

一种新型的定点到浮点转换的全流水线 FPGA 实现

张 益, 袁 著, 罗金选

(电子科技大学 电子科学技术研究院, 四川 成都 611731)

摘 要: 利用 FPGA 器件完全采用底层自主设计, 实现定点数到浮点数的转换. 提出了一种全新的实现方法, 变对数为减法, 通过占用 1% 的逻辑资源, 实现 3 个时钟周期输出数据. 避免了局限于使用 IP core 的束缚, 为后续 ASIC 设计打下了基础.

关键词: 定点数; 浮点数; FPGA; 流水线

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2012)08-0179-06

A New FPGA Realization of the Full-Pipeline of Conversion from Fixed-point to Floating Point

ZHANG Yi, YUAN Zhu, LUO Jin-xuan

(Research Institute of Electronic Science and Technology, University of Electronic Science and Technology of China, Chengdu 611731, China)

Abstract: In this paper, completely independent and bottom-level design is conducted by utilizing FPGA device, realizing conversion from fixed-point to floating point. A completely new method is proposed for such realization to output data by taking 3 clock cycles. In this method, subtraction calculation instead of logarithm calculation is adopted, occupying 1% logic resources and avoiding the restrictions of using IP core and laying a foundation for follow-up ASIC design.

Key words: fixed-point; floating point; FPGA; pipeline

1 引言

众所周知, 数字系统普遍采用定点和浮点两种形式来表示数据. 由于定点数占用资源少, 处理速度快, 易于传输和存储, 因而在硬件中多采用此种方式. 但在某些实时处理系统中, 比如图像压缩, 对数据的精度及动态范围都要求比较高, 这时需要采用浮点数才能满足要求. 但是浮点数相比定点数一是占用的系统资源较多, 二是处理速度较慢, 且在传输上也不方便, 所以在很多处理过程需要进行定点-浮点的相互转换以达到精度与实时性的平衡^[1-2].

图 1 是在某无人机载高清图像压缩系统部分框图. 如图 1 所示, 虚线框内部的 DCT 和量化表除法为数据处理的核心部分, 框内均为浮点数运算, 因此, 虚线框左侧与右侧需分别做定点-浮点以及浮

点-定点的转换.

本文正是针对此类应用提出了一种定点到浮点转换的全新的实现方法, 且在该平台上完成验证和实现, 在保证精度的同时提高了转换效率.

2 常见定点浮点处理方法

对于精度要求不是特别高的场合, 可以不直接转换. 先对浮点算法进行仔细的分析, 结合工程的实际要求, 综合考虑诸多因素, 然后再将其转化为定点算法并通过硬件来实现, 例如截位法, 即用定点来代替浮点运算. 这种方案节省资源, 但误差相对较大.

对于系统中部分模块精度要求很高的情况, 可以对数据精度处理, 核心思想是对数据分段处理, 即对精度要求比较高的地方采用浮点数运算, 对精度要求不高的地方, 或者传输部分, 采用定点数形式.

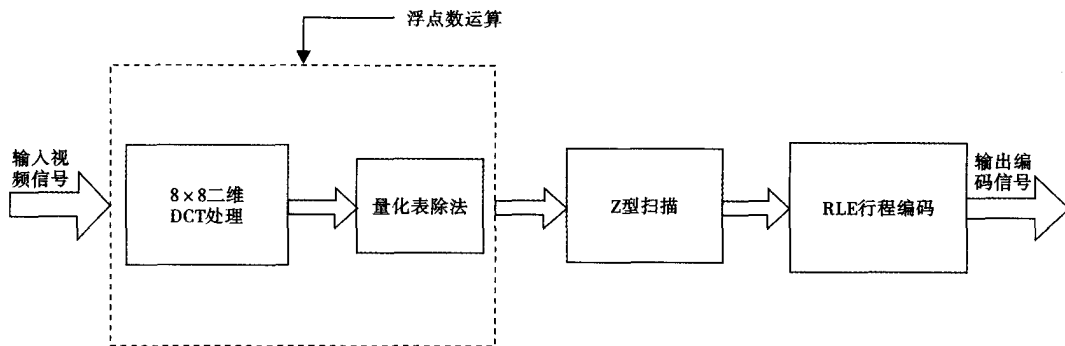


图1 某图像压缩系统框图

这种方法就涉及到定点数到浮点数的转换,资源耗费情况较上一种方案稍多,但精度得到了大大的提高。由于FPGA硬件资源的利用具有可编程性,因此这种方法的实现是完全有可能的^[2]。

