RISC versus CISC

Article in IEEE Potentials · September 1995

DOI: 10.1109/45.464688 · Source: IEEE Xplore

Tariq Jamil
Sultan Qaboos University

RISC versus CISC

Why less is more

Tariq Jamil

Rapid advances in the integrated-circuits technology are challenging today's computer architects. Efforts are underway by Intel to devise innovative techniques for computations in the arena of Complex Instruction Set Computer (CISC) through Pentium and P6 microprocessors. Three giants of the computer industry, namely IBM, Apple, and Motorola, have joined hands to explore Reduced Instruction Set Computer (RISC) through PowerPC chips.

The developments in the microelectronics technology are expected to yield 50 to 100 million transistors on a single chip by the year 2000. The challenge is to design faster and more intelligent computers which can exploit such on-chip capability to the utmost limit by incorporating efficient techniques to handle computations.

The microprocessor families Intel x86 and Motorola M68000 are characterized by their abundant instruction sets, multiple addressing modes, and multiple instruction foımats and sizes. Their control is microprogrammed, and different instructions execute within a different number of cycles. The control units of such microprocessors are complex because they have to distinguish between a large number of opcodes, addressing modes, and formats. This type of system belongs to the category called Complex Instruction Set Computer (CISC).

	RISC	***CISC***
Registers	16-32 general purpose	8-16 general purpose
Datatypes	Usually two (integer and floating point)	Usually more than two
Instructions	Load/store general registers, operations on datatypes in registers	Correspond to datatypes

Instruction formats	Fixed length, two main types;
	load/store;
	R:= R op R.
		Variable length, many types;
		load/store;
		R:=R op R;
		R:=Mem op R;
		M:=Mem op Mem.

Encoding	1 instruction=1 operand or 1 operation	1 instruction=1 statement
Design objective	Trade off program length, minimize time to execute instruction	Minimum program length, Maximum work/instruction
Implémentation	Hard-wired processor and software; fast processor and fast cache for instructions; instructions take one clock cycle; simple pipeline	Microprogrammed processor; slow primary memory and fast clock; instructions take variable time; pipeline is complex; larger implementation may result in longer design time
Caching	Essential for instructions	Useful
Compiler design	Should stress best ordering	Should stress finding right instructions
Philosophy	Move all functions to software	Move any useful software function into hardware, including diagnostics.

Table 1 Comparison between RISC and CISC

The direct opposite of the traditional CISC design, there emerged a new trend of computer design in the early eighties, called RISC—Reduced Instruction Set Computer. This was an attempt by the computer designers to meet the demand for efficient computing by "reducing" the number of instructions, addressing modes, and formats. In an ideal RISC, all instructions have the same size, and execute within a single clock cycle. This reduces the complexity in the control unit which is usually hard-wired. More room on the chip's real estate means more features can be embedded to enhance performance.

Why RISC?

Several studies tracing the programs' behavior have revealed that 25% of the instructions in the instructions' set make up 95% of the execution time. This means that about 75% of the hardware-supported instructions often are not used at all.

A study conducted by Patterson and Ditzel concluded that for a particular IBM 360 compiler, 10 instructions accounted for 80% of all the instructions executed. 16 instructions accounted for 90%, 21 instructions accounted for 95%, and 30 instructions accounted for 99% of all the instructions executed.

Thus, a great majority of CISC instructions are never used. If these "extraneous" instructions are eliminated from the instructions' set, the control unit, which is responsible for decoding instructions, will be less complex. This will reduce the chip area dedicated for this unit, thus allowing embedding of some other additional performance enhancement features on the chip, such as cache and memory management unit.

By reducing the control area on

0278-6648/95/$4.00 © 1995 IEEE

the VLSI chip, the regularization factor—defined as the total number of devices on the chip, excluding ROMs, divided by the number of drawn devices (such as registers, ALUs, counters, etc.)—is increased. In general, the higher the regularization factor, the lower the VLSI design cost. Thus RISC also allows a more cost effective chip design.

A simpler and smaller control unit in a RISC has fewer gates. This means shorter propagation paths (fewer gates to go through) for the control unit signals. The result is a faster operation. Furthermore, owing to the simplicity in the design of RISC control unit, it has a higher degree of reliability.

