ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 37,No. 3,May 2022,pp. 695－702
DOI:10. 16337/j. 1004-9037. 2022. 03. 018

http://sjcj. nuaa. edu. cn
E-mail:sjcj@nuaa. edu. cn
Tel/Fax:＋86-025-84892742

ZUC-256流密码轻量级硬件设计与实现

李　沐[1]，崔益军[2]，倪子颖[2]，王成华[2]，刘伟强[2]

(1. 北京卫星信息工程研究所,北京 100086; 2. 南京航空航天大学电子信息工程学院/集成电路学院,南京 211106)

摘　要： ZUC-256是由中国开发的一种应对于5G通信和量子计算机的流密码,该算法主要包含ZUC-256流密码和一种基于该流密码的完整性算法(EIA3)。本文设计了2种不同的ZUC-256流密码轻量级电路结构,以及1种基于ZUC-256流密码的EIA3算法结构。基于FPGA对设计的电路结构和算法结构进行实现,并进行了性能对比。对比结果表明:本文设计的2种电路结构最高达到了6.72 Gb/s的吞吐率,相较于现有的ZUC-256电路设计在速度上提高了45.24%;本文设计的2种电路相较于之前的ZUC-128占用资源更少,在面积上分别减少38.48%和30.90%;本文设计的EIA3算法结构仅用0.71 μs即可对128位的数据进行加密。

关键词： 流密码;ZUC-256;FPGA;硬件安全;轻量级加密

中图分类号： TN918　　**文献标志码：** A

Lightweight Hardware Design and Implementations of ZUC-256 Stream Cipher on FPGA

LI Mu[1], CUI Yijun[2], NI Ziying[2], WANG Chenghua[2], LIU Weiqiang[2]

(1. Beijing Institute of Satellite Information Engineering, Beijing 100086, China; 2. College of Electronic and Information Engineering/College of Integrated Circuits, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)

Abstract: ZUC-256 is a stream cipher developed in China for 5G communication and post-quantum, which mainly includes the ZUC-256 stream cipher and the integrity algorithm (EIA3). This paper designs two kinds of hardware structures of ZUC-256 stream cipher and an EIA3 algorithm structure based on ZUC-256. And then the designed structures are implemented based on PFGA, and their performance is compared. Comparison results show that: The two new ZUC-256 designs reach a throughput of 6.72 Gb/s, which is 45.24% faster than the current ZUC-256 design, and they uses fewer resources than the previous ZUC-128 design, reducing the area by 38.48% and 30.90%, respectively. And the EIA3 algorithm based on ZUC-256 can complete encryption of 128 bit data within 0.71 μs.

Key words: stream cipher; ZUC-256; FPGA; hardware security; lightweight encryption

---

基金项目：国家自然科学基金(62134002,62104107);江苏省自然科学基金(BK20210287);中央高校基本科研业务费专项资金(NS2021031)。

收稿日期：2022-01-20;**修订日期：**2022-03-22

引 言

流密码是一种对称性的密码算法[1],它可以很容易地在硬件中实现并且得到很高的性能。由于流密码在加密和解密过程中速度快,并且在传播过程中误差小,所以经常被用于无线通信。ZUC-128[2]和ZUC-256[3]都是流密码。ZUC-128算法能够提供128位的安全性,但由于下一代5G通信和量子化的发展,ZUC-128在不久的将来将无法满足要求。基于以上原因,作为ZUC-128的升级版本,ZUC-256算法于2018年被提出。ZUC-256算法一共包括2个子算法,分别是ZUC流密码(ZUC)和完整性算法(EIA3)[3]。ZUC-256算法的输入来自1个256位的密钥和1个184位的初始向量,在输出阶段每个周期输出1个32位的流密码。EIA3算法基于ZUC流密码产生的结果,可以通过判断信息的每1个比特来计算出1个Tag结果。

