

引用格式:陈锋,张磊,宫永生.基于 SOC 的三模冗余纠错系统设计与实现[J].微电子学与计算机,2019,36(7):54-58. [CHEN F, ZHANG L, GONG Y S. Design and realization of TMR error correction system based on SOC[J]. Microelectronics & Computer, 2019, 36(7): 54-58. ]

## 基于 SOC 的三模冗余纠错系统设计与实现

陈 锋<sup>1,2</sup>, 张 磊<sup>2</sup>, 宫永生<sup>2</sup>

(1 中国科学院大学, 北京 100049; 2 中国科学院空间应用工程与技术中心, 北京 100094)

**摘 要:** 针对航天领域逐渐开始使用的 SOC 器件的 DDR 缓存容易在空间应用环境下发生数据错误的问题, 本文设计并实现了一种三模冗余纠错系统, 目的是保证空间应用环境下 SOC 器件 DDR 缓存数据的正确性. 本系统通过硬件电路的形式进行三模冗余表决, 并把表决结果写回 DDR 相应的缓存地址空间. 既减少了对 CPU 资源的占用, 又提高了表决速度. 本设计提出了一种反馈纠错机制, 区别于一般的三模冗余能容错但不纠错的特性, 能快速、批量地进行数据纠错. 并通过 Xilinx 公司的 XC702 开发板验证, 当开发板的 DDR 注入错误数据后, 本设计可以成功地在开发板上进行系统容错并纠正源地址的错误数据. 达到了确保 SOC 器件 DDR 缓存正确性的目的.

**关键词:** SOC; TMR; 容错; 纠错

**中图分类号:** V19

**文献标识码:** A

**文章编号:** 1000-7180(2019)07-0054-05

## Design and realization of TMR error correction system based on SOC

CHEN Feng<sup>1, 2</sup>, ZHANG Lei<sup>2</sup>, GONG Yong-sheng<sup>2</sup>

(1 University of Chinese Academy of Sciences, Beijing 100049, China;

2 Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences, Beijing 100094, China)

**Abstract:** Aiming at the problem that DDR cache of SOC devices used in the aerospace field is easy to generate data errors in the space application environment, this paper designs and implements a three-mode redundancy error correction system to ensure the accuracy of DDR cache data in the space application environment. The system carries on the three-mode redundancy voting through the hardware circuit form, and writes the voting result back DDR corresponding cache address space. It not only reduces the CPU resource, but also increases the voting speed. In this design, a feedback error correction mechanism is proposed, which is different from the general three-mode redundancy, which can tolerate but not correct errors, and can quickly and in batches correct errors. It is verified through XC702 development board of Xilinx company. After the DDR of the development board is injected with error data, this design can successfully carry out the system fault tolerance and correct the error data of the source address on the development board. The accuracy of the SOC device DDR cache is ensured.

**Key words:** SOC; TMR; errors tolerance; errors correction

### 1 引言

在航天领域使用 SOC 器件存在一个很严重的问题, 即如何是保证 DDR 缓存的正确性. 本文采用的方法是: 对 DDR 缓存进行三模冗余 (Triple Modular Redundancy, TMR) 容错并纠错 DDR 源地址

错误数据<sup>[1]</sup>.

TMR 设计是电路设计领域应用广泛的容错方法, 它常用于一些重要的航天器件的容错加固设计. TMR 的核心是利用冗余去屏蔽已发生的故障对整个电路的影响<sup>[2]</sup>. 但是, 它不能对出错的模块进行修复. 当一个模块出错后, 我们只是将错误通过多数表

决器屏蔽,但是该模块的错误数据仍然存在.而且一般的 TMR 也不能对错误进行检测和定位,以便系统进行修复.如果出现错误得不到及时修复,那么当另外两个模块的其中一个的对应地址的数据再次出现错误时,TMR 将失效<sup>[3]</sup>.通过将 TMR 表决器的输出反馈到这三个模块中,可以在容错的基础上达到纠正错误的功能.