纵观业界对于定点到浮点转换的方法,除了上述的使用FPGA来实现之外,还可以在定点DSP上实现浮点运算。定点DSP能从支持浮点运算中获得益处。选择符合DSP运算能力和本地数据宽度的FLP数据格式,增加FLP硬件或软件扩展并提供合适的研发工具^[3]。

还有一种方法可将浮点算法通过定点算法来实现,即在SoC芯片中基于统计分析的转换方法。在进行定点SoC芯片设计时,为了降低成本、功耗,提高精度,提高CPU、总线和内存的使用效率以及实时性要求,一般采用低主频定点CPU核外加芯片内嵌硬件加速模块的模式。将浮点算法到近似定点数算法的转换一般通过使用有限精度的定点近似数来表示无限精度的浮点数。对此策略可粗略地分为两个方向:一种是有限字长统计分析方法,该方法始于对量化错误的统计建模,适合于数字滤波器、FFT等;另一种是数据仿真方法,该方法是最常用的信噪比计算方法,其最大好处就是无差别的处理非线性系统和线性系统^[4]。

对于位宽要求不高的场合,可以使用查表法。这种方法在小位宽时尚可,一旦位宽增加,例如32 bit,将带来的是庞大计算量和资源耗费的增加^[5]。

此外,还可用GAL器件或者纯硬件电路来搭建等等,此处不再赘述。本文是基于上述背景,提出一种全新的定点到浮点转换的FPGA实现方法,减少了CPU的开销。与全球最大的几个FPGA开发商的IP core相比,比开发商一的定点到浮点转换的IP core输出数据快1时钟周期(其IP core4个时钟周期之后输出数据),而开发商二未见相关IP core。

3 浮点数运算原理

在开发设计硬件结构前,需根据应用的数据精度及动态范围确定需要进行数据转换的模块和位宽,按照工程设计的实际需求在数学运算上对算法进行处理并计算验证,从而保证硬件体系结构实现的结果符合设计需求,并且高效地运行^[2]。

为了更好地说明如何完成浮点算法向定点算法的转化,首先对浮点数的概念进行简要回顾。

小数点的位置可按需浮动,这就是浮点数。例如:

$$N = r^E \cdot M$$

式中, r 为浮点数阶码的底,与尾数的基数相同,通常 $r=2$ 。 E 和 M 都是带符号的定点数。

浮点数由三部分组成:指数部分 E (又称为阶码Exponent)、尾数部分 M (Mantissa)和符号位。在大多数计算机中,尾数为纯小数,常用原码或补码表示;阶码为纯整数,常用移码或补码表示。计算机中,通常用约定的4部分来表示一个浮点数:其中, E_f 、 S 分别称为阶码 E 和尾数 M 的符号位^[2]。

E_f	E	S	M
-------	-----	-----	-----

按照IEEE754标准,常用的浮点格式如图2所示。

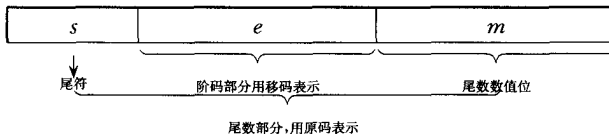


图2 IEEE 754标准的浮点格式

IEEE 754标准中有三种形式的浮点数,格式见表1。

短浮点数即单精度浮点数,长浮点数即双精度浮点数,都采用隐含尾数最高数位的方法,所以增加了一位尾数。临时浮点数又被称为扩展精度浮点数,没有隐含位。

表1 IEEE 754 标准中的三种浮点数

类型	数符	阶码	尾数位	总位数	偏置值
短浮点数	1	8	23	32	7FH(127)
长浮点数	1	11	52	64	3FFH
临时浮点数	1	15	64	80	3FFFH

在短浮点数中,最高位为数符位,其后8位为阶码,用2为底的移码表示,偏移值为127(叫移127码).其余23位是尾数.对规格化的二进制浮点数,约定最高位总是“1”,一般将这个“1”隐含,故尾数数值实际上是1位隐含位加23位小数位,即24位.

短浮点数的移码的偏置值是127(3FH);长浮点数的偏置值是1023(3FFH).存储浮点数阶码部分之前,偏置值要先加到阶码真值上.

