

文章编号: 2097-1974(2024)02-0086-05

DOI: 10.7654/j.issn.2097-1974.20240214

雷达对抗中基于FPGA输入分块重映射的数据处理方法

薛晓琴^{1,2}, 孟刚^{1,2}, 陈阳¹, 李宏博³, 赵健³

(1. 北京航天长征飞行器研究所, 北京, 100076; 2. 试验物理与计算数学国家重点实验室, 北京, 100076;

3. 哈尔滨工业大学, 哈尔滨, 150006)

摘要: 雷达对抗中, 干扰机侦收雷达的信号, 经AD采样后输入FPGA进行处理并产生干扰信号。FPGA平台利用Xilinx等官方提供的FFT IP核对高速ADC输出的数据进行FFT处理时, 需要进行“并行-串行”的数据预处理, 降低了处理速度, 同时也没有对FPGA内部资源进行最大化利用。基于以上问题, 提出一种基于FPGA输入分块重映射的数据处理方法。该方法通过数据输入重映射模块将输入雷达数据优化为并行分块的数据流格式, 再通过FFT蝶形网络输出离散傅里叶变换数据, 最后进行并行结果输出。试验证明, 本方法能有效节省FPGA的运算时间, 提高雷达数据处理速度, 优化利用了FPGA内部资源, 具有实时性好、灵活性高的特点。

关键词: 雷达对抗; FPGA; 分块重映射; 数据处理; 蝶形网络

中图分类号: V44

文献标识码: A

Data Processing Method based on FPGA Input Block Remapping in Radar Countermeasures

XUE Xiaoqin^{1,2}, MENG Gang^{1,2}, CHEN Yang¹, LI Hongbo³, ZHAO Jian³

(1. Beijing Institute of Space Long March Vehicle, Beijing, 100076; 2. National Key Laboratory of Science and Technology on Test Physics and Numerical Mathematics, Beijing, 100076; 3. Harbin Institute of Technology, Harbin, 150006)

Abstract: In radar countermeasures, the jammer detects the radar signal, then the signal is sampled by AD and input to FPGA for processing and generating interference signal. When using the FFT IP core officially provided by the FPGA platform Xilinx to perform FFT processing on the data output by the high-speed ADC, data preprocessing of "parallel-serial" is required, which greatly slows down the FFT processing speed. At the same time, the internal resources of FPGA is not maximized. An data processing method based on FPGA input block remapping is proposed. This method uses the data input remapping module to optimize the input data into a parallel block data stream format. Then the FFT butterfly networks process the input data into blocks to obtain the discrete Fourier transform data. Finally, parallel data are output. Experimental results show that this method can effectively save FPGA computing time, improve radar data processing speed, optimize the utilization of FPGA internal resources, and has the characteristics of good real-time performance and high flexibility.

Keywords: radar countermeasures; FPGA; block remapping; data processing; butterfly networks

0 引言

随着雷达技术的发展, 高瞬时带宽的雷达出现, 尤其是瞬时带宽达到1 GHz以上的X波段的宽带雷达, 它具有超高的目标检测和识别能力、抗有源和无源干扰能力。为了应对高瞬时带宽雷达, 干扰机需要进行大时、频宽信号的数据采集和存储, 这对干扰机的数据处理能力提出了很大的挑战, 因此, 研究如何提升数据处理速度是现有雷达对抗中很关键的问题。

雷达对抗中, 干扰机天线侦收雷达信号, 雷达信号经过正交混频后产生零中频IQ信号, 由AD采样后输入FPGA进行存储, 当信号识别确定为威胁信号时, FPGA产生干扰信号后被放大转发。在利用FPGA处理超高速(Gsps)ADC采样后的数据时, 由于FPGA逻辑的速度一般跟不上超高速ADC的总线速度, 因此大多数FPGA具有串行器/解串行器(SERDES)模块^[1], 用以将转换器端的快速、窄带串行接口转换为FPGA端的慢速、宽带并行接口, 这