本系统充分利用 FPGA 运行的并行性,将 TMR 的数据表决和反馈纠正 DDR 中源地址错误数据移植到 FPGA 中执行,简化了软件执行流程,缩短了时间开销<sup>[4]</sup>.在本系统设计中,把 TMR 设计封装成 IP 核的形式,利用 DMA IP 核实现 TMR 的三个模块(在本系统中为 DDR 中三段缓存地址)的数据容错,以及将 TMR 表决器输出反馈写到这三个模块的源数据地址段中覆盖错误的源数据<sup>[5-6]</sup>.相对于一般的 TMR 设计,本系统能很好的解决 SOC 器件在空间应用环境下 DDR 缓存出错问题,并及时纠正错误缓存.

## 2 TMR 纠错系统硬件设计

### 2.1 系统硬件结构设计

本系统设计是通过在 FPGA 上使用硬件描述语言(Hardware Description Language, HDL)编程实现的.把 TMR 的功能封装成 IP 核,方便系统调用.

TMR 基本原理为:使用三个完全相同的模块存储或执行相同数据,由表决器以三中取二原则对其输出进行表决,以确定整个系统的输出. TMR 系统可以容忍一个模块发生故障而不影响整个系统运行的正确性<sup>[7]</sup>.本系统在 TMR 的基础上,把表决器的输出反馈回这三个模块的源地址段(即对应 DDR 中的三段缓存地址空间),以实现迅速的纠错.为了提高 TMR 的数据读取,以及表决后数据写回的速率.采用 DMA(Direct Memory Access) IP 核的方式来执行搬运数据工作,可做到快速的、批量的数据读写. TMR 单次处理数据块为 DDR 中的 2K 缓存数据,可以通过修改 TMR IP 核改变单次处理的数据块.本系统结构框图如图 1 所示.

本系统的数据总线采用 AXI4 总线,控制总线采用 AXI4\_LITE 总线.对于挂载在 DMA IP 核上的 FIFO(First Input First Output)模块,采用无地址信号的 AXI4\_Stream 总线<sup>[8]</sup>.对图 1 中的各个 IP 核的介绍如下:

(1)PS 为 SOC 架构 FPGA 板卡上的处理器.主

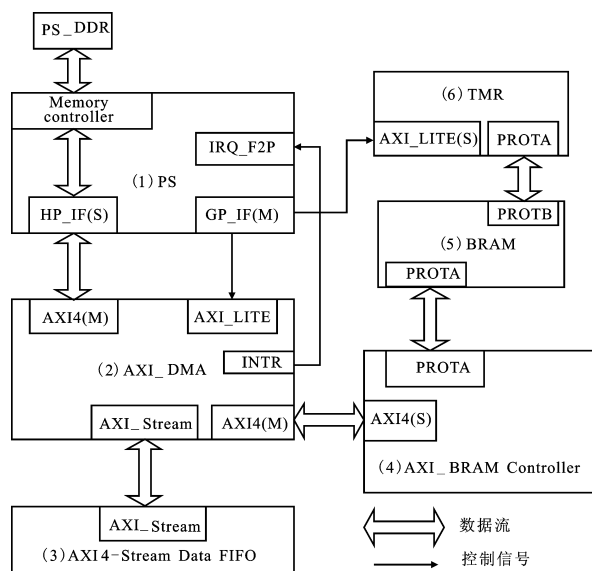


图 1 系统硬件结构框图

要负责 DMA IP 核的读写控制,以及使能和接收 TMR IP 核的反馈信号.

(2)AXI\_DMA 为 xilinx 官方 IP 核,主要负责 TMR 的三个模块运算数据的读取,以及把 TMR IP 核表决结果写回到 TMR 的三个模块.

(3)AXI4-Stream Data FIFO 为挂载在 DMA IP 核上的 FIFO 模块,用于 DMA 读写数据的缓存.

(4)AXI\_BRAM Controller 为 BRAM(Block Random Access Memory) IP 核连接到 AXI4 总线的控制器.用于 DMA IP 核通过 AXI4 总线对 BRAM IP 核进行读写操作.

(5)BRAM 用于 TMR 三个模块数据表决前的缓存以及表决结果的缓存.

(6)TMR 为三模冗余表决器 IP 核.它通过 AXI4\_LITE 总线接受处理器的控制,以及把表决完成标志位反馈给处理器.

