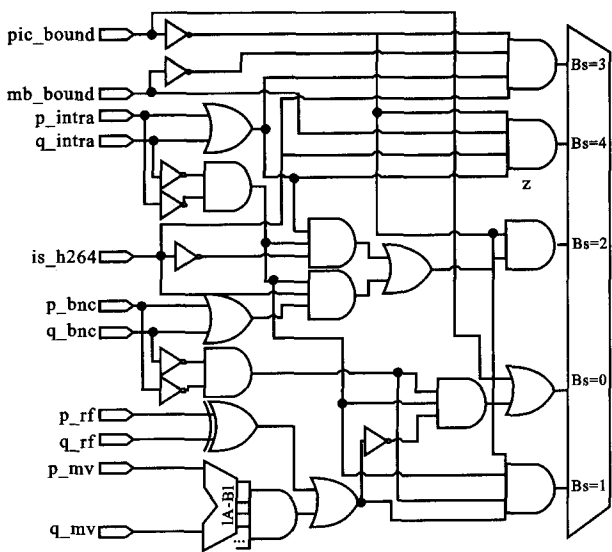


图 4 边界滤波强度计算单元

AVS 标准 C 模型对输出结果进行验证. 利用 Synopsys Design Compiler 进行逻辑综合,在 TSMC 65 nm 工艺下该结构工作频率可达到 312 MHz. 对多个标准码流进行测试处理,解码 H. 264 和 AVS 一个宏块分别最大消耗 351 和 189 个时钟周期. 如果考虑 30f/s@1080p 的处理,只需要工作在 85. 3 MHz 即可实现对 1080p 的实时解码.

与文献[4-5]的结果对比(表 1). 文献[4]虽然在各方面性能较好,但只能支持 H. 264 Baseline 的运动矢量预测,而 H. 264 运动矢量预测所需最大周期数正好出现在 Baseline 不具有的 B 帧预测中. 本文在实现兼容 H. 264 High Profile 4. 1 和 AVS Jizhun Profile 6. 0 的帧内模式预测、运动矢量预测以及边界滤波强度计算后,在面积开销增加不大的情况下明显能达到比文献[4-5]更好的解码性能.

表 1 结果比较

比较项	文献[4]	文献[5]	本文
支持特性	H. 264 运动 矢量预测	AVS 和 MPEG-2 运动 矢量预测	H. 264 和 AVS 帧内模式预测、运 动矢量预测、边界滤波强度计算
工艺	TSMC 180 nm	TSMC 180 nm	TSMC 65 nm
最大频率	108 MHz	148 MHz	312 MHz
门数	52k	62k	66k
SRAM	23k	34k	38k
最大周期 数/MB	H. 264 : 260	AVS: 310	H. 264 : 351
平均周期 数/MB	H. 264 : 160	—	AVS: 189(包括 Bs 计算) H. 264 : 165 AVS: 76(包括 Bs 计算)
处理能力	1 920×1 080 H. 264 Baseline	1 920×1 080 MPEG2, AVS Jizhun Profile 6. 2	1 920×1 080 H. 264 High Pro- file 4. 1, AVS Jizhun Profile 6. 0

5 结束语

本文根据 H. 264 和 AVS 算法的特点,提出了一种有效实现 H. 264 和 AVS 解码器中控制占优的

宏块预测与边界滤波强度计算算法的 VLSI 架构。通过利用算法之间的共性与依赖关系,该架构最大程度的复用硬件设计和减少存储空间,并达到高效的解码性能。该结构不仅能够支持 H. 264 1080p@HiP 和 AVS 1080p@JiZhun Profile 实时处理的需求,还能应用于当前的可重构多媒体系统,提高系统的解码性能。综合和仿真后的结果表明,该结构在面积和性能上均优于当前类似文献的结果,远远满足 H. 264 和 AVS 高清实时处理的需求。

参考文献:

- [1] Geng T S, LIU Leibo, Yin Shouyi, et al. Parallel implementation of computing intensive decoding algorithms of H. 264 on reconfigurable SoC[C]// Proc of 2010 IEEE International Symposium Circuits and Systems (ISCAS). Paris: IEEE, 2010:1153-1156.
- [2] Zhu M, Liu Leibo, Yin Shouyi, et al. A reconfigurable multi-processor SoC for media applications[C]// Proc of 2010 IEEE International Symposium, Circuits and Systems (ISCAS). Paris: IEEE, 2010:2011-2014.
- [3] 黄玄, 陈杰, 李霞, 等. AVS 高清视频变长码解码器算法与电路实现[J]. 微电子学与计算机, 2009, 26(1):76-80.
- [4] Yin H B, Zhang D P, Wang Xiumin, et al. An efficient MV prediction VLSI architecture for H. 264 video decoder[C]// 2008 International Conference on Audio, Language and Image Processing, Shanghai: ICALIP, 2008: 423-428.
- [5] Zheng J H, Wu D, Deng L, et al. A motion vector predictor architecture for AVS and MPEG-2 HDTV decoder[J]. Journal of Computer Science and technology, 2006, 21(3): 424-431.
- [6] 谢朝辉, 冯燕, 李谦, 等. H. 264 和 AVS 多模视频解码器中运动矢量预测的硬件[J]. 微电子与计算机, 2006, 23(11):54-57.
- [7] Dang P. High performance architecture of an application specific processor for the H. 264 deblocking filter[J]. IEEE Trans on VLSI Systems, 2008, 16 (10): 1321-1334.

作者简介:

陶昱良 男, (1985-), 硕士研究生. 研究方向为视频编解码处理的 ASIC 和 SoC 设计, 高性能可重构运算阵列架构。

何卫锋 男, (1978-), 博士, 副教授. 研究方向为视频编解码系统的 VLSI 设计, SoC 设计方法学与系统集成。

王 琴 女, (1975-), 博士, 副教授, 研究方向为音视频处理应用开发 SoC, 高性能低功耗体系架构。

(上接第 25 页)

该自演化系统模型是在较高的抽象水平建立的, 可用以加速验证、性能分析和系统结构的探索。目前, 整个系统在行为级已经仿真实现, 后期的工作可以综合细化到 RTL, 使整个系统能在现场可编程门阵列(Field Programmable Gate Array, FPGA)上实现。此系统还可以增加其他的模块, 比如: 通用串口, 通用 I/O, 总线等模块以完成更复杂的功能。

另外, 由于目前自演化系统中演化控制采用的是简单的标准遗传算法, 滤波效果不是很好。后期可进一步调整遗传算法控制参数, 并改进遗传算法, 以提高算法效率和搜索结果。

参考文献:

- [1] 姚睿, 王友仁, 于盛林, 等. 具有在线修复能力的强容错三模冗余系统设计及实验研究[J]. 电子学报, 2010, 38(1):177-183.
- [2] Yao Rui, Yu Shenglin, Wang Youren, et al. Online evolution of the algorithm-level digital hardware system using evolvable core[J]. Journal of Chinese Computer Systems, 2009, 30(1):188-192.
- [3] 陈绍贺, 赵明, 王京. 基于 SystemC 的片上系统设计[J]. 微电子学与计算机, 2005, 22(4):51-52.
- [4] 李烨挺, 梁利平. 一种基于 SystemC 的片上网络建模与仿真方法[J]. 微电子学与计算机, 2010, 27(3):78-82.
- [5] Sekanina L. Virtual reconfigurable circuits for real-world applications of evolvable hardware[C]// Proc of the 5th Int Conf on Evolvable Systems: From Biology to Hardware (ICES'03), Trondheim, Norway: Springer, 2003:186-197.
- [6] Chen Xi, Xu Ningyi. SystemC SoC design[M]. Beijing: Science Press, 2004.
- [7] SystemC. [EB/OL]. [2010-09-13]. <http://www.systemc.org>.
- [8] SystemC version2.0 User Guide. Synopsys, Inc. [EB/OL]. (2002)[2010-09-13]. <http://www.systemc.org>.

作者简介:

王晓艳 女, (1983-), 硕士研究生. 研究方向为计算机测控、嵌入式系统设计及应用。

姚 睿 女, (1974-), 博士, 副教授. 研究方向为进化硬件理论与技术、嵌入式测控系统自修复。