目前关于ZUC-256硬件实现的研究较少,而对于ZUC-128的硬件设计已经有了大量的研究,对于ZUC-256的硬件设计具有一定参考价值。Wang等[4]在2011年提出了3种不同的优化架构来实现ZUC-128,并且比较了这3种架构的整体性和复杂性;在2012年提出一种高吞吐率的结构[5],但其会消耗较多资源。2016年,一个基于ZUC-128的机密性算法(EEA3)架构被提出[6]。2020年Wang等[7]提出一种五级流水的ZUC-256流密码实现方案,其吞吐率达到了3.687 Gb/s。由于ZUC算法主要用于通信传输信息的加密,因此常用于资源受限设备,所以对于ZUC算法的低资源设计尤为关键。基于此,本文基于FPGA(Xilinx Kintex-7)架构设计并实现了2种ZUC-256的流密码结构和1种EIA3算法结构。

1 ZUC-256流密码算法和EIA3算法

1.1 ZUC-256流密码

ZUC-256流密码算法[3]包括3个逻辑层次:线性反馈移位寄存器(Linear feedback shift register, LFSR)、位重组(Bit recombination, BR)和有限状态自动机(Finite state machine, FSM),记为F。ZUC-256流密码将1个256位的原始密匙K和1个184位的初始向量$\boldsymbol{IV}$作为输入,并在每个循环内输出32位密码。

LFSR层有16个31位的向量$(s_0,s_1,\cdots,s_{15})$,并包含2个阶段,分别为初始化阶段和工作阶段。在初始化阶段,LFSR层的每个向量是由位常数D、256位初始密钥K和184位初始向量$\boldsymbol{IV}$组成。然后,LFSR通过将s_{15}、s_{13}、s_{10}、s_4、s_0合并计算,得到的结果与F层中的1个31位输入u相加得到新的s_0。u是通过将W向右移动1位得到,而W来自于有限状态自动机F。在16个向量向右移动后,s_{15}得到1个新的数值。工作阶段与初始化阶段的区别是在工作阶段时W将被设置为0。

BR层使用8个31位来自与LFSR层的向量来计算出4个32位的数据,分别为X_0、X_1、X_2和X_3。在ZUC-256中,选取s_0、s_2、s_5、s_7、s_9、s_{11}、s_{14}、s_{15}作为输入,生成的4个结果分别满足:$X_0=s_{15\mathrm{H}}\|s_{14\mathrm{L}}$;$X_1=s_{11\mathrm{L}}\|s_{9\mathrm{H}}$;$X_2=s_{7\mathrm{L}}\|s_{5\mathrm{H}}$和$X_3=s_{2\mathrm{L}}\|s_{0\mathrm{H}}$。

有限状态自动机F包含2个存储单元R_1和R_2,分别来自于BR层的输出X_0、X_1、X_2和X_3,并且由这2个存储单元来生成新的W值。存储单元的初始值均设置为0。在FSM层中,需要通过加法、异或和S-box运算得到结果。在ZUC-256中,S-box的值与ZUC-128相比没有发生变化,依旧使用2个不同的S-box,每个S-box使用2次。

ZUC-256流密码在工作状态下的结构图如图1所示。图中S表示S-box的值,L_1、L_2表示异或运算。ZUC-256流密码每次运算需要进行32轮的初始化运算,前32轮的计算使用W移位1位得到新的16个向量组合来产生新的s_{15}向量,然后将LFSR层中的16个向量逐一向右移动。在第33轮中执行与前32轮相同的操作但是此时W的值应被设置为0。之后,每一轮的运算都可以连续产生32位流密码结果。