以短浮点数为例,其值可以表示为

$$X = (-1)^s \cdot 2^{(e-127)} \cdot (1.m)$$

因此,其转换步骤及方法可以归纳如下:

- ① 将十进制数转换成二进制数(如果本身是二进制数,则此步骤省略);
- ② 根据正负,确定符号位,正数 $s=0$,负数 $s=1$;
- ③ 规格化二进制数,即 $(1.m) \cdot 2^r$ 的形式;
- ④ 计算出阶码的移码(偏置值+阶码真值),注意:短浮点数的阶码偏置值是1111111(127), $e=r+(1111111)_2$;
- ⑤ 以短浮点数格式存储该数,即

s	e	m
-----	-----	-----

4 定点数到浮点数转换的 FPGA 实现

4.1 实现思想

为达到硬件开发设计要求,需根据应用的数据精度及动态范围确定需进行数据转换的模块和位宽,按照实际需求在数学运算上对算法进行处理并计算验证^[2].在前面提到的浮点数原理的基础之上,以下将硬件实现步骤进行介绍.整个系统是基于 Altera 公司的 Stratix II EP2S60 FPGA 芯片完成的^[6].

本模块以常见的8位定点数到32位浮点数转换为例.首先判断正负确定 s .对于确定指数 r ,通常是取对数,或者检测该定点数有效数据位的宽度.无论检测宽度还是取对数,都会很耗资源,且对数运算

需多次进行除法,增加数据位宽以及算法复杂度.我们在此提出一种新的思想,变对数为减法,充分利用 FPGA 的并发结构,数据位宽没有增加,且在最短的时间、用最少的资源实现求指数.

下面介绍具体实现方法.将输入数据 i_data 分别与1、2、4、8、16、32、64、128相减,得到8个差值 i_dec0 、 i_dec1 、……、 i_dec7 ,再用7来减其中负数的个数,即可得指数 r .这样取对数就变成了几次加减法,大大提高了效率,节省了资源.与8个数做减法可一步完成,计算负数个数以及跟7相减一步完成,2个时钟周期即可完成该步骤.计算负数的个数此处通过提取符号位 $i_dec0[8]$ 、 $i_dec1[8]$ 、……、 $i_dec7[8]$ 中1的个数来实现.得出 r 之后,再加127即得 e .同时提取出输入数据中的低 r 位,右侧添0补足23位,即得 m ,整个确定 s 、 e 、 m 的过程如图3所示.

图3中 i_dec0 至 i_dec7 分别为 i_data 与1、2、4、8、16、32、64、128的差值, $i_dec0[8]$ 至 $i_dec7[8]$ 为8个差值的符号位.在确定 s 和 m 时,均对输入数据 i_data 有2个时钟周期的延时,是为满足 FPGA 流水线设计的要求.在确定 e 时,共占用2时钟周期,故在此处延迟2个时钟,同时输出 s 、 e 、 m ,以形成全流水线操作.

4.2 FPGA 实现结果

将上述思想转换为代码在 Modelsim 上编译、仿真之后,得到仿真结果的波形如图4所示.

图4中 i_data 为8位定点数据输入,范围0~255, f_data 为32位浮点数据输出.整个系统属于全流水操作,数据在时钟节拍的驱动下源源不断地输入系统.需要注意的是输入使能 en ,只有在 en 高电平的时候输入数据才有效,否则输入数据不会送入到系统中进行转换,即高电平使能. out_en 为输出使能指示,在其为高电平期间,表示此时为有效的数据输出,同样是高电平输出使能.输入数据3个时钟周期之后输出数据. i_data1 、 i_data2 为延时信号.

本模块在 Quartus 中成功完成编译、综合及时序分析.图5是通过 Quartus 综合工具生成的 RTL 图,表2、表3是 FPGA 内部资源耗费与时序约束情况.

