

A Low Power Asynchronous Java Processor for Contactless Smart Card

Chun-Pong YU, Chiu-Sing CHOY, *Hao MIN, Cheong-Fat CHAN, Kong-Pang PUN

Department of Electronic Engineering, CUHK, Hong Kong

*ASIC & System State Key Laboratory, Fudan University, P. R. China

Abstract - This paper presents the design of a low power 16-bit asynchronous java processor for contactless smart card. It can directly execute the java bytecodes in a subset of the instruction set defined in the Java Card Virtual Machine specification. [1] The remaining java bytecodes are handled by software routines. Also, we intend to use asynchronous circuit design technique to reduce the power consumption of the java processor core. It has been fabricated in a CMOS 0.35-um technology and the experimental result shows that it is suitable for the contactless smart card.

I. Introduction

Java Card technology enables programs written in the Java programming language to run in smart cards. From software perspective, Java virtual machine defines a stack-based processor for implementing java bytecodes. Software JVM can emulate its architecture. Also, it is possible to design a microprocessor that can execute these bytecodes directly in order to increase the operating performance [2]. Since there are only a few proposed designs of java processor and even less in the contactless smart card context, the processor presented here is novel. Our proposed design is a 16-bit asynchronous java processor that can directly execute some useful instructions defined in [1] with 16-bit operands including register read/write operations, arithmetic operations, logic operations and branch operations. The rest of them including method invocation, method return, array operation and object manipulation are handled by software routines. The hardware features are shown in table 1.

TABLE 1
Hardware features

●	16-bit ALU
●	16X16-bit RAM
●	10 level 16-bit Stack
●	16-bit address bus
●	Two 16-bit I/O ports

II. The Architecture of the Java Processor

Asynchronous circuit design technique is used in order to reduce the power consumption. In this design, we use the four-phase handshaking protocol (Four-phase bundled data interface protocol) since simpler control logic can be realized for this protocol compared with the two-phase handshaking protocol. In the system of contactless smart card, low power consumption is preferred over high performance. So, the normally opaque latch controller shown in fig. 1 is chosen for our design. [3]

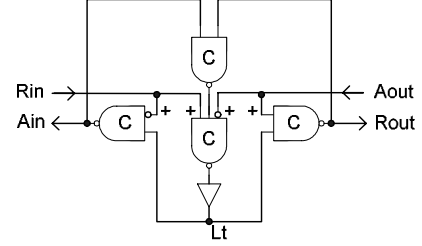


Fig. 1 The normally opaque latch controller [3]

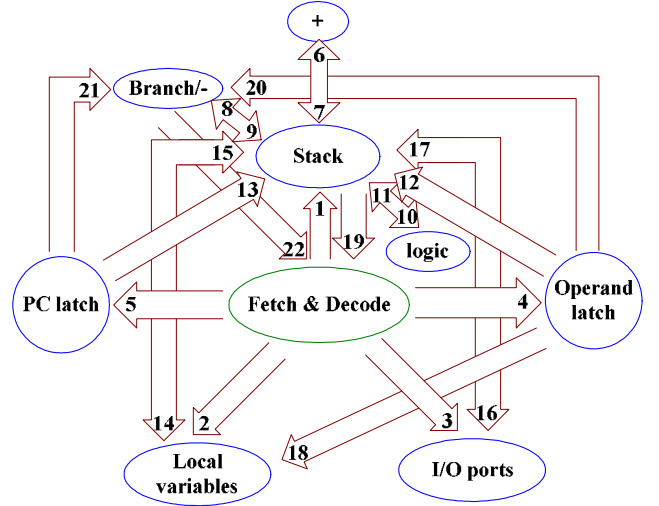


Fig. 2 Paths for transmitting control bits and data bits (path 1, path 2 and path 3 are the paths for transmitting control bits)

TABLE 2
The subset of the instruction set

Instruction	Paths involved
sipush const	1, 4, 12
iload index	1, 2, 4, 18, 15
istore index	1, 2, 4, 18, 14
iadd	1, 6, 7
isub	1, 8, 9
ishl	1, 10, 11
ishr	1, 10, 11
iand	1, 10, 11
ior	1, 10, 11
ixor	1, 10, 11
if_scmpeq_w offset	1, 4, 5, 8, 20, 21, 22
if_scmpne_w offset	1, 4, 5, 8, 20, 21, 22
if_scmplt_w offset	1, 4, 5, 8, 20, 21, 22
if_scmpge_w offset	1, 4, 5, 8, 20, 21, 22

goto offset	None
jsr offset.	1, 5, 13
ret	1, 19
nop	None
stack2port	1, 3, 16
port2stack	1, 3, 17
sleep	None

Paths for transmitting control bits and data bits among the functional blocks are shown in fig. 2. In the asynchronous java processor, latches are used instead of flip-flops. Since the asynchronous circuit design technique is used, all the functional blocks including the stack, the block of local variables, the I/O ports, the PC latch, the operand latch, the addition block, the logic operation block and the branch operation block shown in this figure operate only when necessary.