本系统采用 Xilinx 的 XC7Z020-1CLG484-I 板卡作为实现平台.在此系统中 TMR 的三个模块即为挂载在 PS 端的 PS\_DDR 的三段缓存空间.系统采用的 BRAM 大小为 8k.前 6k 缓存空间分别存储来自于 PS\_DDR 三段缓存空间的各 2k 数据,由 DMA IP 核进行数据搬运,用于 TMR 表决.最后 2k 缓存空间用于暂存 TMR 表决输出结果,此结果由 DMA IP 核搬运写回 DDR 的三段缓存空间的相应的源数据地址段. TMR IP 核每次表决的数据块大小为 2k.

### 2.2 TMR IP 核设计及实现

由于 BRAM 作为 TMR 执行表决前后的数据

缓存,那么它就是 TMR 的数据交互对象,而处理器就是 TMR 的控制信号来源以及反馈对象.

BRAM IP 核设置为双端口类型,端口 A 用于 DMA 的读写操作,端口 B 用于 TMR 的读写操作. 设置 BRAM 读写数据宽度为 32 bits,设置操作模式为写优先. 把双端口的读延迟设置为一个时钟周期. 8k 的 BRAM 逻辑地址空间为 0x0000 ~ 0x1FFF. 0x0000~0x07FF 为 TMR 模块 1 的数据缓存空间,0x0800~0x0FFF 为 TMR 模块 2 的数据缓存空间,0x1000~0x17FF 为 TMR 模块 3 的数据缓存空间,0x1800~0x1FFF 为 TMR 表决结果的数据缓存空间.

在工程设计中,把 BRAM IP 核与 TMR IP 核封装在一起,形成具有缓存功能的 TMR IP 核,如图 2 中的 AXI\_TMR 就是此 IP 核结构框图. 图中 BRAM 的端口 A 要映射到 AXI\_TMR IP 核外,作为 DMA 的读写操作端口<sup>[9]</sup>.

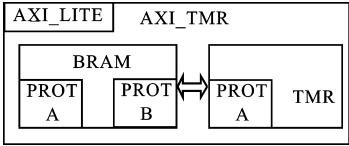


图 2 TMR IP 核结构框图

把 AXI\_TMR IP 核封装成 AXI\_LITE 总线形式,寄存器数量默认四个,但只用到两个即 slv\_reg0 和 slv\_reg1. 由于只用到 slv\_reg0 的写功能和 slv\_reg1 的读功能. 所以为了减少占用 FPGA 的逻辑资源,屏蔽掉 slv\_reg0 的读功能和 slv\_reg1 的写功能以及 slv\_reg2、slv\_reg3 的代码.

对于 slv\_reg0 和 slv\_reg1 寄存器,只用到它的第 0 位. 把 slv\_reg0 的第 0 位映射为 TMR 的状态机入口寄存器,slv\_reg0 则成为了 TMR 的使能寄存器,也就成为了 AXI\_TMR IP 核的使能寄存器. 把 TMR 完成 2K 数据操作的标志信号映射给 slv\_reg1 的第 0 位,当 TMR 完成操作时,标志信号输出高电平即二进制 1,则 slv\_reg1 的第 0 位也为 1. 当 PS 读取 slv\_reg1 的第 0 位为 1 时,便可以使能 DMA 读取 BRAM 中最后 2K 暂存的 TMR 表决结果,写回 DDR 的三段缓存空间,如表 1 所示.

表 2 为三模冗余真值表,根据此表得到 TMR 的表决公式为:  $ab+bc+ac=y$ . 在电路上表现为三路信号两两相与,得到的结果再经过或门就得到表决输出  $y$ <sup>[10]</sup>.

表 1 AXI\_LITE 总线四个寄存器使用说明

|          |                                                                                                |
|----------|------------------------------------------------------------------------------------------------|
| slv_reg0 | 使用 slv_reg0 寄存器的第 0 位作为 TMR 的状态机入口寄存器标志位,当第 0 位为 1 时,启动 TMR. 为 0 时,关闭 TMR.                     |
| slv_reg1 | 使用 slv_reg1 寄存器的第 0 位作为 TMR 完成 2K 数据操作的标志位,当第 0 位为 1 时,表明 TMR 完成 2K 数据操作. 为 0 时,表明未完成或未启动 TMR. |
| slv_reg2 | 已屏蔽                                                                                            |
| slv_reg3 | 已屏蔽                                                                                            |