样既降低了系统进行FFT运算的速度，也没有最大化利用FPGA的内部资源。

为了解决利用FPGA对高速采样数据进行FFT变换时速度慢、资源利用不合理、灵活性低的问题，国内外学者进行了大量研究，文献[3]使用硬件描述语言Verilog在FPGA上实现了基于时间抽样的基2-FFT；文献[4]提出了一种改进的缩放FFT技术，其将抽取数据分成更小的部分，并对其应用加窗FFT；文献[5]通过对3个简易计数器的设计，解决了基2时间抽取快速傅里叶变换中蝶形计算地址选取困难的问题；文献[6]提供了一种高精度的快速傅里叶变换算法，其结果表明IEEE-754单精度浮点仿真精度明显高于定点仿真；文献[7]提出了多路复用和流水线式蝶形处理单元的方法，有效地提高了FFT的最大吞吐量；文献[8]提出了采用坐标旋转数字乘法器(Coordinate Rotation Digital Computer, CRDIC)和树形加法器(Kogge-Stone Adder, KSA)进行8点基2-FFT的方法，该方法提供了更好的延迟性能；文献[9]中采用素数因子分解与库利-图基分解相结合的混合分解模式，在省去了一步旋转因子乘法运算的同时有效减小了存储空间和运算量，并采用通用蝶形单元模块设计使得算法能够同时适应基2、基3、基4的FFT运算。

本文提出一种基于FPGA的输入分块重映射的数据处理方法，利用数据输入重映射模块将输入数据优化为并行分块的数据流格式，简化了FFT流程，降低了数据处理时延，优化了资源配置，并且进行了仿真试验，验证了本文所提方法的优越性。

1 FPGA片内FFT实现传统流程

设序列 $x(n)$ 的长度为 N ，其中 $N=2^m$ ， m 为正整数，当序列长度不满足这一条件时，可以通过补零的方式使序列长度满足要求。正是因为 $x(n)$ 的长度为2的整数次幂，所以这种FFT算法被称作基2-FFT，FPGA官方提供的FFT IP核也是基于这种方法设计的。

$x(n)$ 的离散傅里叶变换公式与离散傅里叶反变换公式^[10]分别如下：

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk} = \text{DFT}[x(n)], 0 \leq k \leq N-1 \quad (1)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)W_N^{-nk} = \text{IDFT}[X(k)], 0 \leq n \leq N-1 \quad (2)$$

其中 $W_N^{nk} = \exp\left(-j\frac{2\pi nk}{N}\right)$ ，做如下约定：

$$x_1(n) = x(2n), 0 \leq n \leq \frac{N}{2} - 1 \quad (3)$$

$$x_2(n) = x(2n+1), 0 \leq n \leq \frac{N}{2} - 1 \quad (4)$$

$$X_1(k) = \text{DFT}[x_1(n)], 0 \leq k \leq \frac{N}{2} - 1 \quad (5)$$

$$X_2(k) = \text{DFT}[x_2(n)], 0 \leq k \leq \frac{N}{2} - 1 \quad (6)$$

则基2-FFT公式可以改写为以下形式：

$$X(k) = \begin{cases} X_1(k) + W_N^k X_2(k) & 0 \leq k \leq \frac{N}{2} - 1 \\ X_1(k - N/2) - W_N^{(k - N/2)} X_2(k - N/2) & \frac{N}{2} \leq k \leq N - 1 \end{cases} \quad (7)$$

由式(7)可知，基2-FFT的原理其实是将一个长序列拆分为奇偶子序列，由奇偶子序列的DFT值计算长序列的DFT值。通过不断的序列拆分可以得到蝶形计算流程图，以一个8点基2-FFT为例，如图1所示。

