

Computer Architecture (instruction set + hardware component + system organization)

1. Instruction set architecture (ISA)

- กำหนด computational characteristics ของ computer

2. Hardware system architecture (HSA)

- จัดการ hardware subsystem
 - central processing unit (CPU)
 - storage system
 - input/output system (I/O)

Computer-family architecture = set of implementation ที่มี ISA เหมือนกัน

Compatibility = ความสามารถที่ computer คนละแบบกัน สามารถ run program เดียวกันได้

- upward compatibility = ความสามารถที่จะทำให้ high-performance family รันโปรแกรมเดียวกับ lower-performance family ได้

Historical Perspective

- First Generation - Vacuum Tube
 - ABC - Iowa State University - John Atanasoff
 - ENIAC - University of Pennsylvania
 - EDSAC - University of Cambridge
 - MARK-I, MARK-II, MARK-III, MARK-IV - Harvard University
 - * Univac ผลิตเพื่อการค้าครั้งแรก ใช้ vacuum-tube
- Second Generation - Transistor
 - * John Bardeen and friends จาก Bell Laboratories สร้าง transistor
- Third Generation - Small-scale integration (SSI)
- Fourth Generation - Very-large-scale integration (VLSI)

Computer ประกอบด้วย

- CPU
 - control unit
ควบคุมการทำงานของ CPU
 - arithmetic and logic unit (ALU)
Arithmetic, logical, shift operation
 - register set
เก็บข้อมูลขณะทำงาน
 - program counter (PC) หรือ Instruction pointer (IP)
เก็บ address ของคำสั่ง (PC เป็นส่วนหนึ่งของ register)

Instruction

- Instruction fields บอกรายละเอียดกับ control unit
- Instruction format
- Instruction size บอกขนาดของคำสั่ง มีหน่วยเป็น byte
- Operation code (op code) เป็นตัวบอกว่าจะทำ operation อะไร
- ส่วนอื่นๆ ที่บอกว่าจะใช้ register อะไร

Instruction set = set of instruction ที่สามารถ execute ได้

Data stream = sequence of data ที่ cpu ทำงานด้วย

Program = sequence of instruction

- Main memory
- I/O system

Machine cycle

- instruction fetch
- instruction execute

หมวดหมู่ของ computer architecture

- single instruction stream, single data stream
- single instruction stream, multiple data stream
- multiple instruction stream, single data stream
- multiple instruction stream, multiple data stream

Multiprocessor architecture มี 2 categories

1. Global memory
 - processor ใช้ memory ร่วมกัน
2. Local memory
 - processor แต่ละตัวมี memory ของตัวเอง

SIMD machines

- 1 CU มีหลาย processing element

MIMD machines

- แต่ละ processor ทำงานเป็นอิสระต่อกัน

คุณสมบัติของ Computer Architecture ที่ดี

1. Generality สามารถทำงานได้กับหลากหลาย application
2. Applicability
3. Efficiency
4. Ease of Use ง่ายสำหรับ system programmer
5. Malleability ง่ายในการสร้าง computer ที่มีขนาดหลากหลาย
6. Expandability ง่ายสำหรับการเพิ่มความสามารถ

ปัจจัยที่มีผลต่อความสำเร็จของ computer architecture

1. Architecture Merit
 - Applicability ความเหมาะสมในการใช้งาน
 - Malleability ความง่ายในการใช้งาน
 - Expandability ความสามารถในการขยาย memory, I/O, processor
 - Compatibility สามารถใช้แทนรุ่นก่อนๆ ใน family
2. System Performance ใช้ benchmarks ในการวัด
 - CPU Performance Measures
 - MIPS millions of instructions per second
 - MFLOPS millions of floating-point operation per second
 - GFLOPS gigaflops billion of floating-point operations per second
 - I/O Performance Measures
 - bandwidth ความเร็ว หน่วยเป็น MBS (megabytes per second)
 - I/O operation per second
 - Other Performance Measures
 - Memory bandwidth ความเร็วในการส่งข้อมูลใน processor MBS
 - Memory access time เวลาที่ CPU ใช้ในการเข้าถึงข้อมูล
 - Memory size
3. System cost
 - ราคา hardware
 - Reliability ความน่าเชื่อถือ
 - Ease of repair
 - Power consumption
 - Weight
 - Ruggedness ทนต่อสภาพแวดล้อม
 - Software system interface