ZUC算法在初始化阶段不输出结果。

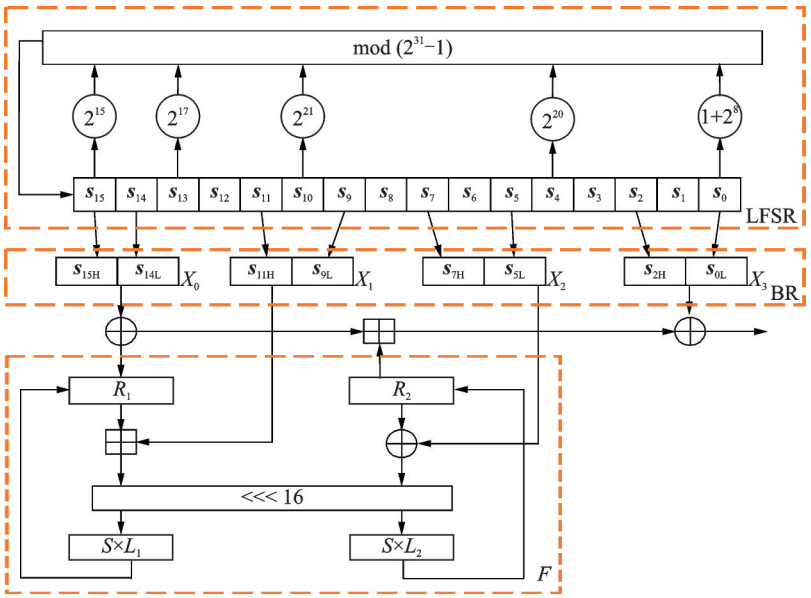

图1　ZUC-256流密码的工作阶段

Fig.1　Work stage of ZUC-256 stream cipher

1.2　EIA3算法

EIA3算法使用通用哈希算法原理生成Tag标签值。如同任何其他形式的哈希算法加密,EIA3算法可以将任意长度的信息转化为指定长度的标签值,信息的微小变化会导致Tag标签值发生显著变化,常用于数字签名中。

在EIA3算法中,根据信息中的每个比特值对Tag值进行迭代。对于ZUC-256的EIA3算法来说,如果当前信息的值为'1',那么新的Tag值就是旧的Tag值和32位流密码的异或结果,且用于计算的32位流密码每次只向右移动1位。当信息检索完成后,ZUC-256流密码再向右移动1位,并且执行同样的操作。在ZUC-256中,Tag值的长度一共有3个版本,分别为32位、64位和128位。Tag的初始值被定义为32位流密码的第1组,设置信息M的长度为L。EIA算法过程如算法1所示。

算法1　EIA3算法

输入:流密码$Z=\{Z_0, \cdots, Z_{t-1}\}$;明文信息$M$;信息的长度$L$

输出:Tag

设置Tag$=\{Z_0, \cdots, Z_{t-1}\}$

for $i=0$ to $l-1$ loop

　　$W_i=\{Z_{t+i}, \cdots, Z_{i+2t-1}\}$

　　if $M_i=1$ then

Tag$=$ Tag$\oplus W_i$

　　End if

End loop

　　$W_l=\{Z_{t+l}, \cdots, Z_{l+2t-1}\}$

　　Tag$=$ Tag$\oplus W_l$

2 两种ZUC-256流密码和EIA3算法硬件结构

2.1 ZUC-256流密码硬件结构

本文研究目的是在硬件上对ZUC-256流密码进行高性能实现。在ZUC-256的结构中，整体结构运行的主要时延来自于LFSR层中的7个数相加取模运算。对于BR层来说，因为只需要改变LFSR层向量的位置，基本上不会产生很大的关键路径延时。有限状态机的FSM计算来自于加法运算、异或运算和S-box检索替换操作，所以F层的电路延迟比LFSR层小得多。因此，本设计的关键路径延迟是在LFSR层，为了实现ZUC-256的高性能设计，应该减少这部分延迟。

ZUC算法中的LFSR层中包含2个阶段，分别为初始化阶段和工作阶段。在初始化阶段，LFSR层需要计算

$$\left(2^{15}s_{15}+2^{17}s_{13}+2^{21}s_{10}+2^{20}s_4+2^8s_0+s_0+u\right)\bmod\left(2^{31}-1\right)=s_{15} \tag{1}$$