Year	*CISC*	*RISC*
1971	Intel 4004	
1972	Intel 8008	
1973		
1974	Intel 8080, Motorola 6800	
1975		IBM 801 started
1976	Intel 8085	
1977	Motorola 6809	
1978	Intel 8086	
1979	Intel 8088, Motorola 68000	
1980	Intel 80186, Motorola 68008	UCB RISC announced
1981	Intel 80188	Stanford MIPS
1982	Intel 80286	IBM 801 announced
1983	Motorola 68010	Transputer
1984	Motorola 68020	
1985	Intel 80386	
1986		MIPS R2000, IBM, ROMP, HP-PA
1987	Motorola 68030	Am29000, SPARC
1988	Intel 80376	M88000, R3000
1989	Intel 80486, Motorola 68040	i860, R6000
1990		Am29050, RS/6000
1991		R4000, MC88110
1992		Alpha, R4400, SuperSPARC
1993	Intel Pentium	PowerPC 601
1994	Motorola 68060	
1995	Intel P6	PowerPC 604

Table 2 Evolution of CISC and RISC microprocessors

What is RISC?

RISC is defined as a computer with most of these characteristics:

- Instructions are conceptually simple.
- Instructions are of uniform length.
- Instructions use one (or very few) instruction formats.
- The instructions' set is orthogonal, i.e., there is little overlapping of instruction functionality.
- Instructions use one (or very few) addressing modes.
- The architecture is a load-and-store architecture, i.e., only LOAD and STORE instructions access memory; all operate instructions are register-to-register instructions.
- The instructions set architecture supports two (or a few more) datatypes (typically integer and floating point).
- Almost all instructions execute in one clock cycle.
- The architecture has a large number of general-purpose registers.
- The architecture has a hard-wired control unit (may be changed to microprogrammed if the application demands).
- An effort is made to support high-level language operations inherently in the machine design by a judicious choice of instructions and by using optimized compilers.

The above points should be considered as a "flexible framework" for a definition of a RISC machine, or rather, a list of design attributes, practiced in most RISC systems.

RISC shortcomings

Ironically, RISC shortcomings are the byproducts of some of its advantages. The principle RISC disadvantage is its reduced number of instructions. Since certain functions which require only one instruction on a CISC machine require two, three, or more RISC instructions, the resulting program code will be longer. More memory will have to be allocated to RISC programs. The instruction traffic between the memory and the CPU will be increased.

Studies show the average RISC program is about 30% longer than a CISC program for the same function. Although the combined execution time of the multiple RISC instructions may be shorter than the single CISC instruction, the question of program size is nonetheless a valid one. Larger programs not only require more memory for storage but are also more susceptible to page faults in virtual memory systems and decrease in the hit ratio for instruction caches. (The apparent amount of main memory is increased by storing "pages" in secondary storage and swapping them in and out as needed.)

Since RISC uses a large number of registers, a complicated register address decoding is involved. There is no floating point operation support in RISC, yet these operations are indispensable in mathematics, science and engineering, therefore any commercial RISC system must include support for a floating-point unit (FPU).

Evolution of CISC microprocessors

Intel started with a four-bit microprocessor 4004 in 1971 which was primarily used in calculators. This was followed by an eight-bit 8008 in 1972, 8080 in 1974, and 8085 in 1976. Motorola entered the microprocessor market with its eight-bit 6800 family in 1974, culminating with its top eight-bit product 6809 in 1977.

The 16-bit computing started with the introduction of 8086 by Intel in 1978, which was soon followed by 8088 in 1979. During the same year, Motorola announced the dawn of its 16-bit microprocessor family, the 68000. During the subsequent six years, Intel marketed successors to its 16-bit microprocessor family, the 80186, the 80188, and 80286, while Motorola shipped out the 68010.

Motorola entered the 32-bit computing arena with its 68020 in 1984, and soon after that, Intel launched its first full-fledged 32-bit microprocessor family, the 80386 in 1985, the 80376 in 1988, and the 80486 in 1989. Keeping competitive with Intel, Motorola produced its version of 32-bit processors, the 68030 in 1987, and the 68040 in 1989.