图中第一、二列的运算单元实现与输入数据求差,第三列的寄存器提取出差值 i_dec0 到 i_dec7 的符号位,随后是几个加法器实现后续的一系列的加减法运算,然后结果进入到一个寄存器中得到 r ,之

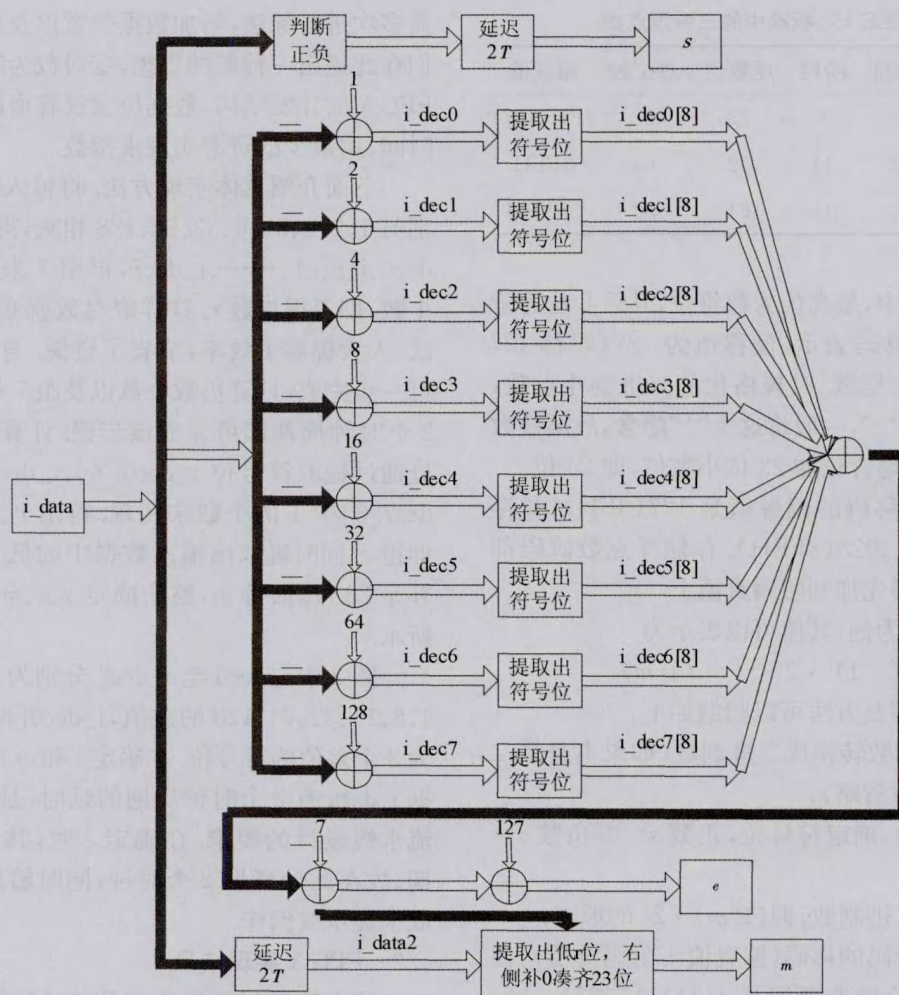
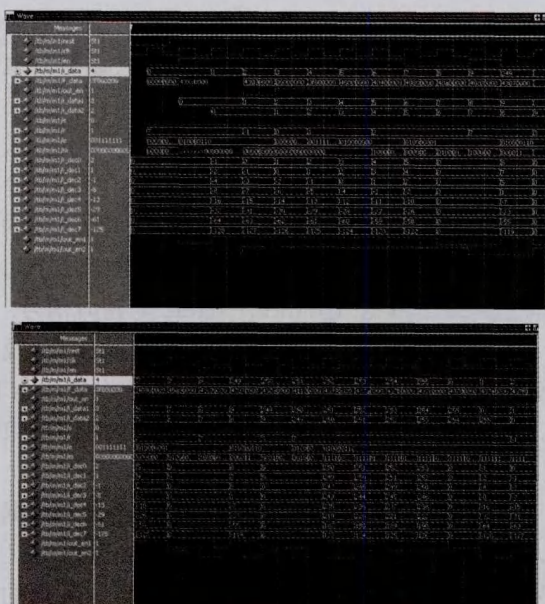
图3 基于FPGA的求解浮点数中 s 、 e 、 m 参数的流程图

图4 基于FPGA的定点到浮点转换仿真结果波形图

表2 定点到浮点数的转换FPGA资源耗费

Quartus II Version	9.1 Build 222 10/21/ 2009 SJ Full Version
Revision Name	fix_to_float
Top-level Entity Name	fix_to_float
Family	Stratix II
Device	EP2S60F1020C4
Timing Models	Final
Met timing requirements	Yes
Logic utilization	<1%
Combinational ALUTs	58/48,352 (<1%)
Dedicated logic registers	44/48,352 (<1%)
Total registers	44
Total pins	44/719 (6%)
Total virtual pins	0
Total block memory bits	0/2,544,192 (0%)
DSP block 9-bit elements	0/288 (0%)
Total PLLs	0/12 (0%)
Total DLLs	0/2 (0%)

