New Advances in Micro-Processors and computer architectures

Prof. (Dr.) K.R. Chowdhary, Director SETG
Email: kr.chowdhary@jietjodhpur.com

Jodhpur Institute of Engineering and Technology, SETG

August 27, 2015

Micro-processors

- Evolved from Transistors and ICs
- Icon of the Informations Age
- Used in fastest computers to simple electronics toys
- Well over 1 billion transistors in a micro-processor.

Earliest applications

- μP has pushed the technology of today
- Desire for every increasing performance led to rapid improvement in technology
- This enabled more complex micro-processors, fabrication processes, design methodology
- Earliest MPs filled the needs of embedded applications
- The history of μPs can be divided into five stages:
  1. The birth of the microprocessor,
  2. The first microcomputers,
  3. A leading role for the microprocessor,
  4. The promise of reduced instruction set computer (RISC), and
  5. Microprocessors after yr 2000.

The birth of the microprocessor

- 4004 - first commercially available microprocessor
- In 1968, Moore and Noyce left Fairchild to form Intel Corporation.
- Intel's history was forever changed by the development of the 4004 for the Busicom calculator company (1971)
- 8008, 200 kHz, 60,000 instructions / sec, 3,500 transistors, 14-bit address, Dec. 1971

The chip set consisted of four chip types:

- the 4001 ROM,
- the 4002 random access memory (RAM) register memory,
- the 4003 I/O shift register, and
- 4004 central processing unit (CPU).

8080: 1972, 6800: 1974

- 8080: More instructions, 64 K address bus, 8-bit IO
- 8080: 6000 transistors,
- 6800: 4000 Transistors
- Z80: outperformed 080
- 8086, 68000
- 8086: Supported wide variety of 8-bit peripheral
- 68000: it did not
- IBM got right to manufacture 8086, hence used in PCs

Software for a New Industry

- It was not MS-DOS, but unix and C led the advancement of next gen. micro-processors
- 1981: IBM brought PC with 8088, 64K RAM
- 1984: PC with GUI, using 68000
- Early in 1980 C, UNIX gave rise to a new architecture: RISC

16-bit, 32-bit, and early RISC MP

- Need to be compatible with 8-bit softwares
- extending memory beyond 64 k
- 8086 supported pipeline of instructions
- Each segment of 16K, 16 segments
- 68000: Fetched one more 16-bit words,
- 32-bit addr and data registers
- 68010: Support for virtual memory
- 68020: Pipe-line overlapped fetch and execute cycles
- On-chip 256 byte instruction cache

Beginning of RISC

- Early 1980s: stage was set in academia for next phase of micro-processor evolution
- RISC movement began in reaction to the complexity of a minicomputer architecture (VAX from DEC)
- Unlike contemporary and earlier complex instruction set computer (CISC) processors, the RISC projects endorsed fixed-length, 32-bit instructions,
- no memory-to-memory instructions (RISC used a load/store architecture),

Beginning of RISC...

- large, general-purpose register files, and pipelining.
- RISC projects formalized a fundamental performance metric for computer architectures:
  1. The amount of CPU time required to execute a given task.
  2. Expressed by: CPU time = instruction count x clock cycles per instruction (CPI) x clock cycle time.
  3. A typical CISC had three or four, while RISCs approached the goal of achieving one cycle per instruction

Beginning of RISC...

- The VAX, IBM 370, and other CISC architectures were characterized by a small subset of frequently used instructions, with many other instructions rarely used
- The project teams at Berkeley and Stanford extensively analyzed the instruction usage characteristics of compiled programs.
- They found that most applications had surprising commonality in their instruction execution and data access patterns

Beginning of RISC...

- From this analysis, the Berkeley group designed RISC I and II based on a large register file, divided into overlapping windows for the stack frames used by the compiler.
- The RISC processors led in introducing pipelining in microprocessors, with a two-stage pipeline for RISC I and a three-stage pipeline for RISC II.
- The RISC I/II ideas found later commercial application in Sun's SPARC* architecture.

Beginning of RISC...

- The Berkeley team recognized the need to tailor the architecture to the compiler and to tune the compiler to the needs of the hardware.
- The notion of using the compiler to address the problem of branch latency (branch delay slots) was used at both Berkeley and Stanford.
- These projects were among the first attempts to treat the compiler and microprocessor as a single system, trading hardware for compiler complexity.
- At Stanford, the microprocessor without interlocking pipe stages (MIPS) project took optimizing compiler technology further.