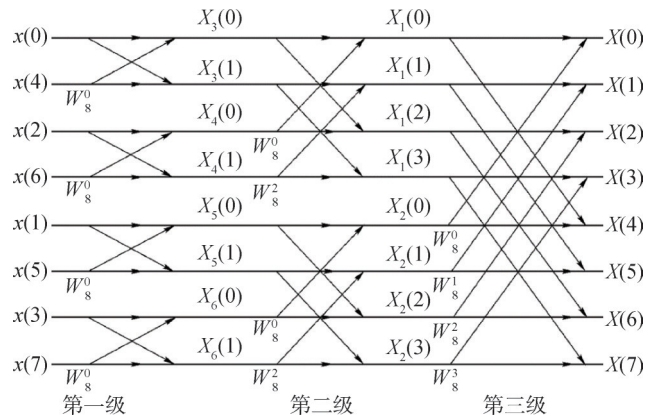


图1 $N=8$ 蝶形计算流程

Fig.1 $N=8$ butterfly computation flow diagram

由式(1)可知，计算每个 $X(k)$ 值需要 N 次复数乘法和 $N-1$ 次复数加法，一共需要计算 N 个值，所以完成一个长度为 N 的序列的DFT，需要 N^2 次复数乘法和 $N(N-1)$ 次复数加法，时间复杂度为 $O(N^2)$ 。而基2-FFT只需要进行 $\frac{N}{2} \log_2 N$ 次基本蝶形计算，每次基本蝶形计算需要1次复数乘法和2次复数加法，一共需要 $\frac{N}{2} \log_2 N$ 次复数乘法和 $M \log_2 N$ 次复数加法，时间复杂度为 $O(N \log_2 N)$ ，相较于DFT计算量大大下降。

FPGA官方FFT IP核计算流程如图2所示。

如果利用官方FFT IP核处理高速ADC采样数据，由于官方IP核只能处理顺序输入的串行数据，所以必须对数据进行“并行-串行”的预处理，以满足IP核处理数据的格式要求，但这样会减慢FFT计算速度，增大时延，因此本文提出一种基于“输入分块重映

射”思想的FFT方法，对其进行改进。

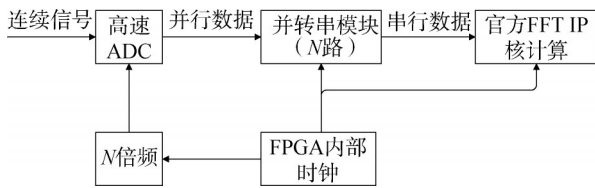


图2 官方FFT IP核计算流程

Fig.2 The official FFT IP kernel computing process

2 输入分块重映射

输入分块重映射的核心思想是直接匹配多通道输入的数据流格式，将并行的ADC采样后无需排列处理，直接将数据输入到IP核中，达到效率最大化，通过设计的输入重映射表将输入数据进行分块映射。数据输入重映射模块是对传统的FFT输入输出倒序关系的进一步改进，将ADC与FPGA接口解串的数字化信号，经过RAM接口使之积累一定的时间点数，输入到IP核的顶层函数，进入蝶形运算模块。通过并行分块输入的方式降低了FPGA所消耗的时钟周期资源，达到匹配并行输出的ADC数据利用率最大化，输入数据重映射关系如图3所示。

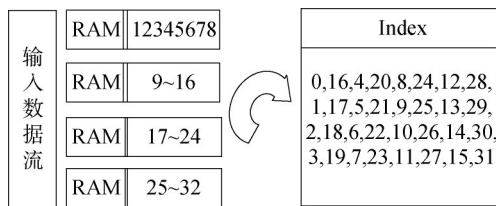


图3 输入数据重映射表的示意

Fig.3 The input data remapping table diagram

3 IP核原理设计

高层次综合 (High-level Synthesis, HLS) 是一种自动设计过程，用于解释所需行为的算法描述并创建数字硬件，从而实现该行为，将行为与时序（例如时钟级别）解耦。HLS可以提高系统设计的抽象水平，使用寄存器传输级别RTL。HLS工具是Xilinx发布的与Vivado套件集成在一起的IP核开发工具，HLS将高级语言（C, C++等）编写的函数视为功能模块（IP核），函数相当于模块功能的RTL描述，高级语言中的函数调用等效于VHDL等电路描述语言中的模块实例化。此方法可以减少编写的代码量，从而显著简化用于系统描述的结构代码，最终加快系统组装进程。

