Processor design

From Wikipedia, the free encyclopedia

Processor design is the design engineering task of creating a microprocessor, a component of computer hardware. It is a subfield of electronics engineering and computer engineering. The design process involves choosing an instruction set and a certain execution paradigm (e.g. VLIW or RISC) and results in a microarchitecture described in e.g. VHDL or Verilog. This description is then manufactured employing some of the various semiconductor device fabrication processes. This results in a die which is bonded onto some chip carrier. This chip carrier is then soldered onto some printed circuit board (PCB).

The mode of operation of any microprocessor is the execution of lists of instructions. Instructions typically include those to compute or manipulate data values using registers, change or retrieve values in read/write memory, perform relational tests between data values and to control program flow.

Contents

Details

CPU design focuses on six main areas:

1. datapaths (such as ALUs and pipelines)
2. control unit: logic which controls the datapaths
3. Memory components such as register files, caches
4. Clock circuitry such as clock drivers, PLLs, clock distribution networks
5. Pad transceiver circuitry
6. Logic gate cell library which is used to implement the logic

CPUs designed for high-performance markets might require custom designs for each of these items to achieve frequency, power-dissipation, and chip-area goals whereas CPUs designed for lower performance markets might lessen the implementation burden by acquiring some of these items by purchasing them as

intellectual property. Control logic implementation techniques (logic synthesis using CAD tools) can be used to implement datapaths, register files, and clocks. Common logic styles used in CPU design include unstructured random logic, finite-state machines, microprogramming (common from 1965 to 1985), and Programmable logic arrays (common in the 1980s, no longer common).

Device types used to implement the logic include:

- Transistor-transistor logic Small Scale Integration logic chips - no longer used for CPUs
- Programmable Array Logic and Programmable logic devices - no longer used for CPUs
- Emitter-coupled logic (ECL) gate arrays - no longer common
- CMOS gate arrays - no longer used for CPUs
- CMOS mass-produced ICs - the vast majority of CPUs by volume
- CMOS ASICs - only for a minority of special applications due to expense
- Field-programmable gate arrays (FPGA) - common for soft microprocessors, and more or less required for reconfigurable computing

A CPU design project generally has these major tasks:

- Programmer-visible instruction set architecture, which can be implemented by a variety of microarchitectures
- Architectural study and performance modeling in ANSI C/C++ or SystemC
- High-level synthesis (HLS) or register transfer level (RTL, e.g. logic) implementation
- RTL verification
- Circuit design of speed critical components (caches, registers, ALUs)
- Logic synthesis or logic-gate-level design
- Timing analysis to confirm that all logic and circuits will run at the specified operating frequency
- Physical design including floorplanning, place and route of logic gates
- Checking that RTL, gate-level, transistor-level and physical-level representations are equivalent
- Checks for signal integrity, chip manufacturability

Re-designing a CPU core to a smaller die-area helps to shrink everything (a "photomask shrink"), resulting in the same number of transistors on a smaller die. It improves performance (smaller transistors switch faster), reduces power (smaller wires have less parasitic capacitance) and reduces cost (more CPUs fit on the same wafer of silicon). Releasing a CPU on the same size die, but with a smaller CPU core, keeps the cost about the same but allows higher levels of integration within one very-large-scale integration chip (additional cache, multiple CPUs, or other components), improving performance and reducing overall system cost.

As with most complex electronic designs, the logic verification effort (proving that the design does not have bugs) now dominates the project schedule of a CPU.

Key CPU architectural innovations include index register, cache, virtual memory, instruction pipelining, superscalar, CISC, RISC, virtual machine, emulators, microprogram, and stack.

Micro-architectural concepts

Research topics

A variety of new CPU design ideas have been proposed, including reconfigurable logic, clockless CPUs, computational RAM, and optical computing.

Performance analysis and benchmarking

Benchmarking is a way of testing CPU speed. Examples include SPECint and SPECfp, developed by Standard Performance Evaluation Corporation, and ConsumerMark developed by the Embedded Microprocessor Benchmark Consortium EEMBC.

Some of the commonly used metrics include:

- Instructions per second - Most consumers pick a computer architecture (normally Intel IA32 architecture) to be able to run a large base of pre-existing pre-compiled software. Being relatively uninformed on computer benchmarks, some of them pick a particular CPU based on operating frequency (see Megahertz Myth).
- FLOPS - The number of floating point operations per second is often important in selecting computers for scientific computations.
- Performance per watt - System designers building parallel computers, such as Google, pick CPUs based on their speed per watt of power, because the cost of powering the CPU outweighs the cost of the CPU itself.[1][2]
- Some system designers building parallel computers pick CPUs based on the speed per dollar.
- System designers building real-time computing systems want to guarantee worst-case response. That is easier to do when the CPU has low interrupt latency and when it has deterministic response. (DSP)
- Computer programmers who program directly in assembly language want a CPU to support a full featured instruction set.
- Low power - For systems with limited power sources (e.g. solar, batteries, human power).
- Small size or low weight - for portable embedded systems, systems for spacecraft.
- Environmental impact - Minimizing environmental impact of computers during manufacturing and recycling as well during use. Reducing waste, reducing hazardous materials. (see Green computing).

