Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 15 (2011) 1605 – 1609

Procedia Engineering

www.elsevier.com/locate/procedia

Advanced in Control Engineering and Information Science

A Dynamically Self-reconfigurable System Design Based on SSE Instruction Set

Kaiyu Wang[a]a*, Zhenan Tang[a], Yun Zhao[a], Hualong Li[a]

[a]*Dalian University of Technology, Gaoxinyuanqu Linggong Road 2, Dalian 116024,PR China*

Abstract

The Dynamic Reconfiguration Technology provides powerful technological support to achieve high-performance general-purpose CPU system in resolving the application of diversity issues, meanwhile improving the enhanced on-chip resource utilization, reducing the complexity of the design, cost and power consumption. The dissertation designs the integer part of the Intel SSE Instruction Set computing Reduced Instruction Set Computer CPU (RISC_CPU) and dynamically self-reconfigurable DISC_CPU, combining the Dynamic Reconfiguration Technology with the general-purpose CPU technology, and achieves Dynamic Instruction Set Computer CPU (DISC_CPU) supporting for multiple SSE (Streaming SIMD Extensions) Instruction Set on a single-chip FPGA.

Selection and/or peer-review under responsibility of [CEIS 2011]

Keywords:Dynamic Reconfiguration;Self-reconfigurable;Dynamic Instruction Set Computer CPU(DISC_CPU);SSE Instruction Set

1. Introduction

With the development of the semiconductor technology and the computer technology, based on Field Programmable Gate Array (FPGA), the advent of dynamic reconfiguration[1,2,3] of technology make it possible that the computer system structure turns to dynamic instruction set computers system. Its basic design idea is that, through dynamically configuring the bulk of the processing unit, storage unit and interconnection unit of chip[4], realizing the Instruction Level Parallelism(ILP), Data Level Parallelism(DLP) and Thread grade Level Parallelism(TLP), meeting the requirements of high performance in a wide range of application. It combines the flexibility of the general processor with the

* Kaiyu Wang. Tel.: +0086-411-84706003-3388;
E-mail address: wkaiyu@dlut.edu.cn.

doi:10.1016/j.proeng.2011.08.299

high performance and high efficiency of the specific processor while improving the hardware resource utilization.

2. Design of Reduced Instruction Set Computer CPU

Compared with the general CPU, there are two advantages of Reduced Instruction Set Computer CPU. In the instruction system, RISC_CPU improves the computing speed and makes the structure of computer simple and reasonable by simplifying the instruction system. In the means of the realization, sequential control signal of RISC_CPU is generated by the hardware routings and combinational logic, faster than the general CPU which read instruction one by one. RISC_CPU, realizing integer part of the Intel SSE Instruction Set, is composed of eight parts:(1)Clock generator (2)Instruction register (3)Arithmetic logic unit (4)Accumulator (5) Address multiplexer (6)Data controller (7)Program counter (8)State controller.

RISC_CPU adopts Harvard structure and storage the instructions and data in different storage. The structure of RISC_CPU is showed in Fig 1.

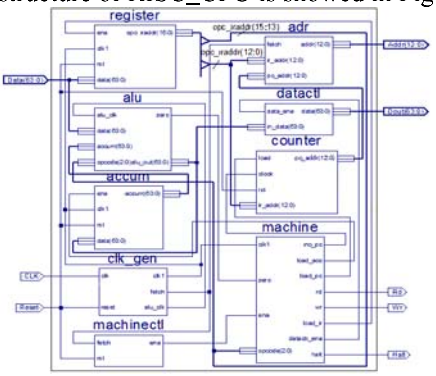

Fig 1.Structure of RISC_CPU

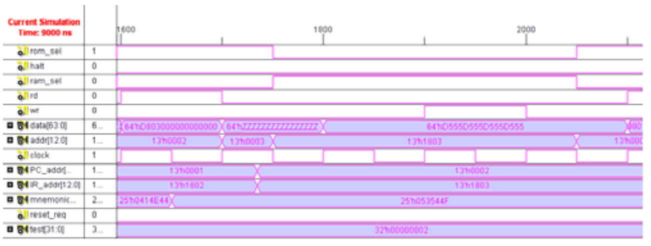

Fig 2. Simulation wave of SSE1

3. SSE1,SSE2 design and simulation

The dissertation designs SSE1 integer operation instructions: PAVGB, PAVGW, PMADDWD and SSE2 integer operation instructions: PADDB, PADDD and PADDQ. The function of instructions is showed in Table 1.There are five general data storage and pseudo-instruction: LDA (load data), STO (data storage), SKZ (zero jump), JMP (jump) and HLT (halt).

