Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 107 (2017) 225 – 230

Procedia
Computer Science

International Congress of Information and Communication Technology (ICICT 2017)

Design and Implementation of Configurable SHIFT Instructions targeted at Symmetrical Cipher Processing

Longmei Nan[a,b,*] , Xiaoyang Zeng[a], Wei Li[a,b], Zhouchuang Wang[b]

[a]ASIC & System State Key Laboratory of Fudan University, Shanghai 201203, China
[b]Zhengzhou Institute of Information Technology, Zhengzhou 450004, China
** Corresponding author: 13110720052@fudan.edu.cn Tel.: 18638263779*

Abstract

High-performance and flexible configurable SHIFT instructions targeted at symmetrical cipher processing are proposed in this paper, in order to dispel the bottleneck of symmetrical cipher algorithms realized by universal processors. Through analyzing the processing characteristics and the structures of many symmetrical cipher algorithms, the proposed SHIFT instructions can support different processing data widths, different SHIFT modes. Furthermore, instruction level parallelism based on VLIW system structure and instruction inner parallelism by operating several sub word SHIFTs at the same time are designed too. Finally, corresponding reconfigurable hardware units to support the execution of each instruction forcefully is also exploited. For the characteristics of high efficiency and flexibility, the specific SHIFT instructions and the reconfigurable hardware processing units can be used as an ameliorative unit for processors to advance the performance in special processing for symmetrical cipher.

Keywords: Symmetrical Cipher; SHIFT; Configurable; VLIW; Parallel Process.

1. Introduction

Symmetrical cipher[1] has a far-ranging application future, which plays an important role in the security of cryptographic algorithms. It can be realized by a general processor with sufficient flexibility, but the operating performance is very low, such as SHIFT operation. So a processor with special instructions[2] is a crucial tide to implement symmetrical cipher, owing to its characteristics of high speed and high flexibility. For the high frequency of SHIFT being used in symmetrical cipher algorithms, its operating performance impacts the symmetrical cipher algorithms' performance extraordinary, so it is significant to propose special SHIFT instructions the without sacrificing flexibility.

Peer-review under responsibility of the scientific committee of the 7th International Congress of Information and Communication Technology
doi:10.1016/j.procs.2017.03.083

2. Analysis of SHIFT operation characteristics

2.1. Analysis of SHIFT operation characteristics in symmetrical ciphers

The SHIFT operation can be summarized as rotary SHIFT and logic SHIFT modes mainly. By analyzing about sixty public symmetrical cipher algorithms collected by NESSIE project and ECRYPT[3] project in this paper, it can be gained that SHIFT operation is used in thirteen symmetrical cipher algorithms(IDEA, RC5, RC6, Mars, Rijndael, CAST-256, CRYPTON, SERPENT, Twofish, AES, DES, 3DES, SAFER+), and each of SHIFTs used has different characteristic, such as the shift number , the shift data width, and the shift direction.

By analyzing, it can be known that the characteristic of SHIFTs used in each algorithm is multiform. Firstly, the data width of SHIFTs is different, but it can be sorted by five kinds haply: within 8bit, within 16bit, within 32bit, within 64bit,and within 128bit. Secondly, the SHIFTs modes are multiplex, they can be sum up as rotary left SHIFT, rotary right SHIFT, logic left SHIFT, logic right SHIFT, immediate variable SHIFT, register variable SHIFT, constant SHIFT, random SHIFT and so on. So we can gain the processing data width and different models of SHIFT operation, this can be used as significant foundation to propose the SHIFT instructions.

2.2. Analysis of SHIFT operation characteristics by universal processors

The instructions expended by SHIFT using universal processors depend on SHIFT's data width, shift number and shift mode intensively. Here the hint of instructions consumed by 64bit data rotary left SHIFT by a RISC universal processor is given in Fig. 1 to incarnate it. It needs 8 instructions (AND, SRL, AND, SRL, SLL, SLL, OR, OR) to finish 64bit data rotary left SHIFT. Furthermore, larger data width will consume more instructions, so it is necessary to design special SHIFT instructions according to corresponding characteristics in symmetrical ciphers.

3. Design of SHIFT instructions for symmetrical cipher processing

3.1. Design of basal SHIFT instructions

In order to design special SHIFT instructions, besides calculating the influencing factors of operation data width, and operation modes mentioned above, questions as operation data source, operation data destination and shift number data source should also be solved.

Firstly, the operation data width of different SHIFTs is between 8bit and 128bit typically, but 64bit and 128bit SHIFTs can be done by special series shift (SHLS or SHRS) proposed by this paper, so considering the consumption of hardware, the operation data width of SHIFT instructions proposed should sustain 8bit, 16bit 32bit, and the operation data source of SHIFT instructions designed should sustain universal registers. Secondly, the SHIFT number of different SHIFTs can't only come from universal registers, computed by other instructions, for register variable SHIFT and random SHIFT, but also come from immediate data, known in advance, for immediate data SHIFT and constant SHIFT. Thirdly, to symmetrical cipher algorithms, four unaided 8bit data shift are used as byte shift to a 32 bit data frequently, so sub-word parallel shift special instructions also should be proposed.

As mentioned above, several basal SHIFT instructions have been designed to process 8bit, 16bit, and 32bit data SHIFT respectively, which are described as follow:

ISHLm Rd, Rs1, #imm; ISHRm Rd, Rs1, #imm; IROLm Rd, Rs1, #imm; IRORm Rd, Rs1, #imm
RSHLm Rd, Rs1, Rs2; RSHRm Rd, Rs1, Rs2; RROLm Rd, Rs1, Rs2; RRORm Rd, Rs1, Rs2

In which, ISHLm, ISHRm, IROLm, IRORm, RSHLm, RSHRm, RROLm, and RRORm represent immediate logic left SHIFT, immediate logic right SHIFT, immediate rotary left SHIFT, immediate rotary right SHIFT, register logic left SHIFT, register logic right SHIFT, register rotary left SHIFT, register rotary right SHIFT respectively, and m represents shift operation data width, which value is 8, 16 or 32. If m=8, it means four byte shift to a 32 bit data, if m = 16, it means two 16bit shift to a 32 bit data, if m = 32, it means the 32 bit data shift in whole. Rs1 represents the source operand universal registers in which deposited the source shift data. Rs2 represents the source operand universal registers in which deposited the shift number data for register SHIFT instructions. Rd represents the