对于式(1)，可以看成计算多次$(A+B)\bmod(2^{31}-1)$，其中A、B表示2个31位输入数据。针对2个数的相加取模，图2中的2种加法器均可以完成以上操作。对于串行模加结构，一共使用2个31位加法器，第1个加法器对输入数据进行加法操作，第2个加法器对第1步得到的进位和加法和进行相加，得到模约减后的结果。该结构一共使用了2个加法器，关键路径同样是来自于2个级联的加法器。对于并行模加结构，同样使用了2个加法器，分别计算$A+B$和$A+B+1$，再通过$A+B$的加法器中产生的进位来选择最终加法结果。相较于串行模加结构，该加法器同样使用了2个加法器，但是第2个加法器的输入为3个，这会造成该次加法的关键路径增加，但是由于加法器是并行结构，关键路径显著少于串行模加结构。基于以上2种不同的加法器，本文设计了2种ZUC-256流密码的结构。

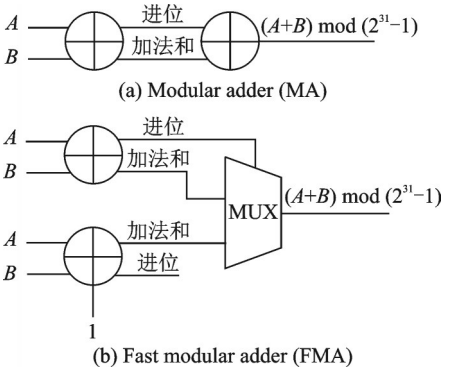

图2 ZUC-256中使用的2种加法器

Fig.2 Two different adders used in ZUC-256

2.1.1 ZUC-256结构1

在LFSR层的初始化阶段，一共需要对7个数字进行加法操作，而在工作阶段则需要加6个数字。因此，如果直接使用MA和FMA加法器，产生的电路关键路径延迟将不可忽略。但是对于多个数据相加，只需要在最后阶段计算出精确的数值，在中间阶段只需要尽量减少数据相加带来的关键路径延时即可。进位保存加法器(Carry save adder, CSA)是一种关键路径延时很小，并且可以将多个数据组合相加的加法器，同样该加法器无法得到最终的加法和。该加法器的原理是，进位和加法和的每1位都是独立产生的，当加法器的位数增加后，对于关键路径没有任何影响，只会增加加法器的资源面积，每次加法产生结果只需要几个门的延时，并且生成3个数据相加的进位和加法和。当输入的数据为3个时，将1个CSA加法器和1个普通加法器级联可以快速得到3个数据的加法结果。

使用这样级联的加法器可以快速完成7个输入数据的加法操作。在本设计中一共使用了包含CSA加法器的2层级联结构，在每1层中输入输出的长度被扩大4位。当所有的加法都完成后，再使用串行模加结构对数据进行模约减操作。由于进位和加法和的结果小于2^{31}，该结构只需要进行一次模约减操作。

ZUC-256结构1的LFSR层结构如图3所示，其中$A\sim F$分别表示31位输入数据。该结构中每轮计算一共使用3个时钟周期：在第1个时钟周期中除了$u=W\ll1$以外的6个数字将被相加，同时计算出u。在第2个时钟周期中将第1步得到的6个数据相加的结果再与u相加；在第3个周期中对16个寄存

器中的数据进行移位操作,最后得到新的s_{15}值。对于FSM层,需要进行加法、异或和S-box等操作。与BR层相比,FSM层的电路延迟较大。由于LFSR层在每个迭代中使用3个时钟,因此可以将F层的计算分解成3个时钟来完成。

当电路运行在工作阶段时,由于生成的密钥在每4个周期内只产生1次,在第1个周期和最后2个时钟周期中会产生不需要的值。因此,需要在第2和第3个时钟周期设置1个信号,以确保在计算EIA3算法时可以使用准确的密钥。

