Comp4611 Tutorial 2

Instruction Set Architecture (ISA)

Sept. 24, 26, 28 2012

Computer Architecture

- Instruction set architecture (ISA)
  - The actual programmer-visible instruction set and serving as the boundary between the software and hardware.
- Organization
  - Includes the high-level aspects of a computer's design such as: The memory system, the bus structure, and the internal CPU unit.
- Hardware
  - Refers to the specifics of the machine such as detailed logic design and packaging technology.

Instruction Set Architecture

- "Instruction Set Architecture is the structure of a computer that a machine language programmer (or a compiler) must understand to write a correct (timing independent) program for that machine."

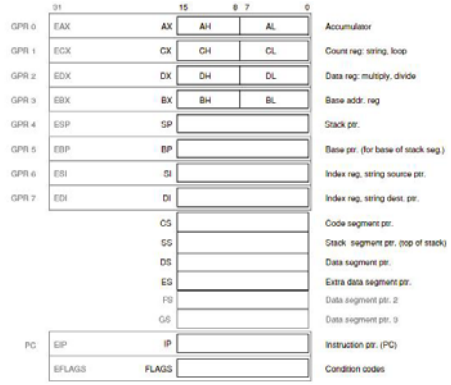
  IBM, Introducing the IBM 360 (1964)

- The ISA defines:
  - Operations that the processor can execute
  - Data Transfer mechanisms + how to access data
  - Control Mechanisms (branch, jump, etc)
  - "Contract" between programmer/compiler and hardware

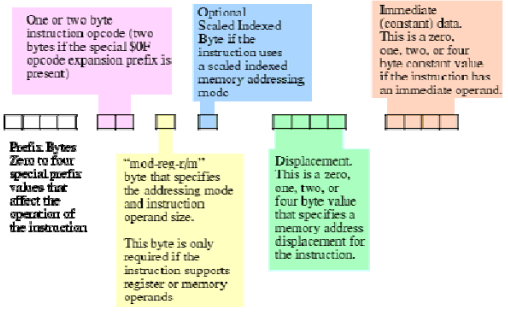
More About ISA

- Example
  - Intel 80x86 family use the similar ISA. The later generation has the ISA covering that of the former generation.
- Benefit
  - Old software can be used on the new hardware and vice versa (backwards compatibility).
- Requirement
  - ISA can provide convenient functionality to higher level (software view).
  - ISA should permit efficient implementation at lower level (hardware view).

Review of 80x86 Registers (IA-32)



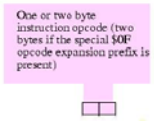
80x86 Instruction Encoding

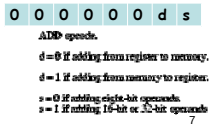


Hybrid length

80x86 Operation Code (opcode)

- The 80x86 supports two basic opcode sizes
  - Standard one-byte opcode
  - Or two-byte opcode
    - Consisting of a $0F opcode expansion prefix byte and a second byte specifying the actual instruction.
  - Provides for up to 512 different instruction classes
    - Although the 80x86 does not yet use them all ← Expandability
- In reality, certain bits in opcode may used for non-instruction-class purposes
  - i.e. 's' bit specifies the size of the operand, 'd' specifies the direction of the transfer

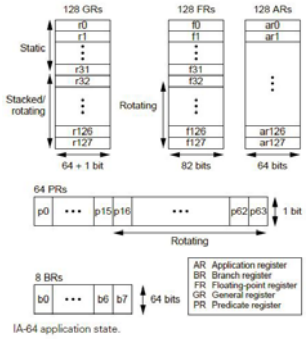




Move on: IA-64

- Incompatible to IA-32
- Adopted in Intel Itanium® processor family
- A full 64-bit address space
- Large directly accessible registers
- 128-bit encoding that has room for three instructions
- Instruction bits to communicate information from the compiler to the hardware
- Trade-offs runtime and compilation time
- New ways of prediction

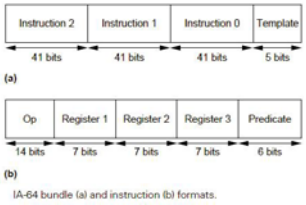
IA-64 registers



IA-64 application state.

- 128 65-bit general registers
- 128 82-bit floating-point registers
- 128 64-bit special-purpose application registers
- 8 64-bit branch registers
- 64 one-bit predicate registers

IA-64 instruction formats



IA-64 bundle (a) and instruction (b) formats.

IA-64 passes the burden of parallelism-hunting back to the compiler. The compiler takes all of your sequential code, examines it for dependencies, shuffles it around, then packs it into 128-bit "bundles" which can safely be executed in parallel. Each 128-bit bundle contains three instructions and a set of template bits.