Instruction-set Architecture

- data type ชนิดของข้อมูล
- hardware
- register
- machine instruction

Data Representation

- unit of information
 - 1 byte = 8 bits
 - 16-bit computer : 1 word = 2 bytes = 16 bits
 - 32-bit computer : 1 word = 4 bytes
 - quad word = 4 words
 - octet = 8 word
- - single precision 1 word
 - double precision 2 word
 - character 1 byte (ASCII)
 - numeric 8,4 bit / หลัก (BCD)
 หลาย byte (integer)

- Integers

- Unsigned-binary numbers
- Binary-Coded Decimals (BCD)
 ใช้ 4 bit เก็บเลขฐาน 10 1หลัก
- Signed-magnitude integer
 - sign bit 0 (+), 1 (-)
- One's Complements
- Two's Complements
- Excess -n Integer
 Biased integer = k + n

- Fraction

- binary fraction
 $0.1001 = 1/2 + 0/4 + 0/8 + 1/16 = 9/16$

- Floating-point Numbers

- sign (S)
- mantissa (M)
- exponent (E)
- radix (R)
 $F = (-1)^S * M * R^E$

- IEEE floating-point standard

- single precision IEEE standard 754
 - S 1bit
 - E 8bit excess-127
 - M 23bit
 - R = 8
- double precision IEEE standard 754
 - S 1bit
 - E 11bit excess-1023
 - M 52bit
 - R=8

- Data Structures

- character string
 - LENGTH
 - EQUAL
 - CONCAT
 - SUBSTRING
- stack
 - push
 - pop
 - top
 - empty
- arrays
- parameter passage structures
 1. ใส่ parameter ลงใน list แล้วส่ง address ของ list
 2. ใส่ address param ลงใน list แล้วส่ง address ของ list
 3. push parameter ลงใน stack
 4. push address parameter ลง stack
 5. ส่ง parameter ทาง register

Register

- memory-address register (MAR)
- memory-buffer register (MBR) เก็บข้อมูลที่นำมาจาก memory
- Register Operation
 - memory-to-register instructions
 - memory-to-memory instructions
 - register-to-memory instructions
 - register-to-register instructions
- Register Architecture
 - address instruction
 - 0-address instruction implicit operands, result *ทำงานได้เร็ว แต่ไม่เหมาะกับ str
 - 1-address instruction 1 memory operand
 - N-address instruction N memory operand
 - M+N address instruction M operand และ N branch address
 - Evaluation stack architectures
 - เน้นส่งผ่าน stack จะไม่ค่อยใช้ parameter
 - Accumulator Machine
 - General-purpose Register-set Machine
 - เครื่องที่มี general purpose register
 - Special-purpose Register-set Machine
 - มี set index register และ set operand

Instruction

- type of instructions
 1. Operate Instruction
 - Arithmetic, Logical, Shift instruction
 - Character and String instruction
 - Stack and Register Manipulation
 2. Memory Access Instruction
 - Load, Store instruction
 - Load address instruction ใช้เพื่อหา address ของข้อมูลหรือคำสั่ง

3. Control Instruction

- Conditional, Unconditional Branch instructions
- Conditional-code setting instruction เปลี่ยนแปลง status bit
- Subroutine-linkage instruction

4. Miscellaneous and privileged Instruction

- I/O instruction
- Interrupts
- Exception
- Privileged instruction

- Vector Instruction

- vector LOAD, STORE, SQUARE, SQUARE ROOT, NEGATE, comparison, merge

- Addressing Techniques

- operand

- immediate operand ค่าอยู่ในคำสั่ง
- register operand ค่าอยู่ใน register
- memory operand ค่าอยู่ใน memory

- register addressing

- register designator(register address) 4-6 bits ให้ระบุ register

- boundary Alignment

-

- Memory addressing

- address space

- compiler จะกำหนด address ให้ตัวแปร เรียกว่า logical address
- compilation เปลี่ยน โปรแกรม -> machine instruction,
Logical address -> instruction address

- absolute addressing

- indexed addressing

มี register ที่เก็บ index ใช้ร่วมกับ address ที่อยู่ใน operand address

- indirect addressing

address ของ operand เก็บอยู่ใน memory

- indexed-indirect addressing

- preindexed indirect addressing indexing มาก่อน indirect
- postindexed indirect addressing indexing ภายหลัง indirect