表 2 三模冗余真值表

| 模块 a | 模块 b | 模块 c | 输出 y |
|------|------|------|------|
| 0    | 0    | 0    | 0    |
| 0    | 0    | 1    | 0    |
| 0    | 1    | 0    | 0    |
| 0    | 1    | 1    | 1    |
| 1    | 0    | 0    | 0    |
| 1    | 0    | 1    | 1    |
| 1    | 1    | 0    | 1    |
| 1    | 1    | 1    | 1    |

根据 TMR 的表决公式,以及 BRAM 的读写特性. 设计 TMR 状态流程图如图 3 所示. 其中 a. addr 为 BRAM 的逻辑首地址.

根据图 3 实现的 AXI\_TMR IP 核封装图如图 4 所示. 其中 clka、rsta、ena、wea、addra、dina、douta 信号为图 2 中 BRAM IP 核的 PROTA 的端口信号.

2.3 AXI\_TMR IP 核仿真

TMRIP 核的仿真采用 vivado 自带的仿真器. 以下为仿真文件设计思路,仿真输入时钟为 100 MHz.

(1)在 BRAM IP 核的 0x0000~0x17FF 地址空间造数. 由于 BRAM 的读写数据宽度设置为 32 bits,即每个数据用四个字节表示. 所以可以在地址空间 0x0000~0x07FF、0x0800~0x0FFF、0x1000~0x17FF 内分别连续写入 0~511、100~611、0~511.

(2)给 AXI\_LITE 接口的 slv\_reg0 寄存器赋值 1,等待 10 个时钟周期再赋值 0;然后等待 9 000 个时钟周期,接着循环赋值 1 和 0;如此反复三次. 当 slv\_reg0 寄存器第 0 位赋值 1 时,便开始使能 AXI\_TMR IP 核. 等待 9 000 个时钟周期是为了让 AXI\_TMR IP 核有足够的时钟周期完成 2k 数据的操作,实际上 TMR 完成 2k 数据需要的时钟周期为  $12 \times$

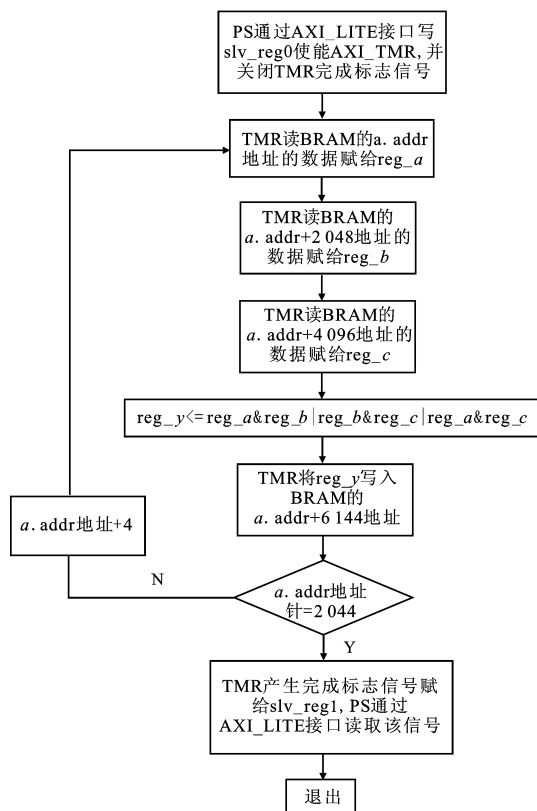


图 3 TMR 状态流程图

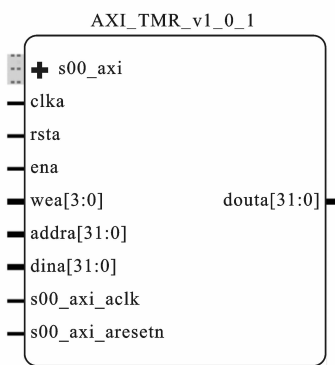


图 4 AXI\_TMR IP 核封装图

512 个时钟远小于 9 000,12 表示 TMR 的状态机的状态个数,512 表示对 2K 数据共做 512 次 TMR 操作.图 5 为 AXI\_TMR IP 核仿真图的部分截图.

