Computer Organization

爲什麼電腦不用十進位而用二進位？

二進制只有兩種狀態

對、錯

本土化、非本土化

陰、陽

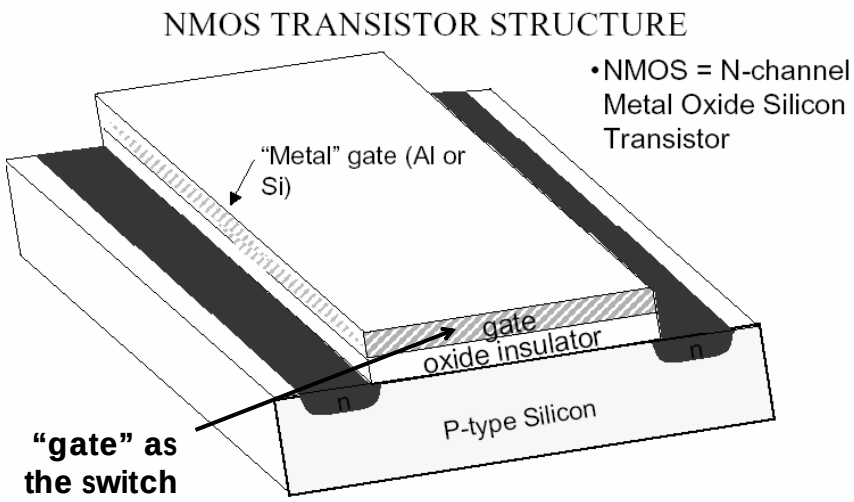
開、關

正、反

真、偽

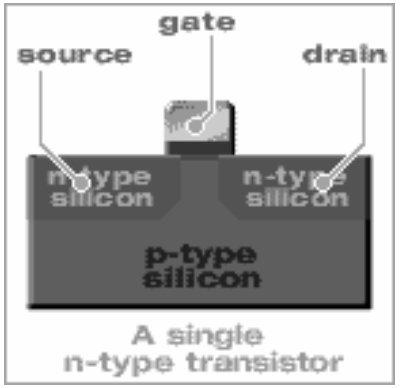
勝、負

有沒有電子開關?



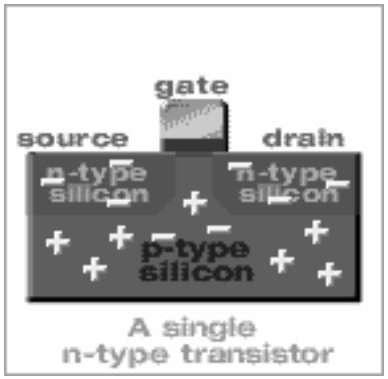
A Working Transistor (1/5)

- Transistors consist of three terminals; the source, the gate, and the drain:



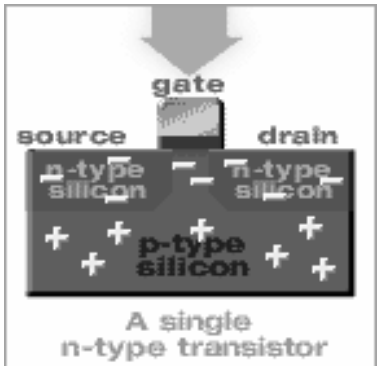
A Working Transistor (2/5)

- In the n-type transistor, both the source and the drain are negatively-charged and sit on a positively-charged well of p-silicon.



A Working Transistor (3/5)

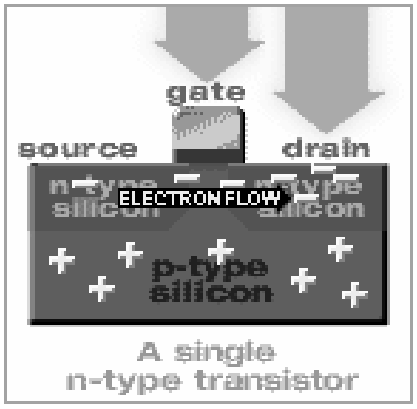
- When positive voltage is applied to the gate, electrons in the p-silicon are attracted to the area under the gate forming an electron channel between the source and the drain.



A Working Transistor (4/5)

- When positive voltage is applied to the drain, the electrons are pulled from the source to the drain. In this state the transistor is on.

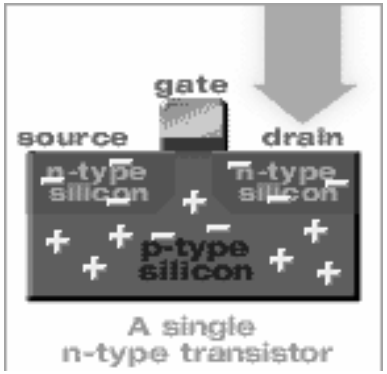
開



A Working Transistor (5/5)

- If the voltage at the gate is removed, electrons are not attracted to the area between the source and drain. The pathway is broken and the transistor is turned off.