- base-displacement addressing

-

- PC-relative addressing

- ใช้ address ใน program counter เป็น base และ operand เป็น offset

- stack-register addressing

- STORE = PUSH, LOAD = POP

- autoincrement and autodecrement addressing

- predecrementing
- postincrementing

- segment register addressing

Addressing design issues

- Physical-address range

ขนาดสูงสุดของ memory ที่อ้างถึงได้ (ถูกจำกัดโดยจำนวน bit ใน physical address)

- Addressing Efficiency

ประสิทธิภาพในการใช้ addressing mode (RISC มี mode ง่ายๆไม่มากม CISC มี mode ยากๆ)

Instruction-set Design

- Completeness

ต้องเพียงพอกับการใช้งานทั่วไป, ครอบคลุมการทำงานพื้นฐาน

- Orthogonality

ไม่มีคำสั่งซ้ำซ้อน

- Compatibility

- source-code compatibility

สามารถ execute ในเครื่องต่างชนิดได้ โดยไม่ต้อง recompiled, reload, relinked

- object-code compatibility

สามารถ execute ในเครื่องต่างชนิดได้ โดยไม่ต้อง recompiled

* portable vs compatibility

- portable สามารถ run ได้ในเครื่องต่างชนิดกัน

* portable ต้อง recompiled

- Instruction Formats

การใช้ op code ที่น้อยไปอาจได้คำสั่งน้อย แต่ถ้ามากไปก็กิน memory

Buses, the CPU and the I/O system

- Buses

ขนส่งข้อมูลระหว่าง component หรือ subsystem

- Bus type

1. Local buses

- ใน CPU

- address bus

Unidirectional ส่ง address จาก PC, stack register ไปยัง memory

- data bus

Bidirectional ส่งข้อมูล, คำสั่ง, address ระหว่าง memory, I/O และ ALU

- control bus

Bidirectional ส่งสัญญาณเพื่อไปควบคุมส่วนประกอบอื่นๆ

2. System bus

- มี controller ของตัวเอง เรียก bus controller

- ใน bus controller มี arbiter เป็นตัวจัดการคำขอใช้ bus

- device จะใช้ bus ได้ต้องได้รับการยินยอมจาก bus arbiter

- โดยทั่วไปใช้เชื่อม CPU, I/O, main-memory system

3. Expand Local Buses

- เป็น local bus ที่ขยายออกไปใช้ข้างนอก CPU ได้

Bus transfer and Control Signal

- device แข่งขันกันเพื่อใช้ bus เรียกว่า bus master

- device ส่ง bus-request signal ไปทาง bus-request line ไปยัง bus arbiter

- device จะได้สิทธิเป็น bus master และมีสิทธิใช้ 1 bus cycle

- ขั้นตอนเหล่านี้เรียกว่า bus protocol

Central Processing Unit (CPU)

- ALU

- มี status register

- control bus รับสัญญาณมาจาก control unit

- status bus ส่งสัญญาณไปยัง control unit

- ส่งข้อมูลทาง input/output data bus (local data bus)

- Control Unit
 - machine cycle
 - * PC เก็บ address instruction ถัดไป
 - fetch
 - IR = memory[PC]
 - PC++
 - decode, execute
 - CU จะส่ง microorder ทาง dedicated control line เพื่อควบคุม device
 - การสร้าง microorders set เรียกว่า microinstruction
 - CU type
 - Microprogrammed Control Unit
 - * fetch คำสั่งมาแล้วแปลงเป็น microinstruction
 - โดยใช้ microprogram translator
 - แปล op code -> microinstruction address
 - microinstruction processor
 - ให้ address ที่ได้ fetch microinstruction ส่งมาทาง control bus line
 - ประกอบด้วย
 - IR เก็บ instruction ที่จะถูก execute
 - control store เก็บ microprogram
 - address-computation circuitry กำหนด address ใน control store
 - microprogram counter เก็บ address microinstruction ถัดไป
 - microinstruction buffer เก็บ microinstruction จาก store
 - microinstruction decoder generate microorder
 - sequencer synchronize การทำงานของ CU
 - ordinary operation mode
 - generate control signal ควบคุม control unit
 - machine startup mode
 - Conventional Control Unit