### 3 TMR 纠错系统软件设计

本系统基于 SDK 软件平台进行软件应用程序设计.在完成图 1 的系统硬件结构设计,并生成比特流后,导入 SDK 中形成硬件平台封装文件.之后,在新建的工程里开始本系统的应用程序设计.图 6 为本系统应用程序设计框图<sup>[11-12]</sup>,在图 6 和具体设计步骤的说明中,TMR 的三个模块(模块 1,2,3)即为

DDR 中的三个地址段(地址段 1,2,3).具体设计步骤如下:

(1) 构建一个 DMA 中断资源初始化函数,以使能 DMA 中断.当 DMA 正确地完成数据搬运时,发送通道和接收通道的中断标志信号(TxDone 和 RxDone)将变 1,如果 DMA 搬运数据出错,中断返回信号 Error 将变 1.

(2) 宏定义 TMR 的三个模块地址空间,该地址空间即 DDR 中的三个地址段,每个模块存储空间为 8k.再定义一个(2k+4)Bytes 的地址空间用于缓存 DMA 搬回 DDR 的 BRAM 最后 2k 数据+BRAM 最后 2k 的最后 4 个 Bytes.

定义这个缓存地址空间的理由如下:根据 BRAM 的特性,DMA 从 BRAM 的 0x1800~0x1FFF 地址空间读数据时,读到的第一个数据是 TMR 模块做完三模冗余后通过 BRAM 端口 B 写入 BRAM 的最后一个数据即写入地址空间 0x1FFC~0x1FFF 的 4 个 Bytes 大小的数据.所以在该缓存空间中应该把 DMA 从 BRAM 搬运的第一个数据放置到最后.然后再从此缓存的基地址+4 Bytes 处开始,通过 DMA 把 2k 的数据搬到 DDR 的三个地址段的相应的地址空间中.

(3) 初始化 TMR 的三个模块,分别往模块 1、2、3 中造数 0~2 047,100~2 147,0~2 047.模块 2 初始化的数据为注入的错误数据.

(4) 分别把 TMR 的三个模块中的 2K 数据即模块 1 的 0~511,模块 2 的 100~611,模块 3 的 0~511.通过 DMA 搬运到 BRAM 中的 0x0000~0x07FF,0x0800~0x0FFF,0x1000~0x17FF 地址空间.当 TMR 模块做完 2K 数据的三模冗余操作并将表决结果存入 BRAM 的 0x1800~0x1FFF 地址空间后,DMA 从该地址空间把 2k 数据搬运回模块 1、2、3 的相应地址(即 DDR 中三个地址段的相应地址).然后,再循环地从 TMR 的三个模块的下一个 2K 地址空间搬运数据到 BRAM 中,如此循环,直到把 TMR 三个模块中的数据表决完.

(5) PS 通过 AXI\_LITE 接口写入 slv\_reg0[0] 一个脉冲.当 TMR 模块接收到此脉冲上升沿的下一个周期开始工作,然后让 CPU 监听 AXI\_LITE 接口的 slv\_reg1[0] 的状态,当 slv\_reg1[0] 变为 1 时,表示 TMR 已经将 2k 数据的表决结果写入 BRAM 的 0x1800~0x1FFF 地址空间.此时,CPU 再次启动 DMA,把表决结果搬运回 TMR 的三个模块的相应地址空间中,以纠正相应的错误数据.如此循环,