Intel introduced Pentium in 1993. It has the capability of fetching, decoding, and executing two instructions in parallel, has a double 64-bit databus in and out of chip, and both instruction and data caches are on-chip. Motorola announced the 68060 processor in 1994, which is also two-issue superscalar and has both instruction and data caches on-chip.

The next microprocessor in the Intel x86 family, currently referred to as P6, was announced quite recently. It is expected to double the performance of the Pentium. On the other side, Motorola has joined hands with IBM and Apple to produce a family of RISC processors, called PowerPCs.

Evolution of RISC microprocessors

As can be seen in Table 2, RISC microprocessors have been steadily developing in parallel with the Intel and Motorola families. In the mid-1970s, IBM started work on its 801, but announced it only after the pronouncement of the University of California, Berkeley project RISC in 1980. Stanford University came out with the MIPS (Microprocessor without Interlocked Pipeline Stages) in 1981. Subsequently, MIPS Computer Systems Co. was founded, which designed commercial versions of the Stanford MIPS, the Rx000 (x = 2,3,4,6).

Incorporating some features of the basic Berkeley RISC, Sun Microsystems announced SPARC (Scalable Processors ARChitecture) in 1987, and the latest top-level SPARC is the SuperSPARC, produced by Sun, in cooperation with Texas Instruments, in 1992. IBM continued its interest in RISC with ROMP (Research Office products division MicroProcessor) in 1986, and RS/6000 in 1990. Motorola started its RISC family with M88000 in 1988, and continued until 1991 with MC88110.

In 1993, three giants of the microprocessor technology, namely IBM, Motorola, and Apple joined efforts to develop the PowerPC family of RISC processors, starting with PowerPC 601 in 1993. Recently, the PowerPC 604 has been announced by these companies. Other RISC products are the INMOS Transputer, Hewlett-Packard Precision Architecture (HP-PA), Advanced Micro Devices Am29000 family, and the Digital Equipment Corporation (DEC) Alpha AXP family.

Today's stars

Intel Pentium (CISC). The Pentium is a 3.1 million transistor, 0.8 micron, BiCMOS technology, three-layer metal, 273-pin grid array package microprocessor. It is a 32-bit system with a double 64-bit databus inside and outside of the chip. There are two separate 8-Kbytes caches on-chip: one for code and one for data. Also, each cache has a separate address translation look-aside buffer (TLB) associated with it. It is this availability of dual cache and dual TLBs that permits the CPU to handle simultaneous instruction and data operand access. Since the Pentium is a two-issue superscalar, two instructions are fetched and decoded simultaneously.

Intel P6 (CISC). The P6 is a 0.6 micron, four-layer metal, BiCMOS technology microprocessor operating at 2.9 V and 133 MHz. It has 5.5 million transistors in the CPU core and a three-issue superscalar microarchitecture. There are a total of five parallel execution units (two integer, one load, one store, and one FPU) which permit out-of-order execution, dynamic branch prediction, and speculative execution. As in Pentium, 8-Kbytes of instruction cache and 8-Kbytes of data cache are available on-chip.

Motorola MC68060 (CISC). The MC68060 is a 2.5 million transistor, 0.5 micron CMOS, three-layer metal, 223-pin pin-grid-array microprocessor. It is a two-issue superscalar system with an instruction cache of 8-Kbytes and a data-cache of 8-Kbytes on-chip.

PowerPC 601 (RISC). The PowerPC 601 is a 66 MHz, 2.8 million transistor, 0.6 micron, four-layer metal CMOS, packaged in 304 pin ceramic quad flat pack microprocessor. It is a three-issue superscalar system and has three independent execution units (integer unit, floating point unit, and branch processing unit). Three simultaneously issued instructions can execute simultaneously only if they are of the appropriate type to be issued to the three independent execution units. That is, if one instruction is integer, one floating-point, and one branch. If two back-to-back integer instructions are encountered, they will be executed in sequence through the integer unit, although branch and floating-point instructions may be dispatched from behind. The PowerPC 601 has a 32-Kbytes unified (code and data in the same cache), eight-way set-associative, 64 bytes per line cache.