Table 1. Instruction of SSE

SSE	Instruction	Function
SSE1	PAVGB	DEST[i*8+7:i*8+0]<=(SRC[i*8+7:i*8+0]+DEST[i*8+7:i*8+0]+1)>>1; (i=0;i<8;i=i+1)
SSE1	PAVGW	DEST[i*16+15:i*16+0]<=(SRC[i*16+15:i*16+0]+DEST[i*16+15:i*16+0]+1)>>1; (i=0;i<4;i=i+1)
SSE1	PMADDWD	DEST[i*32+31:i*32+0]<=(DEST[i*32+15:i*32+0]*SRC[i*32+15:i*32+0] +DEST[i*32+31:i*32+16]*SRC[i*32+31:i*32+16]); (i=0;i<2;i=i+1)
SSE2	PADDB	DEST[i*8+7:i*8+0]<=DEST[i*8+7:i*8+0]+SRC[i*8+7:i*8+0]; (i=0;i<8;i=i+1)
SSE2	PADDD	DEST[i*32+31:i*32+0]<=DEST[i*32+31:i*32+0]+SRC[i*32+31:i*32+0]; (i=0;i<2;i=i+1)

SSE2	PADDQ	DEST[63:0]<=DEST[63:0]+SRC[63:0];

3.1. Simulation of SSE1 Instruction Set

RISC_CPU read the data from ROM or RAM according to the RD and ADDR signals and write the data to the RAM according to the WR and ADDR signals. The instruction and address of simulation is stored in ROM. It can execute different instructions by changing the machine code in the ROM. The initial data and simulation results are saved in the RAM. The simulation waveform is showed in Fig 2, and the data is showed in Table 2.

Table 2. Simulation result of SSE1

Time(ns)	PC	Instruction	Address	Data	Accumulator
950.0	0000	LDA	1801	a a a a a a a a a a a a a a a a	a a a a a a a a a a a a a a a a
1450.0	0001	PAVGW	1802	f f f f f f f f f f f f f f f f	d 5 5 5 d 5 5 5 d 5 5 5 d 5 5 5
1950.0	0002	STO	1803	d 5 5 5 d 5 5 5 d 5 5 5 d 5 5 5	d 5 5 5 d 5 5 5 d 5 5 5 d 5 5 5
2450.0	0003	PMADDWD	1801	a a a a a a a a a a a a a a a a	1 c 7 0 3 8 e 4 1 c 7 0 3 8 e 4
2950.0	0004	STO	1803	1 c 7 0 3 8 e 4 1 c 7 0 3 8 e 4	1 c 7 0 3 8 e 4 1 c 7 0 3 8 e 4
3450.0	0005	SKZ	0000	z z z z z z z z z z z z z z z z	1 c 7 0 3 8 e 4 1 c 7 0 3 8 e 4
3950.0	0006	HLT	0000	z z z z z z z z z z z z z z z z	1 c 7 0 3 8 e 4 1 c 7 0 3 8 e 4
4450.0	0007	PAVGB	1800	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	0 f 3 9 1 d 7 3 0 f 3 9 1 d 7 3
4950.0	0008	STO	1803	0 f 3 9 1 d 7 3 0 f 3 9 1 d 7 3	0 f 3 9 1 d 7 3 0 f 3 9 1 d 7 3
5450.0	0009	JMP	000b	z z z z z z z z z z z z z z z z	0 f 3 9 1 d 7 3 0 f 3 9 1 d 7 3
5950.0	000b	STO	1803	0 f 3 9 1 d 7 3 0 f 3 9 1 d 7 3	0 f 3 9 1 d 7 3 0 f 3 9 1 d 7 3
6450.0	000c	HLT	0000	z z z z z z z z z z z z z z z z	0 f 3 9 1 d 7 3 0 f 3 9 1 d 7 3

3.2. SSE2 Instruction Set simulation

Based on the SSE1 instruction set, SSE2 instruction set of RISC_CPU is realized by modifying the operation instructions of the ALU. The simulation waveform is showed in Fig 3, and the data is showed in Table 3.