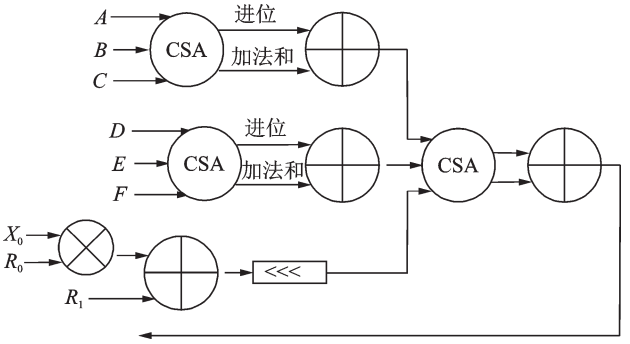

图3　ZUC-256结构1中LFSR层结构

Fig.3　LFSR layer architecture of ZUC-256 Structure 1

2.1.2　ZUC-256结构2

在ZUC-256结构1的设计中,如果每个CSA加法器的进位和加法和并不完全计算出来,等到最后时刻再计算,那么电路的关键路径将会极大减少。受文献[7]启发,本文设计了ZUC-256结构2,使用多个CSA加法器级联,并且由于单个CSA加法器的关键路径很低,在同一个周期中计算多个级联CSA加法器。

ZUC-256结构2的LFSR层结构如图4所示。在第1层中,该结构使用2个CSA加法器对于6个数据同时进行加法操作,这样产生了2个进位和2个加法和。对于前一级中CSA加法器产生的进位在后级中进行加法操作,需要将进位值向左移动1位。当6个数值的加法计算结束之后,首先将最后一级CSA加法器中的进位和加法和相加。这里的加法操作使用1次并行模加结构的加法器,因此得到的结果就是模约减后的结果。

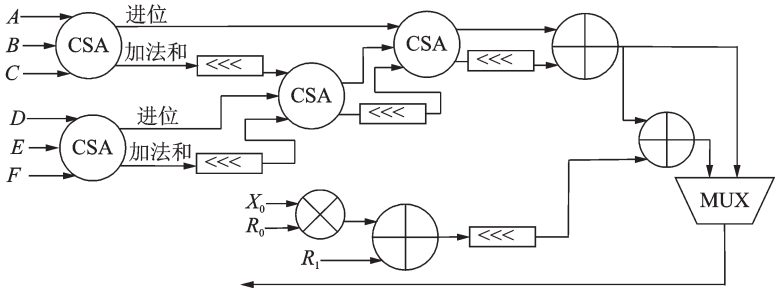

图4　ZUC-256结构2中LFSR层结构

Fig.4　LFSR layer architecture of ZUC-256 Structure 2

由于本文设计目的是通过达到更高的频率来提高ZUC-256流密码的吞吐率,因此对于电路结构的设计思路是在初始化阶段将LFSR层的计算分割成3个周期,但是在工作阶段由于W不参与计算,将其分割成2个周期。在初始化阶段,用第1个周期来计算3层级联的4个CSA加法器和u;在第2个周期中,使用1个并行模加加法器来计算6个输入数据之和的模约减;在第3个周期中,同样再使用1次并行模加加法器,并得到最终结果。在工作阶段,W值不再参与运算,此时只需要计算6个数字的总和,当第1个并行模加加法器计算完成后,这个结果便是最终结果。同样地,由于LFSR层的周期减少,使得该结构不能使用3个周期内计算BR层和FSM层。幸运的是,这2层的电路延迟很小,可以在2个周期内完成并且不影响整体运行的频率。对于这2层,在第1个周期计算$w_1=(R_1+X_1) \bmod 2^{32}$和$w_2=(R_2+$

X_2)mod($2^{31}-1$),然后重新组合,最后进行线性变换;在第2个周期中,该结构只进行S-box操作。

2.2 EIA3算法的硬件实现

EIA3算法的Tag值是基于ZUC-256流密码算法进行计算,并且相较于ZUC-128中的算法做出了较大改动。在本文设计中,EIA3的输出使用64位的Tag值。由于流密码每轮输出32位,Tag值的加载需要2个周期。从算法1可以看出,如果Tag的长度为64位,Tag的第1次计算将从第64位开始。所以该结构需要在第5个周期计算Tag值。由于Tag值的计算在每个迭代中只移动1位,这样在每个周期中只计算一次会拖慢运算速度,因此本文设计中单个周期可以对32位的信息进行处理。