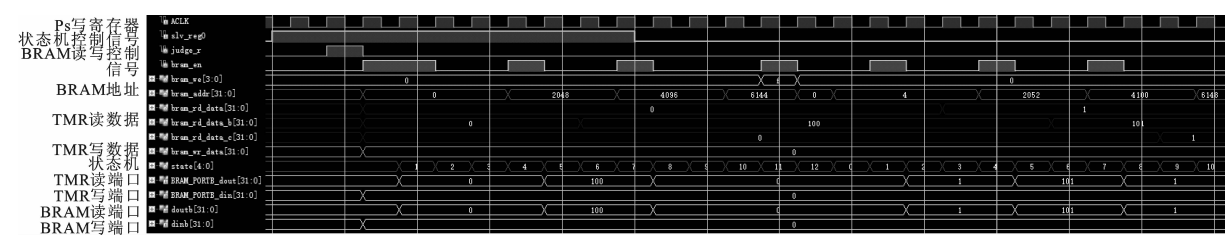


图 5 AXI\_TMR IP 核仿真图

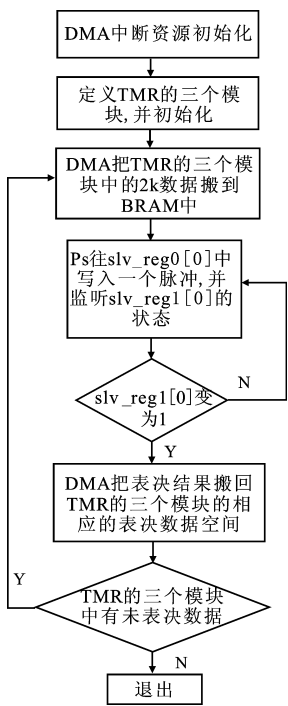


图 6 应用程序设计框图

直到把 TMR 三个模块中的数据表决完。

4 板子上验证

验证平台为 Xilinx XC7Z020-1CLG484-I. 通过串口调试工具将表决的结果打印出来. 实验结果表明, 虽然, 模块 2(即 DDR 地址段 2)为注入的错误数据, 但因为模块 1 和 2 数据完全相同, 所以表决结果正确. 而且, 在表决结果反馈给 DDR 的三个地址段 1, 2, 3 后, 再去读取地址段 2 的数据, 发现错误的的数据被纠正了. 实验结果表明, 本系统既屏蔽了 DDR2 的错误数据, 同时在每次表决结束后又快速地纠正了模块 2 的错误数据.

5 结束语

在当前航天领域逐渐使用 SOC 器件的背景下, 本文设计并实现了一种三模冗余纠错系统, 经过开发板验证, 证明该系统在屏蔽 DDR 错误数据的同

时, 可以及时快速地、批量地纠正 DDR 缓存数据, 从而保证 SOC 器件 DDR 缓存的正确性.

参考文献:

[1] 谭宜涛, 杨海钢, 黄娟, 等. 基于关键路径的三模冗余表决器插入算法[J]. 电子与信息学报, 2012, 34(2): 487-492.

[2] 桂江华, 徐睿, 卓琳. 基于三模冗余架构的集成电路加固设计[J]. 中国电子科学研究院学报, 2013, 8(6): 643-646.

[3] 张超, 赵伟, 刘峥. 基于 FPGA 的三模冗余容错技术研究[J]. 现代电子技术, 2011, 34(05): 167-171.

[4] 王猛, 岳素格, 赵元富, 陆振林. 一种改进的三模冗余任务同步方法[J]. 微电子学与计算机, 2015, 32(9): 177-180.

[5] 杨玉辰, 周国昌, 巨艇, 等. 三模冗余反馈纠错技术在星载电路加固设计中的应用与实现[J]. 空间电子技术, 2017, 14(2): 34-37.

[6] 王豪, 刘博, 张旭光. 一种面向航天应用的高可靠 FPGA 设计架构研究[J]. 航天控制, 2015, 33(6): 64-69.

[7] 韩月涛, 潘伟萍, 杨帆, 等. 基于 FPGA 的三模冗余 UART 电路设计[J]. 电子测量技术, 2011, 34(3): 57-61.

[8] ds190-Zynq-7000 SoC Data Sheet Overview v. 1. 11. 1[EB/OL]. 2018. [http://www.xilinx.com/suport/documentation/data\\_sheets/ds190-Zynq-7000-Overview.pdf](http://www.xilinx.com/suport/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf).

[9] AO R, CHEN Q Q, LI Z W, et al. Multi-objective evolutionary design of selective triple modular redundancy systems against SEUs[J]. Chinese Journal of Aeronautics, 2015, 28(3): 804-813.

[10] SHUBHAM C ANJANKAR, MAHESH T KOLTE, AJINKYA PUND, et al. FPGA based multiple fault tolerant and recoverable technique using triple modular redundancy (FRTMR)[J]. Procedia Computer Science, 2016, DOI:10.1016/j.procs.2016.03.109.