Move on: Intel® EM64T

- Enhancement to IA-32 architecture
- Legacy Mode : 32-bit O/S and 32-bit applications
- Compatibility Mode : 64-bit O/S and 32-bit applications
- 64-bit Mode : 64-bit O/S and 64-bit applications

Software Visible Register	64-Bit Mode			Legacy and Compatibility Modes		
	Name	Number	Size (bits)	Name	Number	Size (bits)
General Purpose Registers	RAX, RBX, RCX, RDX, RBP, RSI, RDI, RSP, R8-15	16	64	EAX, EBX, ECX, EDX, EBP, ESI, EDI, ESP	8	32
Instruction Pointer	RIP	1	64	EIP	1	32
Flags	EFLAGS	1	32	EFLAGS	1	32
FP Registers	ST0-7	8	80	ST0-7	8	80
Multi-Media Registers	MM0-7	8	64	MM0-7	8	64
Streaming SIMD Registers	XMM0-15	16	128	XMM0-7	8	128
Stack Width	-		64	-		16 or 32

Intel® IA-64 vs Intel® EM64T

- **Is Intel® EM64T the same technology used in the Itanium® 2 processor?**
- No.
- Intel® EM64T is an extension to Intel's processors based on the IA-32 architecture.
- The Itanium processor family is based on the EPIC architecture.
- These are two separate families of processors, based on two different architectures.
- The Itanium processor family is specifically designed for the most demanding mission-critical applications.

The evolvement of ISA:
Practical Example: Multimedia ISAs

- Motivation:
  - Demand (or to be created by Intel) of multimedia/ graphics/ communication processing
- Observation:
  - The common processing involve repetitive integer and FP operations
  - In original x86 architecture, those operations require a large number of instructions
- Solution:
  - Intel tries to design a new instruction set to speedup the operations
  - Put those operations into a single hardware instruction (also known as vector operations)

Move on

- Intel MMX ('97)
  - **M**ulti**M**edia e**X**tensions, a set of instructions built into Intel microprocessors and other x86-compatible microprocessors to handle many common multimedia operations.
- SSE ('99)
  - **S**treaming **S**IMD **E**xtensions is introduced in Pentium III series processors to enable the performance of 3D and video applications. SSE
  - Added 70 new instructions.
- SSE2 ('01)
  - An extension to the basic SSE instruction set to solve the pipeline problem of issuing FPU (float process unit) and SSE instructions at the same time.
  - Added 144 new instructions to SSE.
- SSE3 ('04)
  - The third iteration of the SSE instruction set for the IA-32 architecture.
  - Capability to work horizontally in a register.

Intel® Pentium® 4 with SSE3

- **13 new streaming instructions that will increase the performance**
  - **Added instructions that add and subtract the multiple values stored within a single register. These instructions simplify the implementation of a number of DSP and 3D operations.**
  - **A new instruction that convert floating point values to integers without having to change the global rounding mode, thus avoiding costly pipeline stalls.**

Intel® Core™ Microarchitecture

http://www.youtube.com/watch?v=4BDfi_sdnFc

- Wide Dynamic Execution
  - Deliver more instructions per clock cycle
  - Allow each core to fetch, dispatch, execute and retire up to four full instructions simultaneously

  → improving execution and energy efficiency

- Intelligent Power Capability
  - Intelligently turns on only the subsystem that are required

  → reduce power consumption

Intel® Core™ Microarchitecture

- Advanced Smart Cache
  - Each execution core can access data from the faster, more efficient cache subsystem

  → improves processor efficiency and performance
- Smart Memory Access
  - Load or pre-fetch data in advance
- Advanced Digital Media Boost
  - The 128-bit SSE instructions are now issued at a throughput rate of one per clock cycle

  → accelerating a broad range of applications, including video, speech and image, photo processing, encryption, financial, engineering and scientific applications.

ISA and uArchitecture

- Both Intel and AMD adopt 80x86 ISA
  - Interoperability of programs
- Different in performance as the underlying microarchitectures are different
  - Optimization issue for different CPUs

Intel 64/x86-64

- **The x86-64** architecture is a 64-bit superset of the 32-bit x86 ISA
- x86-64 was designed by AMD who named it **AMD64**
- It has been cloned by Intel under the name **Intel 64**
- The vendor-neutral names *x86-64* or x64
- All instructions in the x86 instruction set can be executed by x86-64 CPUs

Intel 64/x86-64

- **Intel 64/x86-64** should not be confused with the Intel Itanium architecture known as ***IA-64*** which is not compatible at native instruction set level x86-64
- **Intel 64** is Intel's implementation of x86-64.
- Used in newer versions of Pentium 4, Celeron D, Xeon and Pentium Dual-Core processors, the Atom D510, N450, N550, N2600 and N2800 and in all versions of the Pentium Extreme Edition, Core 2, Core i7, Core i5, and Core i3 processors.

Intel 64/x86-64

- **Additional registers**
  - The number of named registers is increased from 8 (i.e. eax, ebx, ecx, edx, ebp, esp, esi, edi) to 16
- **Larger virtual address space**
  - Current models can address up to 256 terabytes
  - Expandable in the future to 16 exabytes
  - Compared to just 4 gigabytes for 32-bit x86