The Promise of RISC

- first commercial RISC CPU (MIPS* R2000) in 1986.
- The workstation manufacturers abandoned Motorola 68K CPUs in favor of their own RISC CPUs.
- With the threat of RISC, even Intel and Motorola designed their own RISC processors, while continuing to supply their flagship CISC processors in increasing volumes to the cost-sensitive PC market, which required compatibility.
- The RISC processors, on the other hand, were targeted at the performance-oriented UNIX workstation market, where price was secondary.
- This set the stage for the battle between price and performance.

Architectural Features

- Pipelines deepened from the simple overlap of fetch, decode, and execute stages (characteristic of the Intel 80386 and Motorola 68030 CPUs) to over five stages (typical of the RISC CPUs) (1.fetch, 2.decode & read reg. file, 3.execute of calcu. addr., 4.load/store operands, 5.write register file).
- Data and instruction caches were incorporated on chip, along with memory management and cache-control functions.
- FPUs were also integrated by the late 1980s. The push to integrate was more pronounced in the CISC processors.

Architectural Features...

- The RISC CPUs, which attempted to execute one instruction per cycle, relied on large, fast caches.
- All these architectural features were enabled by the predictable advance of IC technology. For example, the number of transistors increased from 275K in the Intel 80386DX to 1.2M in the Intel 80486DX.

Architectural Features

- The Intel 80386 and the Motorola 68030 (introduced in 1987) were considered to be second-generation CISC processors with limited pipelining.
- The 80386 provided a fully binary-compatible upgrade to Intel's first-generation processors (the 8086, 80186, and 80286).
- The new base+index+displacement addressing mode allowed the full 32-bit memory space to be easily addressed, a great improvement over the 64-KB segment limitation of the previous generation.
- Internally, it had a Harvard architecture (separate buses for fetching data and instructions) with separate 256-byte caches.
- Both the 80386 and the 68030 had three-stage pipelines and were clocked at 20 MHz.

RISC CPUs

- The new commercial RISC CPUs were similar to the design established by Berkeley RISC and Stanford MIPS.
- Instructions are all 32 bits wide. Register files typically had thirty-two 32-bit general-purpose registers.
- The opcodes provided only the basic instructions. The only instructions that accessed memory and the memory-mapped I/O space were load and store instructions, hence the name load/store architecture.
- Memory was addressed by register plus displacement or register plus register.

RISC CPUs...

- The number of addressing modes was fewer than previous CISC CPUs and few data types were supported.
- The typical RISC CPU had a five-stage pipeline.
- Each stage of the pipeline performed its processing in one clock, taking inputs, stored in registers, from the previous stage and storing its results in registers to be processed by the next stage.
- In the absence of branches, assuming all instructions and data were in cache, and all instructions took only one clock to execute, the pipeline remained full and proceeded without stalling, yielding an ideal CPI of one.

RISC CPUs...

- Note that the goal of the processor designer and compiler writer is to prevent stalls as much as possible.
- Decoding was also simplified compared to the CISC CPUs, by having fewer opcodes and eliminating complex instructions.
- Sun Microsystems developed the SPARC architecture based on Berkeley RISC for its own workstations
- The unique feature of SPARC was the windowed register file, which reduced memory traffic caused by saving and restoring registers on procedure calls.
- Many others as RISC: Alpha (DEC), HP, MIPS, SUN.

Future Directions

- The expected performance enhancements delivered by microprocessors may slow down in the future because of problems associated with IC technology, computer architecture, and market forces.
- The shrinking line widths of the next-generation ICs will require new lithographic techniques to draw finer lines. Thus far, the intrinsic delay of the transistor itself has been reduced to enable commensurate increases in clock speed.

Future Directions

- Limitations in exploiting parallelism must also be overcome.
- Applications need to be written in a manner that exposes more parallelism. This is already under way with multi-threading.
- Advances in compilers will find parallelism across larger sections of a program than previously possible.
- Due to limitation of further increase in clock rate, multi-core processors arrived, to realize the parallelism.
- However, the software development lacked: no parallel programming languages, no parallel compilers, hence the resource utilization lacked!

Future Directions

- Future lies with multi-cores, and parallel compilers, parallel programming languages.
- C to CETUS source-to-source compilers.
- C to Cuda compilers

Some Questions

- How to implement calls without stack?
- What is difference between Hardwired and Micro-controlled control design?
- Why parallel programming is important?
- What is (DDM) Data driven multi-threading?
- What is language for hardware design ?
- What is branch prediction?
- What is cache coherence?
- Simulators for teaching architectures: gnusim8085, pdp8 simulator,