后是两列包括选择器、译码器、或门等组合电路,得到的结果包括 m 、 e 、 out_en 等进入到最后一列寄存器中,由此送到输出端。从 RTL 图可以看出本模块

的 FPGA 资源利用情况,图中采用全流水线设计,效率高,逻辑结构清晰,时序稳定,大大避免了竞争与冒险。

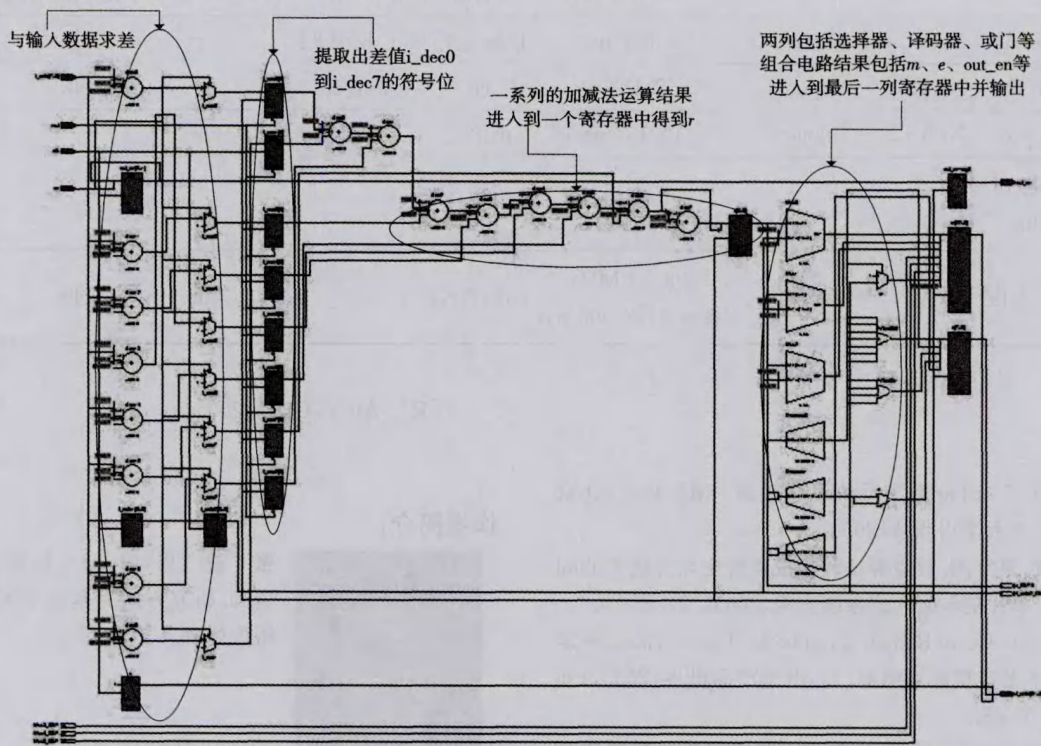


图5 定点到浮点数的转换 FPGA 实现 RTL 图

从表2中可以看出本模块中 FPGA 资源耗费情况,使用逻辑单元其中 Combinational ALUT 58 个, Dedicated logic register 44 个,占整个 FPGA 芯片内部资源的 1% 不到,可以看出,通过较高的资源利用率实现了所要求的功能,大大提高了效率。从表3的 Timing Analyzer 报告中可以看出时钟建立情况,实际的时钟频率可达到 399.04 MHz,即周期 2.506 ns。

综合完成之后,将其下载到 Altera 的 FPGA 基板上进行调试,并成功得到了验证(见图6)。

5 结束语

本文从定点到浮点转换的运算原理入手,与其他几种定点浮点的处理方法相比较,提出了一种全新的基于 FPGA 的定点到浮点的转换方法,较其他方法而言,提高了速度和效率,节省了资源,并且已在基于 EP2S60 的 FPGA 硬件平台上得到了验证。

最后给出了实现框图、仿真结果波形图、综合之后的 RTL 图以及 FPGA 资源耗费情况。该模块已经在某子弹药项目以及某无人机载高清图像压缩系统平台上得到运用,获得了良好的效率和可靠性,为后续数据处理平台提供了准确稳定的保证。