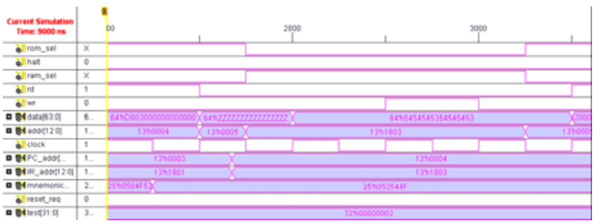

Fig3. Simulation wave of SSE2

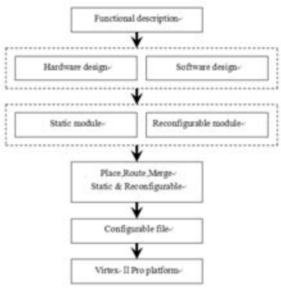

Fig 4. Design of a self-reconfigurable DISC system flow

Table 3. Simulation result of SSE2

Time(ns)	PC	Instruction	Address	Data	Accumulator
950.0	0000	LDA	1801	a a a a a a a a a a a a a a a a	a a a a a a a a a a a a a a a a
1450.0	0001	PADDD	1802	f f f f f f f f f f f f f f f f	a a a a a a a 9 a a a a a a a 9
1950.0	0002	STO	1803	a a a a a a a 9 a a a a a a a 9	a a a a a a a 9 a a a a a a a 9
2450.0	0003	PADDB	1801	a a a a a a a a a a a a a a a a	5 4 5 4 5 4 5 3 5 4 5 4 5 4 5 3
2950.0	0004	STO	1803	5 4 5 4 5 4 5 3 5 4 5 4 5 4 5 3	5 4 5 4 5 4 5 3 5 4 5 4 5 4 5 3
3450.0	0005	SKZ	0000	z z z z z z z z z z z z z z z z	5 4 5 4 5 4 5 3 5 4 5 4 5 4 5 3
3950.0	0006	HLT	0000	z z z z z z z z z z z z z z z z	5 4 5 4 5 4 5 3 5 4 5 4 5 4 5 3
4450.0	0007	PADDQ	1800	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	5 5 5 5 5 5 5 4 5 5 5 5 5 5 5 4
4950.0	0008	STO	1803	5 5 5 5 5 5 5 4 5 5 5 5 5 5 5 4	5 5 5 5 5 5 5 4 5 5 5 5 5 5 5 4
5450.0	0009	JMP	000b	z z z z z z z z z z z z z z z z	5 5 5 5 5 5 5 4 5 5 5 5 5 5 5 4
5950.0	000b	STO	1803	5 5 5 5 5 5 5 4 5 5 5 5 5 5 5 4	5 5 5 5 5 5 5 4 5 5 5 5 5 5 5 4
6450.0	000c	HLT	0000	z z z z z z z z z z z z z z z z	5 5 5 5 5 5 5 4 5 5 5 5 5 5 5 4

4. DISC_CPU self-reconfigurable design

The DISC_CPU supports multiple SSE Instruction Sets on a single-chip FPGA, realizes the ILP, DLP and TLP, and improves the generality of the chip. The development platform applied in this design is reconfigurable FPGA Virtex- Ⅱ Pro. The configurable signals are controlled by the program of PowerPC405.

4.1. The design flow of DISC_CPU

In this design, the ALU is designed as a reconfigurable module while the rest part of RISC_CPU and related peripheral are designed as a static module. Self-reconfigurable[5] design flow of DISC_CPU is showed in Fig 4.