本文通过使用改进输入结构的蝶形算法，从FFT的计算过程以及FPGA的硬件架构出发，利用输入重

映射模块实现FFT的并行输入，同时将蝶形运算系数初始化进IP核中。利用HLS实现IP核综合，IP核设计框架如图4所示。

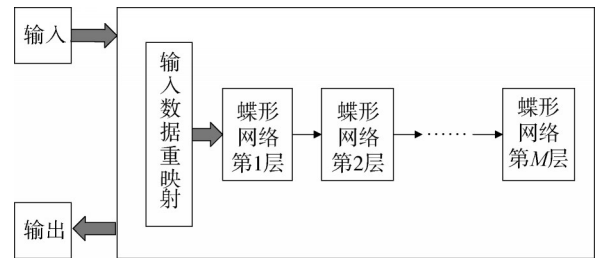


图4 IP核设计框架

Fig.4 IP kernel design framework

数据输入重映射模块和数据输出模块封装为FPGA上的RAM通信接口，数据输出模块封装为8路RAM通信接口。

数据输入重映射模块的输入数据大小为 1×32 ，其中，8个分为一组进行并行输入，所述蝶形运算网络模块中包括 $\log_2 N$ 个FFT蝶形网络，其中， N 是FFT运算的点数。所述蝶形运算网络模块中，第 i 个FFT蝶形网络的输入数据输入后，数据维度不变，输出数据的计算间隔为 $2i$ ， $i=1, 2, \dots, \log_2 N$ 。

蝶形运算网络模块包括旋转因子运算尺寸等参数的结构化、计算模块程序编写、程序按指令优化。参数结构化，将设计好的旋转因子权重参数一层一层地提取出来，分别存储在FPGA的查找表中，再将所得到的参数进行 $(N, 0)$ 的定点量化；计算模块的程序编写，根据参数预输入尺寸在HLS开发工具中分别编写程序，一层对应一个循环。 $\log_2 N$ 个循环用于通过蝶形运算来计算基2-FFT，最后一个阶段用于存储 N 点FFT的输出，将时域的 N 点序列转换为频域的 N 点复数序列。

计算模块程序编写，是在HLS中对FFT蝶形运算网络中的第1层、第2层与第3层都编写了单独的计算模块，每一个计算模块都在HLS进行了优化，包括流水线、数组分割与函数流水化等。其中，计算模块包含了旋转因子的相乘与累加模块，为了使效率最大化，设计了输入数据重映射模块，该模块以查找表的结构进行调度，保证输入数据的并行优势达到最大程度的利用，在合理利用 $\prod_2^{\log_2 N} \log_2 N$ 尺寸的内存空间的同时，在现实资源约束的前提下最大化展开运算函数，最大限度地减少了运算时间。所述蝶形运算网络模块的参数权值、旋转因子和偏差值以FPGA的硬件架构

进行结构化, 并进行 $(N, 0)$ 的定点量化。所述旋转因子分为实数集合和虚数集合, 设置为查找表的形式嵌入到FPGA中。

基于FPGA的输入分块重映射FFT方法IP核设计框架, 包括以下步骤:

a) 数据输入, 调整输入信号尺寸大小, 提取信号的采样点, 将提取的采样点进行分块重映射到蝶形网络运算单元中;

b) 蝶形网络运算, 该蝶形网络运算包括 $\log_2 N$ 个计算层;

c) 并行数据输出, 输出FFT变换的最后复数结果。

其中, 步骤a分为以下几个步骤:

1) 参数权值结构化, 将FFT卷积网络的权重参数一块一块地重新映射, 再将得到的参数进行FPGA架构下的定点量化;

2) 输入数据重映射表计算, 如图2所示, 将并行的、顺序的输入数据映射为FFT输出的数据;

3) 映射表模块程序代码编写, 根据定点量化后的参数在HLS工具中进行映射模块代码编写、优化, 每一个映射模块按照地址进行RAM读入。

步骤b分为以下几个步骤:

1) 旋转因子余弦和正弦系数结构化, 将旋转因子 w_N^k 的系数进行计算, 利用正弦函数的周期性, 在复数域呈现中心对称, 所占用内存空间地址由 N 下降到 $N/2$ 。然后匹配FPGA的架构进行系数的架构化, 转换为ap_fifo的格式, 该格式的输入数据将输入后级的蝶形运算;

2) 时间抽取基数2-FFT将DFT递归划分为偶数索引和奇数索引时间样本的两个半长DFT, 这些较短的FFT的输出可重新用于计算许多输出, 从而大大降低总计算成本;

3) 蝶形因子的计算公式为

$$e^{-j\frac{2\pi}{N}(k-M)} = e^{-j\frac{2\pi}{N}k} e^{-j\frac{2\pi}{N}\frac{N}{2}} = e^{-j\frac{2\pi}{N}k} e^{-j\pi} = e^{-j\frac{2\pi}{N}k} \quad (8)$$

式中 k 为输入数字信号的时间序号; M 为蝶形运算的分解数, 一般地, $M=N/2$ 。

当按照基2进行时间抽取时, 计算复杂度降低。

旋转因子余弦和正弦系数如表1所示, 可以看到输入数据进行FFT的具体框架:

a) 输入 1×32 数大小的信号, 8个一组为并行;

b) 将数据传入图1中的数据重映射层, 映射为FFT所需要的输出顺序;

c) 将输入数据输入蝶形运算层1, 数据维度不

变, 输出数据的计算间隔为2;

d) 将输入数据输入蝶形运算层2, 数据维度不变, 输出数据的计算间隔为4;

e) 将输入数据输入蝶形运算层 i , 数据维度不变, 输出数据的计算间隔为 $2i$;

f) 输出并行的数据。

表1 旋转因子余弦和正弦系数

Tab.1 The cosine and sine coefficients of rotation factors

余弦系数				正弦系数			
1.0000	0.9808	0.9239	0.8314	0	0.1951	0.3827	0.5555
0.7071	0.5555	0.3826	0.1951	0.7071	0.8315	0.9239	0.9801
0.6123	-0.1951	-0.3827	-0.5555	1.0000	0.9808	0.9239	0.8315
-0.7071	-0.8315	-0.9239	-0.9808	0.7071	0.5555	0.3827	0.1951

4 试验结果与分析

在Xilinx公司V7系列芯片xc7vx690t上进行仿真试验, 时钟周期设置为10 ns, 不确定度设置为0.13 ns, 分别利用官方IP核与本文提出的IP核进行32点FFT, 比较两者占用资源与计算速度的差异。

根据第3节中的方法概述, 编写C++源文件, 利用HLS综合生成IP核, 其接口如图5所示。输入信号数据通过RAM的接口分别输入到IP核中。xin_0_address0~7为8路并行输入的地址线; ce0和q0分别为接口使能和输入数据通道; xout_0_imag_address0和xout_0_real_address0分别为8路并行FFT实部虚部结果的输出地址; ce0、we0、d0分别为接口使能、写使能和输出数据通道。图5中其余的接口为IP核的相关控制接口, 系统默认端口类型是默认的ap协议端口类型, ap_clk同步采样时钟, ap_rst连接系统复位信号, ap_start是预留的起始信号, 输出的FFT计算结果在ap_done信号从低变高开始有效, ap_idle表示模块拉低不再空闲, ap_ready拉高表示系统可以接收新的输入。