[11] ug702-Partial Reconfiguration User Guide v14. 5[EB/OL]. 2013. [http://www.xilinx.com/suport/documentation/data\\_manu/xilinx14\\_7/ug702.pdf](http://www.xilinx.com/suport/documentation/data_manu/xilinx14_7/ug702.pdf).

对比检测后的图片,如图 11 所示,YOLOV3 错误识别边缘残缺零件,同时误将背景识别为零件,YOLOV3-AEEF 仅识别出完整零件,不存在将背景识别为零件的情况.

从表 2 中可以看到,YOLOV3-AEEF 算法与其他的算法相比,查准率、查准率不是最高,但  $F_1$  分数较标准 YOLOV3 提升 10.5%,同时远超其他算法.综上,所提出的自适应边缘误差函数与改进 PSO 算法共同作用,可有效提升网络的综合性能.

5 结束语

本文在 YOLOV3 网络的基础上,设计自适应边缘误差函数,与改进 PSO 算法结合,得到 YOLOV3-AEEF 算法.训练过程加载预训练的权重,以防梯度爆炸.实验结果表明,当样本图片边缘存在较多残缺零件干扰时,YOLOV3-AEEF 网络仍能够准确识别目标,减少边缘区域的误识别,在保证较高查全率的情况下,查准率较 YOLOV3 提高 21%,提升了网络的综合性能.

参考文献:

[1] 翟俊海,张素芳,郝璞.卷积神经网络及其研究进展[J].河北大学学报(自然科学版),2017,37(6):640-651.

[2] 周飞燕,金林鹏,董军.卷积神经网络研究综述[J].计算机学报,2017,40(6):1229-1251.

[3] 李庆利,张帆,韩忠义,等.零件识别图像预处理算法研究[J].组合机床与自动化加工技术,2014,10(10):

63-65.

[4] 郑志强,刘妍妍,潘长城,等.改进 YOLOV3 遥感图像飞机识别应用[J].电光与控制,2018,6(9):15-19.

[5] 吴志洋,卓勇,廖生辉.改进的多目标回归实时人脸检测算法[J].计算机工程与应用,2018,54(11):1-7.

[6] 周志华.机器学习[M].北京:清华大学出版社,2015.

[7] 陈雪芳.小波分析和粒子群优化神经网络的语音端点检测[J].微电子学与计算机,2013,30(9):94-97.

[8] 彭刚,杨诗琪,黄心汉,等.改进的基于区域卷积神经网络的微操作系统目标检测方法[J].模式识别与人工智能,2018,31(2):142-149.

[9] 王联合国,洪毅,赵付青,等.一种模拟退火和粒子群混合优化算法[J].计算机仿真,2008,25(11):179-182.

[10] CHEN Y, YURI OWECHKO, SWARUP MEDASA-NI. Method for particle swarm optimization with random walk. US08793200B1[P]. 20140729.

[11] KAO Y T, ERWIE ZAHARA. A hybrid genetic algorithm and particle swarm optimization for multimodal functions[J]. Applied Soft Computing, 2007, 32(10): 849-857.

作者简介:

刘学平 男,(1965-),副教授,硕士生导师.研究方向为智能控制.

李珂乾(通讯作者) 男,(1995-),硕士研究生.研究方向为智能算法、深度学习. E-mail:lyq19952238@163.com.

刘 励 女,(1996-),硕士研究生.研究方向为智能算法.

王 哲 男,(1995-),硕士研究生.研究方向为深度学习.

(上接第 58 页)

[12] pg162-logicover IP AXI Master Burst v2.0[EB/OL] 2013. [http://www.xiilinx.com/suport/documentation/axi\\_matster\\_burst/v2\\_0/pg162](http://www.xiilinx.com/suport/documentation/axi_matster_burst/v2_0/pg162)<http://www.xiilinx.com/suport/documentation/axi-master-burst.pdf>.

作者简介:

陈 锋 男,(1992-),硕士研究生.研究方向为 FPGA 部分动态重配置. E-mail:chenfeng16@csu.ac.cn.

张 磊 男,(1971-),硕士,高级工程师.研究方向为卫星通信.

宫永生 男,(1984-),博士,副研究员.研究方向为空间数据存储.