1H-08

Frix: Feasible and Reconfigurable IBM PC Compatible SoC

Yuki Matsuda, Eri Ogawa, Tomohiro Misono, Ryohei Kobayashi and Kenji Kise

Graduate School of Information Science and Engineering, Tokyo Institute of Technology

1 Introduction

In order to develop high performance computer system, the evaluation environment for architectural ideas is desired. The evaluation is more reliable when their ideas are evaluated on the environment running a general-purpose OS.

Full system software simulators are widely used for this purpose. Software simulators are very flexible and developers are easy to use them, but the full system simulation with software takes too much time. FPGA (Field Programmable Gate Array) is reconfigurable hardware and users can implement their own circuits. FPGA-based simulators can run much faster than the software-based ones. Thus, full system FPGA-based simulators are desired.

We have proposed a feasible and reconfigurable IBM PC Compatible SoC, named Frix [1]. Frix is an FPGA-based computer system with an x86 soft core processor which is compatible with Intel 80486SX. Frix can boot FreeDOS 1.1 and TinyCore 5.3 (Linux kernel 4.3). The corresponding FPGA board is Terasic's Altera DE2-115 FPGA board and Digilent's Nexys4 (DDR) FPGA board.

The source code of Frix is written in Verilog HDL, and we have released it as open-source. Frix can downloaded from `http://www.arch.cs.titech.ac.jp/a/Frix`. Researchers can add their proposed architecture to Frix and evaluate its performance. On the other hand, learners can understand how computer systems run, by reading the source code of Frix.

In this paper, we firstly show the design of Frix and finally evaluate the efficiency of Frix as evaluation environments.

2 Frix

Figure 1 shows the design overview of Frix. Frix uses a VGA monitor, a SD card (storage) and a PS/2 keyboard as inout. The interconnection of Frix is written in Verilog HDL, and the processor is an x86 processor. In this section, we firstly explain the baseline of our research, called ao486 SoC[2]. Finally, we explain the design of Frix.

2.1 ao486 SoC

Upon developing this system, we used the open-source SoC called ao486 SoC[2] as a baseline. Ao486 SoC has an soft core processor which is compatible with Intel 80486SX and is written in Verilog HDL. The corresponding board of ao486 SoC is Terasic's Altera DE2-115 FPGA board. According to the author's document, this system can boot Windows95 and Linux Kernel 3.13. However, the ao486 SoC uses soft core processors called Nios II and a bus system called Avalon Interconnect, which are Altera's IP cores.

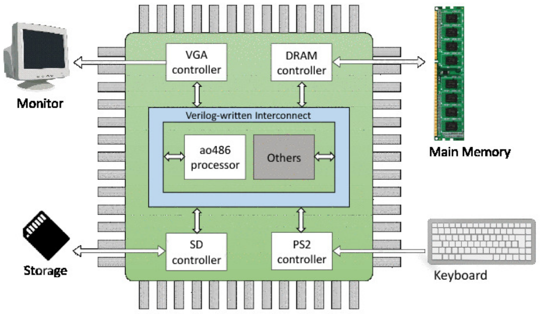

Figure 1: The design overview of Frix.

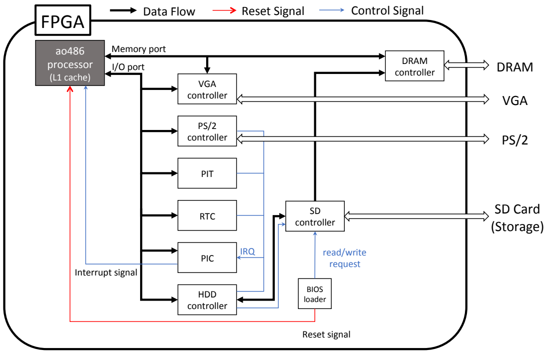

Figure 2: The block diagrams of Frix.

2.2 Design of Frix

Figure 2 shows the block diagrams of Frix. We use several ao486 modules for this system: ao486 processor, some controllers (VGA, PS/2, SD, HDD), PIT (Programmable Interval Timer), RTC (Real Time Clock) and PIC (Programmable Interrupt Controller). We newly added BIOS loader module and bus module, both of which are written in Verilog HDL, and Frix can run on two major vendors' FPGA.

Figure 3 shows the snapshot of DOOM running on Nexys4 DDR FPGA board. DOOM is running on FreeDOS 1.1. Frix can also boot TinyCore 5.3 (Linux kernel 4.3). Researchers can evaluate their proposed architecture on Frix with general purpose benchmarks, like SPEC CPU, on TinyCore.

For more information about Frix, please refer to our MCSoC paper [1].