4.2. Design result

After comparing placing and routing of two self-reconfigurable System, we discover that the placing and routing of the static components in the design don't change when DISC_CPU self-reconfigures, and changes only happen in the resources utilization and routing of the self-reconfigurable regions, realizing the change of arithmetic functions, achieving the desired effect of time division multiplexer in the same region, fulfilling the design of calculation functions unrelated with time. The system placing and routing of SSE1 and SSE2 are showed in Fig 5 and Fig 6. The reconfigurable module is in the rectangle in a dotted line. The erased reconfigurable system is showed in Fig 7.The resource utilization is showed in Table 4.

5. Conclusion

This paper design a self-reconfigurable system design of Dynamic Instruction Set Computer CPU, fulfilling the design and verification simulation of parts of SSE1 and SSE2 functioning in integer computing in the SSE instruction set, accomplishing the dynamic self-reconfiguration of different SSE instruction sets.

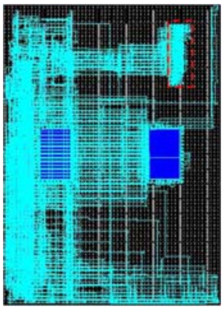

Fig 5. Placing and routing of SSE1

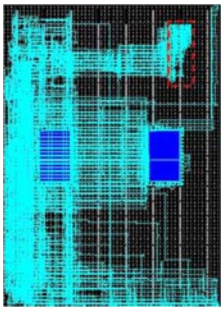

Fig 6. Placing and routing of SSE2

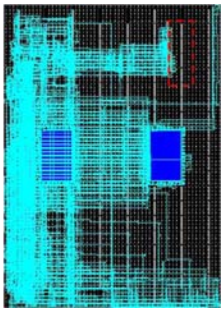

Fig 7. Wiping the region of PR

Table 4. List of resource utilization

module	resource utilization					
base	Number of Slices:	1497	out of	13696	10.9%
	Number of Slice Flip Flops:	1454	out of	27392	5.3%
	Number of 4 input LUTs:	1909	out of	27392	7.0%
	Number of BRAMs:	50	out of	136	36.8%
	Number of PPC405s:	1	out of	2	50%
sse1	Number of Slices:	121	out of	13696	0.9%
	Number of Slice Flip Flops:	32	out of	27392	0.1%
	Number of 4 input LUTs:	232	out of	27392	0.8%
sse2	Number of Slices:	119	out of	13696	0.9%
	Number of Slice Flip Flops:	32	out of	27392	0.1%
	Number of 4 input LUTs:	213	out of	27392	0.8%

The advantages of self-reconfigurable system design of dynamic instruction set computer CPU are showed in these aspects below: (1)Several RISC_CPU use TDM in the same self-reconfigurable region, improving the utilization of resources on chip; (2) Every RISC_CPU can be designed separately, not affecting each other, which makes the design less complex; (3)Unused RISC_CPU don't occupy device resources during the operation of system, decreasing the static power consumption of system. The information of configuration can be erased when there are no objects to be handled, which decreases the static power consumption further; (4)DISC_CPU based in self-reconfigure technology will meet the demands of shortening the design cycle of processor, accelerating the update, realizing high performance general processor chip.

References

[1] M. Handa, R. Vemuri. An efficient algorithm for finding empty space for online FPGA placement[C]. Design Automation Conference, 2004.

[2] H. Walder, M. Platzner. Non-preemptive multitasking on FPGAs: Task placement and foot-print transform[C]. International Conference on Engineering of Reconfigur- able Systems and Architectures, 2002.

[3] K. Bazargan, R. Kastner, M. Sarrafzadeh. Fast template placement for reconfigu- rable computing systems[J]. IEEE Design and Test, Special Issue on Reconfigurable Computing, 2000, 17(1):293-297.

[4] K. Compton, S. Hauck. Reconfigurable Computing: A Survey of Systems and Software [J]. ACM Computing Surveys, 2002,34(2):191–210.

[5] R. W. Taylor. A self-reconfiguring processor[R]. IEEE Symposium on Field Programm- able Custom Computing Machines, 50-59, 1993.