關



相關電壓電流特性

及電路分析等知識

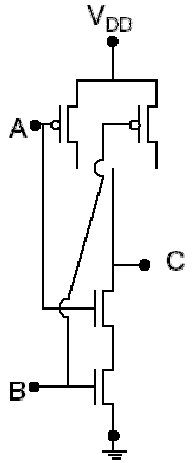
我們是在_____課中介紹的

答：「電子電路學」「超大型積體電路設計」

有了開關就可以做邏輯閘

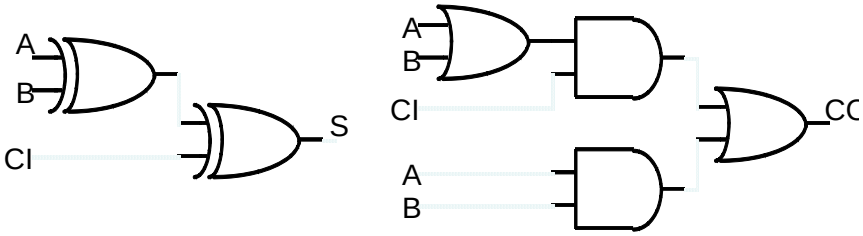
- CMOS NAND:

A	B	A B	C= $\overline{A\,B}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0



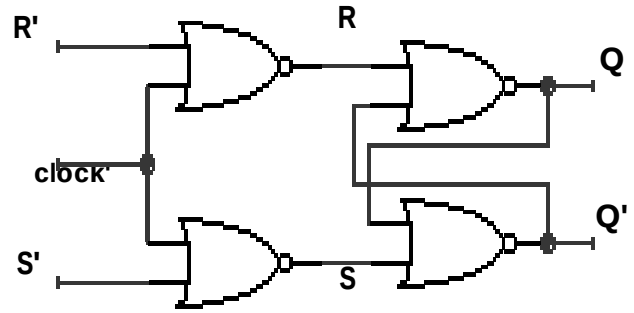
有了邏輯閘就可做邏輯電路

- 加法器：



也可以做記憶元件

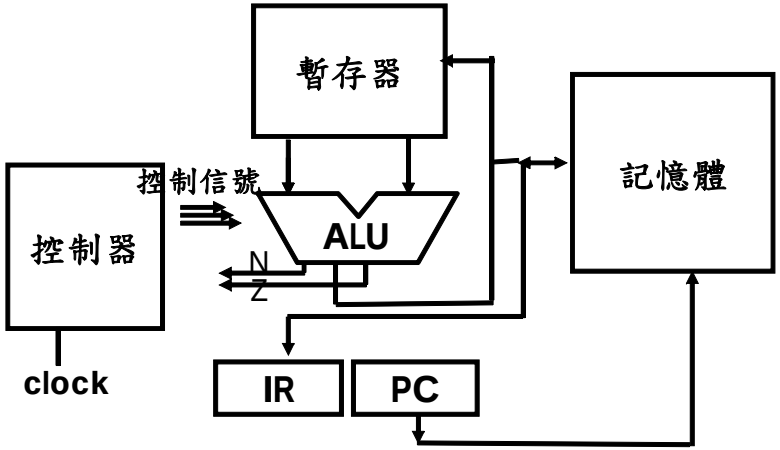
- 可存一個bit的東西：



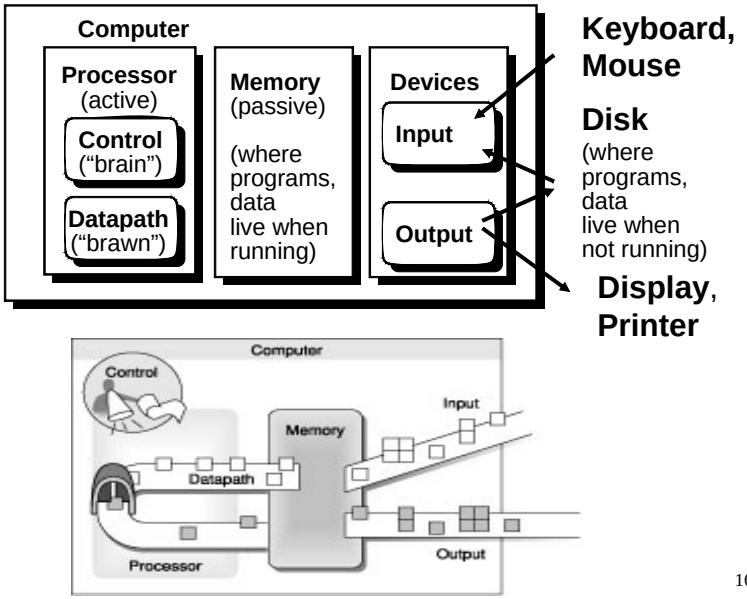
這部份的學問叫_____

答：「數位邏輯設計」

最後，
電腦的主要部份就都可以做了



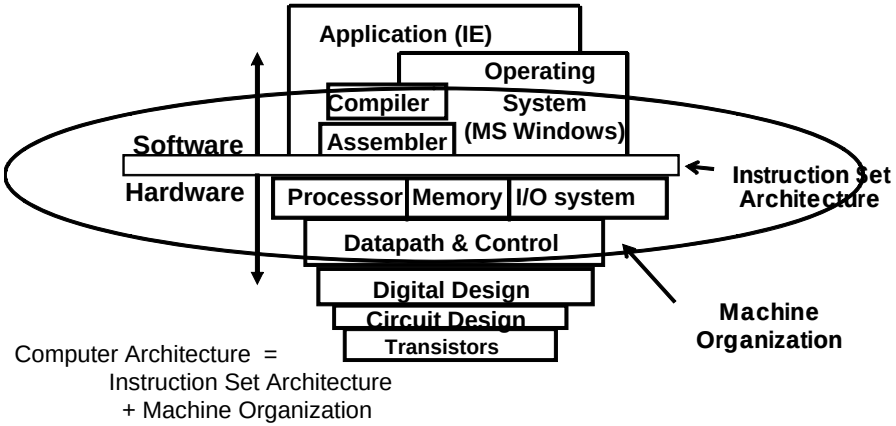
Basic Organization of Any Computer



Computer Organization

- Capabilities and performance characteristics of principal functional units, e.g., registers, ALU, shifters, ...
- Ways in which these components are interconnected (*structure*)
- Information flows between components *(data, datapath*)
- Logic and means by which such information flow is controlled
- *Register Transfer Level* (RTL) description

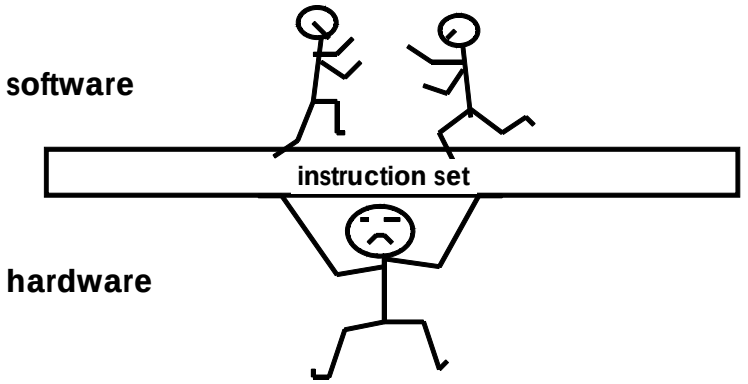
What is Computer Architecture?



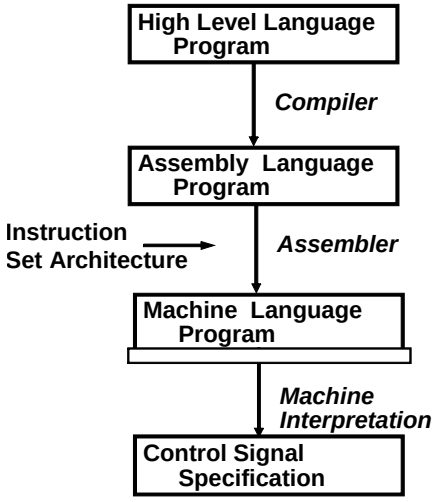
Computer Architecture = Instruction Set Architecture + Machine Organization

- Coordination of many *levels of abstraction*
- Under a rapidly changing set of forces
- Design, Measurement, *and* Evaluation