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs		fftreturn value
ap_rst	in	1	ap_ctrl_hs		fftreturn value
ap_start	in	1	ap_ctrl_hs		fftreturn value
ap_done	out	1	ap_ctrl_hs		fftreturn value
ap_idle	out	1	ap_ctrl_hs		fftreturn value
ap_ready	out	1	ap_ctrl_hs		fftreturn value
xin_0_address0~7	out	3	ap_memory	xin_0	array
xin_0_ce0~7	out	1	ap_memory	xin_0	array
xin_0_q0~7	in	32	ap_memory	xin_0	array
xout_0_real_address0~7	out	3	ap_memory	xout_0_real	array
xout_0_real_ce0~7	out	1	ap_memory	xout_0_real	array
xout_0_real_we0~7	out	1	ap_memory	xout_0_real	array
xout_0_real_d0~7	out	32	ap_memory	xout_0_real	array
xout_0_imag_address0~7	out	3	ap_memory	xout_0_imag	array
xout_0_imag_ce0~7	out	1	ap_memory	xout_0_imag	array
xout_0_imag_we0~7	out	1	ap_memory	xout_0_imag	array
xout_0_imag_d0~7	out	32	ap_memory	xout_0_imag	array

图5 本文设计的IP核接口

Fig.5 The IP kernel interface designed in the paper

生成的IP核所消耗片内资源如图6所示, BRAM可以被配置为双口RAM增加读写速度, DSP48E是乘法器消耗的FPGA片内资源。数据事先写入RAM后, 每输入8组并行数据就等于输入一个地址进行查表, 找出地址对应的内容, 然后输出。官方FFT IP核所占用的资源如图7所示, 两者对比发现, 本文提出的方法使用的资源更多。

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	6143	-
FIFO	-	-	-	-	-
Instance	0	192	21432	16266	-
Memory	0	-	10	3	-
Multiplexer	-	-	-	2591	-
Register	0	-	14472	1184	-
Total	0	192	35914	26187	0
Available	2940	3600	866400	433200	0
Utilization (%)	0	5	4	6	0

图6 新方法的IP核占用的资源

Fig.6 Resource occupied by IP kernel of new method

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	6	-
FIFO	8	-	221	324	-
Instance	6	24	19796	16117	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	-	-
Register	-	-	-	-	-
Total	14	24	20017	16447	0
Available	2940	3600	866400	433200	0
Utilization (%)	~0	~0	2	3	0

图7 官方IP核占用的资源

Fig.7 Official resource occupied by IP kernel

分别利用上面生成的改进FFT IP核与传统顺序输入的Xilinx官方FFT IP核进行32点基2-FFT计算, 得到的两者的时序图如图8-9所示, 本文所设计的IP核计算32点FFT仅用101个时钟周期, 成倍小于官方的3195个时钟周期。

试验数据表明, 利用本文的方法处理32点FFT, 做到了并行化, 在10 ns时钟约束下, 处理时间达到了1.085 μs的延迟。

Latency (cycles)		Latency (absolute)		Interval (cycles)		Type
min	max	min	max	min	max	
101	101	1.085 us	1.085 us	101	101	none

图8 本文FFT IP核运算的时延分析

Fig.8 Delay analysis of FFT IP kernel in the paper

Latency (cycles)		Latency (absolute)		Interval (cycles)		Type
min	max	min	max	min	max	
3195	3195	10.543 us	10.543 us	3196	3196	dataflow

图9 Xilinx官方FFT IP核的时延分析

Fig.9 Delay analysis of Xilinx official FFT IP kernel

5 结束语

本文针对现有雷达干扰机数据处理能力不足的问题, 基于现有Xilinx官方FFT IP核与高速ADC速度不匹配的情况, 提出了一种基于FPGA的输入分块重映射的数据处理方法, 通过采用分块重映射策略并结合内置旋转因子表的方法, 改进了原有的“并行-串行”的计算流程, 极大地提高了使用FPGA对高速ADC数据进行FFT变换的速度, 实现了对雷达高速信号并行处理的目标。