There may be tradeoffs in optimizing some of these metrics. In particular, many design techniques that make a CPU run faster make the "performance per watt", "performance per dollar", and "deterministic response" much worse, and vice versa.

Markets

There are several different markets in which CPUs are used. Since each of these markets differ in their requirements for CPUs, the devices designed for one market are in most cases inappropriate for the other markets.

General purpose computing

The vast majority of revenues generated from CPU sales is for general purpose computing, that is, desktop, laptop, and server computers commonly used in businesses and homes. In this market, the Intel IA-32 architecture dominates, with its rivals PowerPC and SPARC maintaining much smaller customer bases. Yearly, hundreds of millions of IA-32 architecture CPUs are used by this market. A growing percentage of these processors are for mobile implementations such as netbooks and laptops.[3]

Since these devices are used to run countless different types of programs, these CPU designs are not specifically targeted at one type of application or one function. The demands of being able to run a wide range of programs efficiently has made these CPU designs among the more advanced technically, along

with some disadvantages of being relatively costly, and having high power consumption.

High-end processor economics

In 1984, most high-performance CPUs required four to five years to develop.[4]

Scientific computing

Scientific computing is a much smaller niche market (in revenue and units shipped). It is used in government research labs and universities. Before 1990, CPU design was often done for this market, but mass market CPUs organized into large clusters have proven to be more affordable. The main remaining area of active hardware design and research for scientific computing is for high-speed data transmission systems to connect mass market CPUs.

Embedded design

As measured by units shipped, most CPUs are embedded in other machinery, such as telephones, clocks, appliances, vehicles, and infrastructure. Embedded processors sell in the volume of many billions of units per year, however, mostly at much lower price points than that of the general purpose processors.

These single-function devices differ from the more familiar general-purpose CPUs in several ways:

- Low cost is of high importance.
- It is important to maintain a low power dissipation as embedded devices often have a limited battery life and it is often impractical to include cooling fans.[5]
- To give lower system cost, peripherals are integrated with the processor on the same silicon chip.
- Keeping peripherals on-chip also reduces power consumption as external GPIO ports typically require buffering so that they can source or sink the relatively high current loads that are required to maintain a strong signal outside of the chip.
    - Many embedded applications have a limited amount of physical space for circuitry; keeping peripherals on-chip will reduce the space required for the circuit board.
    - The program and data memories are often integrated on the same chip. When the only allowed program memory is ROM, the device is known as a microcontroller.
- For many embedded applications, interrupt latency will be more critical than in some general-purpose processors.

Embedded processor economics

The embedded CPU family with the largest number of total units shipped is the 8051, averaging nearly a billion units per year.[6] The 8051 is widely used because it is very inexpensive. The design time is now roughly zero, because it is widely available as commercial intellectual property. It is now often embedded as a small part of a larger system on a chip. The silicon cost of an 8051 is now as low as US$0.001, because some implementations use as few as 2,200 logic gates and take 0.0127 square millimeters of silicon.[7][8]

As of 2009, more CPUs are produced using the ARM architecture instruction set than any other 32-bit instruction set.[9][10] The ARM architecture and the first ARM chip were designed in about one and a half years and 5 human years of work time.[11]

The 32-bit Parallax Propeller microcontroller architecture and the first chip were designed by two people in about 10 human years of work time.[12]

The 8-bit AVR architecture and first AVR microcontroller was conceived and designed by two students at the Norwegian Institute of Technology.

The 8-bit 6502 architecture and the first MOS Technology 6502 chip were designed in 13 months by a group of about 9 people.[13]

Research and educational CPU design

The 32 bit Berkeley RISC I and RISC II architecture and the first chips were mostly designed by a series of students as part of a four quarter sequence of graduate courses.[14] This design became the basis of the commercial SPARC processor design.