Table 1: Execution time and miss rate

exec time (sec)	gzip	gcc	mcf	crafty	parser	perlbmk	gap	vortex	bzip2	twolf	gmean
baseline	362.5	224.5	26.8	670.1	439.4	244.3	130.6	1172.1	907.4	330.0	306.0
AMI	358.5	224.0	28.4	667.4	438.6	241.6	129.5	1170.2	906.1	330.0	306.5
relative	0.989	0.998	1.059	0.996	0.998	0.989	0.992	0.998	0.999	1.000	1.002
miss rate	gzip	gcc	mcf	crafty	parser	perlbmk	gap	vortex	bzip2	twolf	gmean
baseline	3.72%	3.34%	18.95%	1.73%	6.41%	1.24%	3.04%	1.50%	1.63%	0.50%	2.56%
AMI	3.68%	3.35%	21.46%	1.62%	6.45%	1.30%	3.05%	1.47%	1.61%	0.53%	2.59%
relative	0.991	1.002	1.132	0.934	1.007	1.049	1.005	0.980	0.989	1.054	1.013

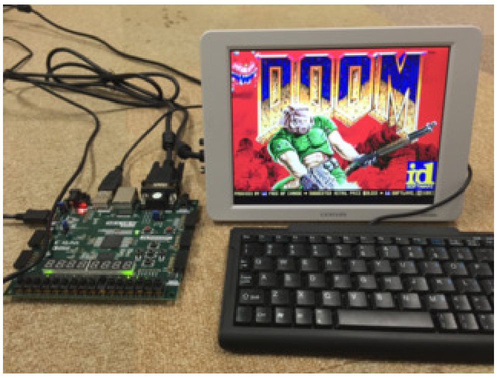

Figure 3: The snapshot of DOOM running on Nexys4 DDR FPGA board. DOOM is the first FPS game and very famous one.

Table 2: Hardware Resource

	Logic Cells		Memory (Kbits)	
	dcache	total	dcache	total
baseline	4202	54568	154.6	2585.9
AMI	4751	55162	154.6	2585.9
relative	1.131	1.011	1.000	1.000

3 Evaluation

3.1 Setup

In order to show the efficiency of Frix as evaluation environments, we redesign data cache on ao486 processor, and evaluate the system performance. The data cache is 16KB 4-way set associative cache. The line size of the cache is 16B, and the number of cache lines per way is 256.

We implement a cache with 255 cache lines on each way, with a novel method called Arbitrary Modulus Indexing (AMI)[3]. Non-power-of-2 cache lines can reduce cache line conflicts, and thus improve system performance.

In our evaluation we use DE2-115 FPGA board, which has Cyclone IV EP4CE115 FPGA. We run subset of SPEC CPU 2000 benchmark on TinyCore 5.3, and evaluate the system performance.

3.2 Result

Table 1 shows execution time and miss rate while the run of each application. The "relative" means division of the result with AMI by the one on baseline. On the result of geometric mean, AMI increases execution time by 1.002 times and miss rate by 1.013 times. This result shows AMI does not always improve the system performance.

Table 2 shows the hardware resource on each configuration. AMI increases occupied logic cells by 1.131 times in the scale of dcache, and by only 1.011 times in the scale of overall system. Hence the overhead of AMI is enough small. In this evaluation, occupied memory is not affected by AMI.

Cyclone IV EP4CE115 FPGA has 114,480 logic cells and 3.888 Kbits embedded memory. In the result of baseline, the usage of logic cells are 54568/114480 = 47.7%. Researchers can use more than the half of logic cells to implement their proposed architecture. The usage of embedded memory are 2585.9/3888 = 66.5%, and thus researchers also use about the 1/3 of total embedded memory.

For every application of the SPEC CPU 2000 benchmark, we confirm that **the output results of Frix exactly match the golden output provided by the benchmark**.

These results mentioned above shows researchers can evaluate their proposed architecture with Frix.

4 Conclusion

We proposed a reconfigurable IBM PC Compatible SoC, named Frix. We released Frix as open-source, and it can be downloaded from `http://www.arch.cs.titech.ac.jp/a/Frix`. Our evaluation showed the efficiency of Frix on computer architecture research.

Acknowledgement

This work is supported in part by JSPS KAKENHI Grant Number 25330056.

References

[1] E. Ogawa, Y. Matsuda, T. Misono, R. Kobayashi, and K. Kise, "Reconfigurable IBM PC Compatible SoC for Computer Architecture Education and Research," in *Proceedings of 2015 IEEE 9th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC)*, Sept 2015, pp. 65–72.

[2] "ao486," https://github.com/alfikpl/ao486.

[3] J. Diamond, D. Fussell, and S. Keckler, "Arbitrary Modulus Indexing," in *Proceedings of 2014 47th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Dec 2014, pp. 140–152.