因为流密码每次产生32位,而Tag值的长度是64位,在该结构中使用1个96位的寄存器来保存连续3个周期中产生的各32位流密码。每当产生1个新的32位流密码时,将该流密码的值从96位寄存器的低32位处输入,同时将该寄存器的高32位输出抛弃。如图5所示,每个信息被设计成1个选择器,图中W_0~W_i均表示64位寄存器。如果信息的i位为0,W_i寄存器将被设置为0;如果信息的i位为1,W_i寄存器将被设置为本周期内96位寄存器右移i位的最后32位。这样1个时钟周期可以并行处理32位消息,可以节省大量的计算时间。

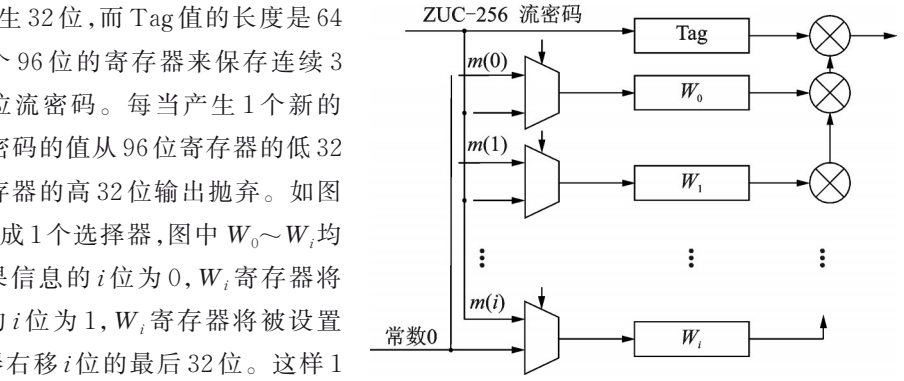

图5 EIA3算法的硬件结构

Fig.5 Hardware structure of EIA3 algorithm

需要注意的是,当最后一组信息进入该结构时,需要在信息的最后1位加上1个比特“1”,然后通过上述方法得到最终的迭代结果。最后,通过计算Tag初始值和最后的迭代结果来得到最终的Tag值。

3 ZUC-256流密码硬件实现与性能对比

本节使用FPGA对于ZUC-256的2种结构和基于ZUC-256的EIA3算法结构进行实现。所有设计均使用Vivado 2020.1进行综合,并且在Kintex-7 FPGA平台上进行实现,然后通过后仿真对其性能进行对比。由于ZUC-256相关的硬件设计结果较少,在本文中引入部分关于ZUC-256的软件设计结果进行对比。为了公平起见,对于ZUC-256流密码的设计,资源部分不包含EIA3算法资源。图6是包含EIA3计算的完整实现仿真图。图6中最开始进行ZUC的31轮初始化运算,之后产生连续的32位ZUC流密码。当第1个流密码产生后,多Tag进行初始化,然后单次对32位信息进行处理,最终得到Tag结果。

本文的ZUC-256硬件实现与对比结果如表1所示。由表1可见,结构1可以达到更高的频率,并且使用了更少的资源,但是由于其在4个周期中才会产生1次有效的结果,因此吞吐率只有4.78 Gb/s。而对于结构2来说,由于改进了其流水线结构,使用了更短的流水线,尽管在频率上不及结构1,但却获得更高的吞吐率,达到6.72 Gb/s。与现有其他ZUC-256的硬件设计相比,本文ZUC-256结构2吞吐率提升了45.24%。与其他基于RISC-V平台的ZUC-256结构相比,本文结构在吞吐率方面也有显著提升。