Techniques are adopted to decrease power consumption by reducing circuit activities. One example is in stack access. The java processor is a stack-based processor. So, its performance is limited by the frequent stack access. We use result-forwarding technique in order to increase the performance of the processor. So, the data written to the stack is forwarded when it is required in the next instruction normally through reading from the top of stack and this saves one write-stack operation.

The 16X16bit RAM of the java processor realizes the block of local variables shown in fig. 2. The address of the RAM (variable index) is provided by the operand latch shown in this figure.

The PC latch and the operand latch shown in fig. 2 are used to store the value of the program counter of the current instruction and the 16-bit operand read from the instruction memory in the fetch/decode unit respectively.

The subset of the instruction set is shown in table 2. The instructions stack2port and port2stack that are not defined in [1] are added into the instruction set of the processor for manipulating the I/O ports. Also, the instruction sleep is added to the instruction set to support the sleep mode. Besides, there is no data transfer when executing goto, nop and sleep since these instructions are only executed in the fetch/decode unit.

III. Implementation and experimental result

The processor design is implemented using 0.35um CMOS process and the chip microphotograph of the java processor is shown in fig. 3. The size of the processor core is 1.2mm².

TABLE 3
Experimental result

Supply(V)	MIPS	Power (mW)	MIPS/W
3.3	18.7	14.1	1326
3	17.2	10.8	1593
2.5	14.2	6.3	2254
2	10.2	2.5	4080

The bytecodes of a test program are stored in the ROM of the java processor. Register read/write operations, arithmetic operations and branch operations are involved in the

program. The experimental result shown in Table 3 is done by executing the bytecodes in the test program.

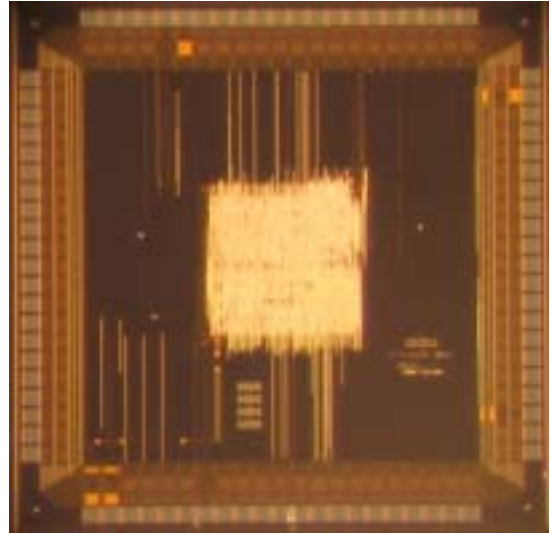


Fig. 3 Chip microphotograph

For a contactless smart card system, the external clock is 13.56MHz [4] and the power consumption has to be less than 30mW. [5] The experimental result shows that the power consumption of the processor is 14.1mW when the speed is 18.7 MIPS (At 3.3V). Obviously, this can fulfil the requirement of the system of contactless smart card. We can reduce the power consumption by decreasing the supply voltage. Also, its power consumption is about equal to zero in the sleep mode.

IV. Conclusion

A low power asynchronous java processor for contactless smart card is designed and it can directly execute the java bytecodes in a subset of the instruction set defined in the Java Card Virtual Machine specification. [1]

V. Acknowledgement

The work presented in this paper is supported by a HKSAR government RGC grant 4176/01E.

Reference

- [1] Sun Microsystems, "Java Card 2.2 Virtual Machine Specification", <http://java.sun.com/products/javacard>, 2002
- [2] Sun Microsystems, "Java Processors - the Coming of Age", <http://industry.java.sun.com/javaneers/stories/story2/0,1072,36281,00.html>, 2001
- [3] K. T. Christensen, P. Jensen, P. Korger, and J. Sparsø, "The design of an asynchronous TinyRISC TR4101 microprocessor core", In Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems, Pages 108-119. IEEE Computer Society Press, 1998
- [4] Abrial, A., Bouvier, J., Renaudin, M., Senn, P., Vivet, P., "A new contactless smart card IC using an on-chip antenna and an asynchronous microcontroller", Solid-State Circuits, IEEE Journal of , Volume: 36 Issue: 7 , Jul 2001, Page(s): 1101 - 1107
- [5] Cormie C., Grimonprez G., "A RISC microprocessor for contactless smart cards", EUROMICRO 97. 'New Frontiers of Information Technology', Proceedings of the 23rd EUROMICRO Conference , 1-4 Sep 1997 Page(s): 658 -663