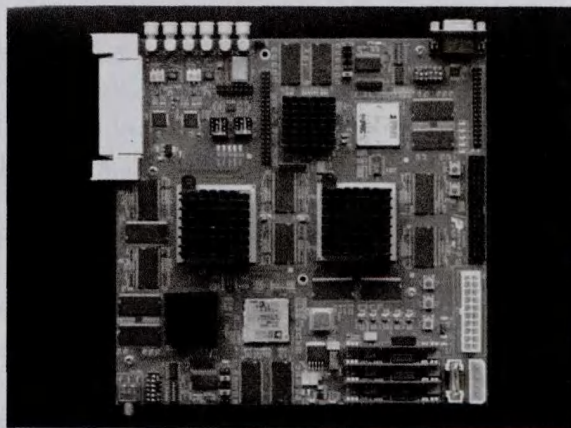


图6 定点到浮点数的转换实现的 FPGA 基板

表 3 定点到浮点数的转换 FPGA 时序约束

Type	Slack	Required Time	Actual Time	From	To	From Clock	To Clock	Failed Paths
Worst-case tsu	N/A	None	4.967 ns	i_data[4]	i_dec1[8]	---	clk	0
Worst-case th	N/A	None	-2.137 ns	en	out_en1	---	clk	0
Worst-case tco	N/A	None	10.137 ns	m[20]	f_data[20]	clk	---	0
Total number of failed paths								0
Clock Setup: 'clk'	N/A	None	399.04 MHz (period=2.506 ns)	i_dec2[8]	r[2]	clk	clk	0

[R]. Altera Corp. 2005.

参考文献:

- [1] 夏宇闻. Verilog 数字系统设计教程 [M]. 北京:北京航空航天大学出版社, 2009.
- [2] 唐清贵, 夏宇闻. 浮点算法转换成硬件定点算法中的问题[J]. 单片机与嵌入式系统应用, 2005(12): 75-76.
- [3] LSI Logic Corp. Ramon Trombetta Tim O Gara. 在定点 DSP 上实现浮点运算[J]. 电子产品世界, 2005, 196(11): 67-68.
- [4] 周凡, 杨军, 尹爱昌. SoC 芯片中基于统计分析的浮点到定点转换方法[J]. 电路与系统学报, 2007, 12(1): 124-129.
- [5] 袁著, 张伟. 离散余弦变换原理以及在 FPGA 上的实现. 综合电子系统技术教育部重点实验室 2009 年度会议暨电子科技大学电子科学技术研究院第五届学术会议论文集[C]//成都: 电子科技大学出版社, 2009.
- [6] Stratix II EP2S60 DSP Development Board Data Sheet

作者简介:



张 益 男, (1982—), 硕士研究生, 讲师. 研究方向为基于 FPGA 的数字图像处理.

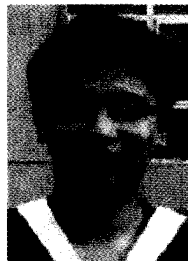
袁 著 男, (1983—), 硕士. 研究方向为雷达数字信号处理.

罗金选 男, (1984—), 硕士研究生. 研究方向为嵌入式系统设计.

(上接第 178 页)

- [9] 王燕妮, 樊养余. 改进 BP 神经网络的自适应预测算法[J]. 计算机工程与应用, 2010, 46(17): 23-26.
- [10] HAGAN Martin T, DEMUTH Howard B, BEALE Mark H. 神经网络设计[M]. 戴葵, 译. 北京: 机械工业出版社, 2002.
- [11] 石伟, 孙永荣. BP 神经网络在车牌识别技术中的应用[J]. 工业控制计算机, 2010, 23(4): 17-19.
- [12] 尹显东, 姚军, 李在铭. 基于 BP 神经网络的图像感兴趣区自动检测技术[J]. 系统工程与电子技术, 2006, 28(2): 192-196.
- [13] 何四华, 杨绍清, 邵晓方, 等. 利用图像区域自然尺度特征的海面舰船目标检测[J]. 红外与激光工程, 2011, 40(9): 1812-1817.

作者简介:



金 凯 男, (1985—), 硕士研究生. 研究方向为信号与信息处理;



何 野 男, (1960—), 教授, 硕士生导师. 研究方向为微电子学、电子设计自动化、计算机辅助设计.