

FPGA-based SIMD Processor

Stanley Y.C. Li, Gap C.K. Cheuk, K.H. Lee and Philip H.W. Leong
{ycli,ckcheuk,khlee,phwl}@cse.cuhk.edu.hk
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, NT Hong Kong

Abstract

A massively parallel single instruction multiple data stream (SIMD) processor designed specifically for cryptographic key search applications is presented. This design aims to exploit fine grain parallelism and the high memory bandwidth available in an FPGA by integrating 95 simple processors and memory on a single FPGA chip. Performance is compared with a previously reported hardwired design on a RC4 key search application.

1 Introduction

Although field programmable gate arrays (FPGAs) can have high performance gains over equivalent microprocessor based systems, they have the disadvantage that the design time is much higher than for an equivalent software based system. In this paper, we present a parallel single instruction multiple data stream (SIMD) processor which aims to achieve high performance, yet be programmable in software so that the FPGA design need not be changed for different applications. The processor was designed so that it could be used to efficiently implement an RC4 key search engine [1], but is hopefully not limited to this application. Furthermore, compilation from a high level language to this machine should be possible.

2 Architecture

The processor has an instruction set which has been modified from the Microchip Technology PIC processor. It is organized in a Harvard architecture, the instruction width being 16-bits and the data width 8-bits. The implementation is organized in a 3-stage pipeline (fetch, decode and execute). Most instructions take 1 cycle but transfer of control and block RAM access instructions incur an extra cycle of delay and hence require software interlocking.

There are three types of data storage associated with the processor, all being 8-bits in width. W is a single register; the register file has a total of 16 registers, two (MARA and MARB) being special purpose registers for memory access instructions, RANK is used to identify the individual processors and the others are general purpose; and the dual port block RAM which is a 512-byte storage. The instruction set of the processor is summarized in Table 1 and the datapath is shown in Figure 1. Program memory is implemented in a single block RAM.

Mnemonic		Description	Cycles
Control operation			
NOP	-	No operation	1
SLEEP	-	Go into standby mode	1
BTAXSC	k	Compare AX skip if =	1(2)
BTCSC	-	Bit test C skip if clear	1(2)
BTESC	-	Bit test E skip if clear	1(2)
PORTIN	-	Load port input to W	1
Byte-oriented register memory operation			
ADDWF	f, d	Add W and f (set C)	1
CLRF	f	Clear f (clear C)	1
INCF	f, d	Increment f (set C)	1
XORWF	f, d	XOR W and f	1
MOVF	f, d	Move f	1
CMPWF	f	Compare W with f (set E)	1
Literal operation			
MOVLW	k	Move literal to W	1
GOTO	k	Unconditional branch	2
Memory operation			
MOVMMW	f, p	Move memory to W	2
MOVWM	f, p	Move W to memory	1
SWAP	f, p	Swap memory	2(1)

Table 1. Instruction set overview. ‘f’ is a register file designator ‘d’ is a destination designator, ‘k’ is an 8-bit literal value and ‘p’ represents bank 0 or 1 of the block RAM.

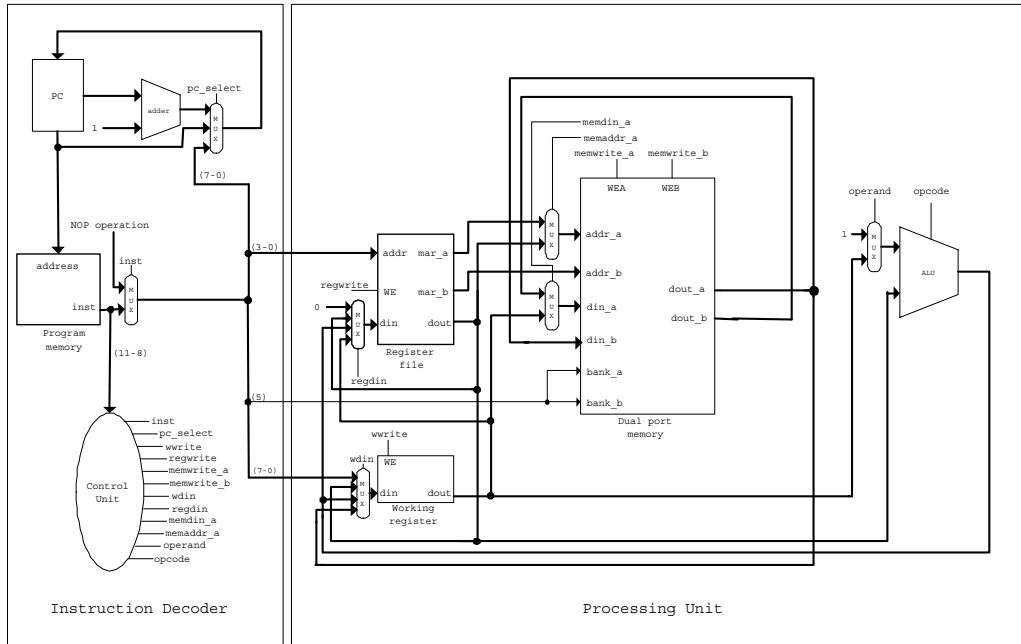


Figure 1. The overall datapath with control signals

3 Results

A SIMD processor consisting of 95 processor cores plus program memory was developed on a Xilinx Virtex 1000E-6 FPGA device, fully utilizing all available block RAMs on the device. The design occupied 9,210 slices (74% utilization) and had a maximum clock frequency of 68 MHz. The critical path is currently in the high fanout signals which distribute the instructions to the processor cores. For the RC4 key search application, no communications between cores are necessary.

An RC4 key search engine application was developed in assembly language for the SIMD processor. The RANK is the first key tested in each processing unit. If the 95 parallel searches fail, the processing unit adds 95 to the key to be tested and it is repeated. When the key is found, the RANK and iteration number are sent to an output port.

This implementation requires 14 cycles for initialization. Thereafter, 4,520 cycles are required to test a key, equivalent to 15,000 keys per second per processing unit. With 95 processors operating in parallel, the overall throughput is 1,425,025 keys per second. An equivalent FPGA design reported by our group where the RC4 datapath is completely hardwired achieved a throughput of 6 million keys per second [1] and a highly optimized software implementation on a 1.5 GHz Intel Pentium 4 processor achieves 100,000 keys per second. Thus the SIMD implementation was $14\times$ faster than the software based implementation on a 1.5 GHz Pen-

tium 4 and $4\times$ slower than a hardwired design.

4 Conclusions

An SIMD processor was applied to the RC4 keysearch problem and able to achieve a high level of parallelism as well as utilize the higher memory bandwidth available on the device. Although the design was $4\times$ slower than a hardwired design, the development time for an application using this machine is significantly lower since designers can adopt a purely programming based model and need not be concerned with lower level details such as datapath design, control design, place and route, design optimization etc. The SIMD approach also has benefits in that the design can be amortized over many different applications potentially resulting in a large overall savings in development effort. Finally, using this approach, there is potential to customize the instruction set of the processor as well as to add coprocessing elements to further accelerate applications.

References

- [1] K.H. Tsoi, K.H. Lee, and P.H.W. Leong. A massively parallel RC4 key search engine. In *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines*, pages 13–21, April 2002.