此外,本文还将ZUC-256的设计结果与ZUC-128设计结果进行对比,如表2所示。从表2中可以看出,本文设计的结构达到了更高的运行频率。这是因为本文结构分割了更多的流水线,使得整体结构上的关键路径减少。并且由于频率的提高,尽管本文设计的结构使用了更多的周期,但是吞吐率仅略

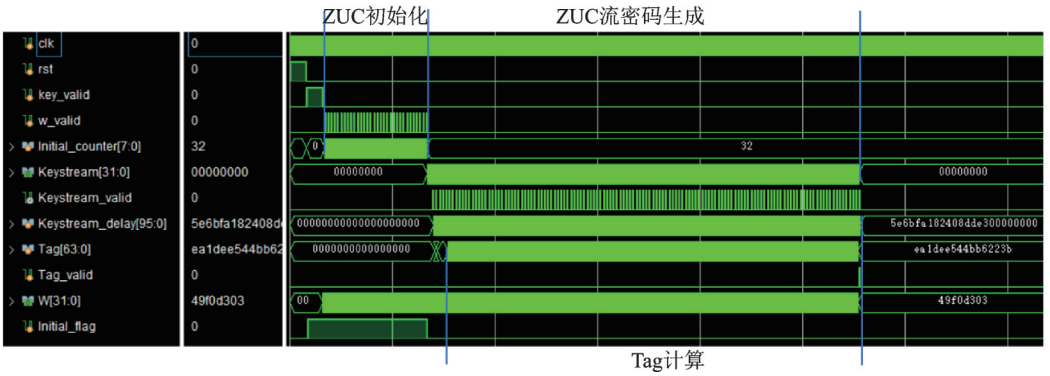

图6　ZUC-256 FPGA实现仿真时序图

Fig.6　Simulation timing diagram of ZUC-256 implementation on FPGA

表1　本文ZUC-256结构与其他ZUC-256结构性能对比

Table 1　Performance comparison of designed and previous ZUC-256 structures

结构	实现平台	频率/MHz	面积/FPGA片	吞吐率/(Gb•s^{-1})	吞吐率/面积/(Mb•(s•片)$^{-1}$)
文献[7]ZUC-256结构	DE2-115	115		3.68	
文献[9]ZUC-256结构	RISC-V平台			0.035 6	
本文ZUC-256结构1	Kintex-7	448	219	4.78	21.8
本文ZUC-256结构2	Kintex-7	420	246	6.72	27.3

表2　本文ZUC-256结构与其他ZUC-128结构性能对比

Table 2　Performance comparison of designed ZUC-256 structures and previous ZUC-128 structures

结构	实现平台	频率/MHz	面积/FPGA片	吞吐率/(Gb•s^{-1})	吞吐率/面积/(Mb•(s•片)$^{-1}$)
文献[4]ZUC-128结构	Virtex-5	108	356	3.46	9.7
文献[8]ZUC-128结构	Virtex-5	222	575	7.11	12.3
本文ZUC-256结构1	Kintex-7	448	219	4.78	21.8
本文ZUC-256结构2	Kintex-7	420	246	6.72	27.3

低于文献[8]的ZUC-128结构。同时本文结构使用了更少的资源,因此在吞吐率/面积方面相较于文献[4,8]的ZUC-128结构分别提升了43.58%和54.95%。对于ZUC-256结构1,在资源上相较于文献[4,8]的ZUC-128结构分别减少38.48%和61.91%。

本文设计的EIA3算法硬件实现主要使用了ZUC-256结构1产生的ZUC-256流密码,算法性能如表3所示,其中Tag位宽均为64位。从表3可以看出,基于ZUC-256的EIA3算法结构在资源上相较于单独的ZUC-256结构2仅增加37个Slice块,运行频率与ZUC-256结构1保持一致。此外,该结构使用流水线设计,一次性可以处理32位数据,对1次128位数据生成64位签名只需要0.71 μs。

表3　基于ZUC-256的EIA3算法的硬件实现及其性能

Table 3　Hardware implementation of EIA3 algorithm based on ZUC-256 and its performance