PowerPC 604 (RISC). The 604 is a 0.5 micron, four metal layers, 3.6 million transistor processor operating at 100 MHz and packaged as 304-pin ceramic quad flat pack. It has a four-issue superscalar architecture, with six execution units, and 16-Kbytes each of instruction-cache and data cache.

The future

It is difficult nowadays to differentiate between RISC and CISC because most RISC features are being incorporated into CISC. The results are hybrid RISC/CISC architectures. For example, Intel's P6 breaks the lengthy CISC instructions into simpler operations that more closely resemble RISC instructions. The simplified operations—Intel calls them *micro-ops*—are then fed into a core that takes advantage of the latest RISC innovations. These micro-ops are easier to dispatch and execute in parallel than their complex x86 antecedents. The same basic mechanism is being used in the latest RISC processors also, including the PowerPC 604.

Last year, Intel formed an alliance with Hewlett-Packard to design a new microprocessor that is expected to

Year	*CPU*	*Technology*	*Layers*	*No. of Instr. issued/cycle*	*Transistor count (x10⁶)*	*Cache sizes (Kbytes)*
1995	PC604	0.5-micron CMOS	4	4	3.6	16 I-cache 16 D-cache
1995	P6	0.6-micron BiCMOS	4	3	5.5	8I-cache 8D-cache
1994	M68060	0.5-micron CMOS	3	2	2.5	8I-cache 8D-cache
1993	PC601	0.6-micron CMOS	4	3	2.8	32 Unified Cache
1993	Pentium	0.8-micron BiCMOS	3	2	3.1	8I-cache 8D-cache

Table 3 Summary of design characteristics of latest microprocessors

appear in 1997 or 1998. The two companies are revealing little about this microprocessor except that it will attempt to leapfrog RISC technology and run all existing software for Intel's x86 and HP's PA-RISC chips.

A new technology called Very Long Instruction Word (VLIW), in which several simple RISC-like operations are packed into very long CISC-like instructions, is also under investigation. Each instruction packet contains operations that are not interdependent, so the CPU can execute them in parallel. It is rumored that the Intel's next chip, the P7, will probably have some form of this new technology embedded in it.

The final verdict on the RISC/CISC controversy is still out. However, it is evident that some form of RISC will always be part of the so-called CISC microprocessors.

Read more about it

The following are excellent sources of information on both RISC and CISC architectures:

- D. Tabak, *Advanced Microprocessors* (Second Edition), McGraw-Hill, 1995.
- K. Hwang, *Advanced Computer Architecture*, McGraw-Hill, 1993.
- R.J. Baron and L. Higbie, *Computer Architecture*, Addison-Wesley, 1992.
- R.J. Baron and L. Higbie, *Computer Architecture Case Studies*, Addison-Wesley, 1992.
- J.L. Hennessy and D.A. Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann Publishers Inc., 1990.
- *Byte*, particularly April 1995 issue.
- *IEEE Micro*, particularly April 1995 issue.

On the World Wide Web—

- This site contains information about all microprocessors known to mankind since the birth of computer engineering: http://infopad.eecs.berkeley.edu/CIC/archive/cpu_history.html
- For those interested in reading about computer architecture in general: http://www.cs.wisc.edu/~arch/www

Acknowledgements

I am deeply indebted in gratitude to my advisor, Dr. R.G. Deshmukh, for his able guidance and encouragement which are still continuing during my doctoral studies at the Florida Institute of Technology. It is to his excellent method of teaching and mentoring that I owe my kindling of interest in the area of computer architecture and parallel processing.

About the author

Tariq Jamil, Student Editor of *IEEE Potentials*, is currently a Ph.D. candidate in the computer engineering program at the Florida Institute of Technology, Melbourne, Florida. He received his B.Sc. (Hons) degree in electrical engineering from the NWFP University of Engineering & Technology, Peshawar, Pakistan in 1989, and earned his M.S. in computer engineering from the Florida Institute of Technology in 1992. His research interests are in computer architecture, parallel processing and dataflow computing. He can be contacted via email at address jamil@ee.fit.edu.