通过32点FFT试验比较, 可以明显看出使用本文所提出的FFT方法构建的IP核时延成倍低于官方版本的FFT IP核, 其性能得到大幅度提升, 利用输入分块重映射的思想, 找到了一种资源消耗与性能提升相平衡的设计方法, 为提高雷达干扰机在数据处理速度方面提供了新思路。

参考文献

- [1] MELO R A, VALINOTI B. Serial QDR LVDS high-speed ADCs on xilinx series 7 FPGAs[C]. Buenos Aires: 2019 X Southern Conference on Programmable Logic (SPL), 2019.
- [2] BAI H Y, LI J M, GUO C. Design of LVDS high speed communication board based on PXI[J]. Instrument Technique and Sensor, 2020(5): 33-36.
- [3] VANMATHI K, SEKAR K, RAMACHANDRAN R. FPGA implementation of Fast Fourier Transform[C]. Coimbatore: 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCC), 2014.
- [4] NEWTON Fernandes, RAVINDRA Chaudhari, AJAY Khandare. FPGA implementation of GPR signal analysis using modified zoom FFT[C]. Kanpur: 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2019.
- [5] LIANG Dong. Design and implementation of base 2DIT-FFT butterfly operation based on FPGA[J]. Information and Communications, 2020(7): 41-43.
- [6] ZHANG D, CHEN L, WU Y. Research on high precision FFT algorithm based on FPGA[C]. Okinawa: 2020 5th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), 2020.

(下转第106页)

- Journal of Northwestern Polytechnical University, 2011, 29(3): 367-373.
- [17] 王生印, 龙腾, 王祝, 等. 基于即时修复式稀疏A*算法的动态航迹规划[J]. 系统工程与电子技术, 2018, 40(12): 2714-2721.
WANG Shengyin, LONG Teng, WANG Zhu, et al. Dynamic path planning using anytime repairing sparse A* algorithm[J]. Systems Engineering and Electronics, 2018, 40(12): 2714-2721.
- [18] 朱黔, 周锐. 具有持续侦查时间约束的协同航路规划[J]. 北京航空航天大学学报, 2015, 42(10): 2130-2138.
ZHU Qian, ZHOU Rui. Cooperative path planning with reconnaissance duration time constraints[J]. Journal of Beijing University of Aeronautics and astronautics, 2015, 42(10): 2130-2138.
- [19] Survivability/Vulnerability Information Analysis Center. Threat warheads & effects/battle damage assessment and repair (TWE/BDAR) training[J]. SURVIAC Bulletin, 1999, 25(2): 1-7.
- [20] 董保童, 刘振祥. 战伤诊断与修复计算机辅助系统[J]. 飞机设计, 2007, 27(4): 75-80.
DONG Baotong, LIU Zhenxiang. Computer assisted battle damage diagnosis and repair system[J]. Aircraft Design, 2007, 27 (4): 75-80.
- [21] 赵盼, 贾希胜, 胡起伟, 等. 战场损伤评估认知发展模型[J]. 计算机工程, 2007, 33(22): 206-208.
ZHAO Pan, JIA Xisheng, HU Qiwei, et al. Cognitive development model for battlefield damage assessment[J]. Computer Engineering, 2007, 33(22): 206-208.
- [22] 郭军, 宋建社, 曹继平, 等. 战场抢修资源重组决策方法[J]. 系统工程与电子技术, 2014, 36(2): 306-311.
GUO Jun, SONG Jianshe, CAO Jiping, et al. Battlefield urgent maintenance resource recombination decision making[J]. Systems Engineering and Electronics, 2014, 36(2): 306-311.
- [23] 郭军, 宋建社, 杨檬, 等. 基于证据理论的多任务抢修重要度决策[J]. 系统工程与电子技术, 2011, 33(3): 581-584.
GUO Jun, SONG Jianshe, YANG Meng, et al. Recovery importance decision making for multi-missions based on Dempster-Shafer theory[J]. Systems Engineering and Electronics, 2011, 33(3): 581-584.
- [24] 冯柯, 钟津, 严骏, 等. 工程装备战场抢修性分析[J]. 解放军理工大学学报(自然科学版), 2004, 5(2): 72-74.
FENG Ke, ZHONG Jin, YAN Jun, et al. Engineering equipment combat resilience[J]. Journal of PLA University of Science and Technology (Natural Science Edition), 2004, 5(2): 72-74.
- [25] 杜思良, 韩家启, 张永亮, 等. 陆军战术级作战任务智能规划技术研究[J]. 舰船电子工程, 2018, 38(12): 25-29.
DU Siliang, HAN Jiaqi, ZHANG Yongliang, et al. Research on intelligent operational task planning for army tactical forces[J]. Ship Electronic Engineering, 2018, 38(12): 25-29.
- [26] 胡葛庆, 胡雷, 陈凌, 等. 装备健康管理的现状、未来与挑战[J]. 国防科技, 2015, 36(1): 10-16.
HU Niaoqing, HU Lei, CHEN Ling, et al. Present future and challenge of equipment DPHM[J]. National Defense Science & Technology, 2015, 36(1): 10-16.
- [27] 邱立军. 武器装备故障预测与健康管理系统的关键技术[J]. 舰船电子工程, 2012, 32(5): 17-18.
QIU Lijun. Study of key technology for prognostics and health management system of weapon equipment[J]. Ship Electronic Engineering, 2012, 32(5): 17-18.
- [28] 马麟丽. 车辆装备保障能力评估方法研究[D]. 沈阳: 东北大学, 2013.
MA Linli. Study on evaluation method of vehicle equipment support ability[D]. Shenyang: Northeastern University, 2013.