结构	平台	频率/MHz	面积/FPGA片	时间/μs
基于ZUC-256的EIA3算法结构	Kintex-7	448	283	0.71

4 结束语

本文设计了2种不同流水线的ZUC-256硬件实现电路:第1种结构使用三级流水线,使用了更少的资源,达到了更高的频率;第2种结构使用两级流水线,可以达到更大的吞吐率。与目前最快的ZUC-256流密码硬件设计相比,第2种设计在吞吐率上提升54.95%。本文还设计了一种基于ZUC-256的EIA3算法的硬件结构,仅仅使用37个Slice块的资源,对1次128位数据生成64位签名只需要0.71 μs。本文设计的ZUC-256结构使用了更少的资源,因此更加适合于资源受限的场合。未来的工作将着眼于ZUC-256在资源受限情况下的更高性能设计。

参 考 文 献:

[1] LÉGLISE P, STANDAERT F X, ROUVROY G. Efficient implementation of recent stream ciphers on reconfirgurable hardware devices[C]// Proceedings of the 26th Symposium on Information Theory in the Benelux.[S.l.]: [s.n.], 2005: 261-268.

[2] BLANK P. ETSI SAGE: Specification of the 3GPP confidentiality and integrity algorithms 128-EEA3[R]. Document 2: ZUC Specification, 2011-01-04.

[3] Design Team. ZUC-256 stream cipher[J]. Journal of Cryptologic Research, 2018: 167-179.

[4] LIU Zhongbin, ZHANG Lingchen, JING Jiwu, et al. Efficient pipelined stream cipher ZUC algorithm in FPGA[C]// Proceedings of the First International Workshop on ZUC Algorithm. China: [s.n.], 2010.

[5] ZHANG Lingchen, XIA Luning, LIU Zongbin, et al. Evaluating the optimized implementations of SNOW3G and ZUC on FPGA[C]// Proceedings of 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications. [S.l.]: IEEE, 2012: 436-442.

[6] FALAQ M, ABDULHAYAN S. LTE security: EEA-3 using ZUC algorithm[J]. International Journal of Innovative Research in Computer and Communication Engineering, 2016. DOI: 10.15680/IJIRCCE.2016.0407057.

[7] WANG Yuankai, WU Liji, ZHANG Xiangmin, et al. A hardware implementation of ZUC-256 stream cipher[C]// Proceedings of 2020 IEEE 14th International Conference on Anti-counterfeiting, Security, and Identification. [S.l.]: IEEE, 2020: 94-97.

[8] KITSOS P, SKLAVOS N, SKODRAS A N. An FPGA implementation of the ZUC stream cipher[C]// Proceedings of 2011 14th Euromicro Conference on Digital System Design. [S.l.]: IEEE, 2011.

[9] WEI Mengmeng, YANG Guoqiang, KONG Fanyu. Software implementation and comparison of ZUC-256, SNOW-V, and AES-256 on RISC-V platform[C]//Proceedings of 2021 IEEE International Conference on Information Communication and Software Engineering (ICICSE). [S.l.]: IEEE, 2021.

作者简介:

李沐(1982-),男,高级工程师,研究方向:通信安全技术,E-mail:lim@spacestar.com.cn。

崔益军(1988-),**通信作者**,男,博士,讲师,研究方向:硬件安全、物理不可克隆函数及后量子密码技术等,E-mail:yijun.cui@nuaa.edu.cn。

倪子颖(1997-),男,科研助理,研究方向:硬件安全、后量子密码,E-mail:nzy@nuaa.edu.cn。

王成华(1963-),男,教授,研究方向:信息安全芯片、物理不可克隆函数,E-mail:chwang@nuaa.edu.cn。

刘伟强(1983-),男,教授,研究方向:数字集成电路设计、混合信号集成电路设计,E-mail:liuweiqiang@nuaa.edu.cn。

(编辑:张黄群)