For about a decade, every student taking the 6.004 class at MIT was part of a team—each team had one semester to design and build a simple 8 bit CPU out of 7400 series integrated circuits. One team of 4 students designed and built a simple 32 bit CPU during that semester. [15]

Some undergraduate courses require a team of 2 to 5 students to design, implement, and test a simple CPU in a FPGA in a single 15 week semester. [16]

The MultiTitan CPU was designed with 2.5 man years of effort, which was considered "relatively little design effort" at the time.[17] 24 people contributed to the 3.5 year MultiTitan research project, which included designing and building a prototype CPU.[18]

Soft microprocessor cores

For embedded systems, the highest performance levels are often not needed or desired due to the power consumption requirements. This allows for the use of processors which can be totally implemented by logic synthesis techniques. These synthesized processors can be implemented in a much shorter amount of time, giving quicker time-to-market.

See also

- Central processing unit
- Comparison of instruction set architectures
- History of general-purpose CPUs
- Microprocessor
- Microarchitecture
- Moore's law
- Amdahl's law
- System-on-a-chip
- Reduced instruction set computer
- Complex instruction set computer
- Minimal instruction set computer
- Electronic design automation

Wikibooks has a book on the topic of: *Microprocessor Design*

■ High-level synthesis

References

1. "EEMBC ConsumerMark". Archived from the original on March 27, 2005.
2. Stephen Shankland (December 9, 2005). "Power could cost more than servers, Google warns".
3. Kerr, Justin. "AMD Loses Market Share as Mobile CPU Sales Outsell Desktop for the First Time." (http://www.maximumpc.com/article/news/amd_los es_market_share_mobile_cpu_sales_outsell_desktop _first_time) Maximum PC. Published 2010-10-26.
4. "New system manages hundreds of transactions per second" article by Robert Horst and Sandra Metz, of Tandem Computers Inc., "Electronics" magazine, 1984 April 19: "While most high-performance CPUs require four to five years to develop, The NonStop TXP processor took just 2+1/2 years -- six months to develop a complete written specification, one year to construct a working prototype, and another year to reach volume production."
5. S. Mittal, "A survey of techniques for improving energy efficiency in embedded computing systems (https://www.academia.edu/4186102/A_survey_of_t echniques_for_improving_energy_efficiency_in_em bedded_computing_systems)", IJCAET, 6(4), 440–459, 2014.
6. http://people.wallawalla.edu/~curt.nelson/engr355/lect
7. Square millimeters per 8051, 0.013 in 45nm line-widths; see (http://www.keil.com/dd/docs/datashts/evatronix/t80 51_ds.pdf)
8. To figure dollars per square millimeter, see [1] (http://www.overclockers.com/forums/showthread.p hp?t=550542), and note that an SOC component has no pin or packaging costs.
9. "ARM Cores Climb Into 3G Territory" (http://www.extremetech.com/extreme/52180-arm-cores-climb-into-3g-territory) by Mark Hachman, 2002.
10. "The Two Percent Solution" (http://www.embedded.com/electronics-blogs/significant-bits/4024488/The-Two-Percent-Solution) by Jim Turley 2002.
11. "ARM's way" (http://web.archive.org/web/20090606152116/http:// atterer.net/acorn/arm.html) 1998
12. "Why the Propeller Works" (http://www.parallax.com/Portals/0/Downloads/docs /article/WhythePropellerWorks.pdf) by Chip Gracey
13. "Interview with William Mensch" (http://silicongenesis.stanford.edu/transcripts/mensc h.htm)
14. 'Design and Implementation of RISC I' (http://www.eecs.berkeley.edu/Pubs/TechRpts/1982/ CSD-82-106.pdf) - original journal article by C.E. Sequin and D.A.Patterson
15. "the VHS" (http://sub-zero.mit.edu/fbyte/hacks/vhs/)
16. "Teaching Computer Design with FPGAs" (http://www.fpgacpu.org/teaching.html) by Jan Gray
17. Norman P. Jouppi and Jeffrey Y. F. Tang. "A 20-MIPS Sustained 32-bit CMOS Microprocessor with High Ratio of Sustained to Peak Performance" (http://citeseerx.ist.psu.edu/viewdoc/download? doi=10.1.1.85.988&rep=rep1&type=pdf). 1989. doi:10.1.1.85.988 (https://dx.doi.org/10.1.1.85.988). p. "i".
18. "MultiTitan: Four Architecture Papers" (ftp://ftp.hpl.hp.com/pub/DEC/WRL/research-reports/WRL-TR-87.8.pdf). 1988. p. 4-5.

■ Hwang, Enoch (2006). *Digital Logic and Microprocessor Design with VHDL*. Thomson. ISBN 0-534-46593-5.
■ Processor Design: An Introduction (http://www.gamezero.com/team-0/articles/math_magic/micro/index.html)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Processor_design&oldid=713757757"

Categories: Central processing unit │ Computer engineering

- This page was last modified on 5 April 2016, at 18:43.
- Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.