作者简介

崔 蕾 (1971—), 女, 高级工程师, 主要研究方向为地面总体技术。
迟学谦 (1985—), 男, 高级工程师, 主要研究方向为飞行器设计。
张家骏 (1990—), 男, 工程师, 主要研究方向为车辆总体设计。
雷雪媛 (1990—), 女, 工程师, 主要研究方向为特种车辆总体设计。

(上接第90页)

- [7] MASSICOTTE D, JABER M A, NEILI C, et al. FPGA implementation for the multiplexed and pipelined building blocks of higher radix- 2^k FFT[C]. San Jose: 2020 IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS), 2020.
- [8] AKHIL R, KOLETI J R, BHASKAR A V, et al. Delay and area analysis of hardware implementation of FFT using FPGA[C]. Bangalore: 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 2020.
- [9] HOU Xiaochen, MENG Xiao, CHEN Hao. Design and implementation of mixed-radix FFT algorithm based on FPGA[J]. Journal of Terahertz Science and Electronic Information Technology, 2021, 19(2): 303-307.
- [10] 丛玉良. 数字信号处理原理及其MATLAB实现[M]. 第3版. 北京: 电子工业出版社, 2015.
CONG Yuliang. The principle and MATLAB implementation of digital signal processing[M]. Third Edition. Beijing: Publishing House of Electronics Industry, 2015.

作者简介

薛晓琴 (1992—), 女, 博士研究生, 主要研究方向为数字信号处理和移动自组网技术。
孟 刚 (1963—), 男, 研究员, 主要研究方向为雷达电子战技术和飞行器设计。
陈 旸 (1977—), 男, 高级工程师, 主要研究方向为导弹总体技术。
李宏博 (1980—), 女, 副研究员, 主要研究方向为智能信息处理, 多源融合认知等。
赵 健 (1989—), 男, 博士研究生, 主要研究方向为智能识别技术。