Instruction Set as a Critical Interface



Another Perspective




```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
lw      $15,  0($2)
lw      $16,  4($2)
sw $16, 0($2)
sw $15, 4($2)
```

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

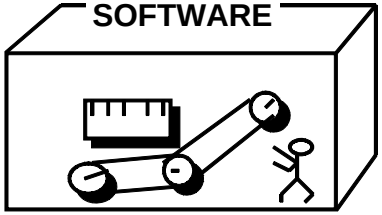
```
ALUOP[0:3] <= InstReg[9:11] & MASK
```

Instruction Set Architecture (ISA)

"... the attributes of a [computing] system as seen by the programmer, *i.e.* the conceptual structure and functional behavior, as distinct from the organization of the data flows and controls, the logic design, and the physical implementation."

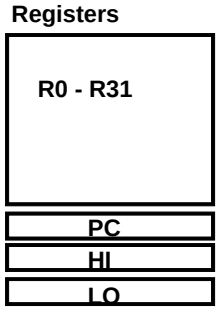
— Amdahl, Blaaw, and Brooks, 1964

- **Organization of Programmable Storage**
- **Data Types and Data Structures: Encodings and Representations**
- **Instruction Set**
- **Instruction Formats**
- **Modes of Addressing and Accessing Data Items and Instructions**
- **Exceptional Conditions**

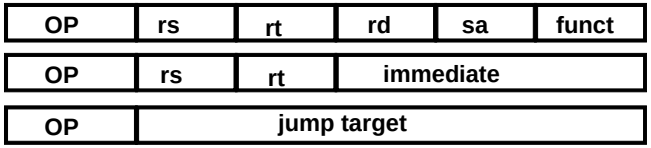


MIPS R3000 ISA

- Instruction categories:
  - Load/Store
  - Computational
  - Jump and Branch
  - Floating Point
    - coprocessor
  - Memory Management
  - Special



3 Instruction Formats: all 32 bits wide



Example ISA

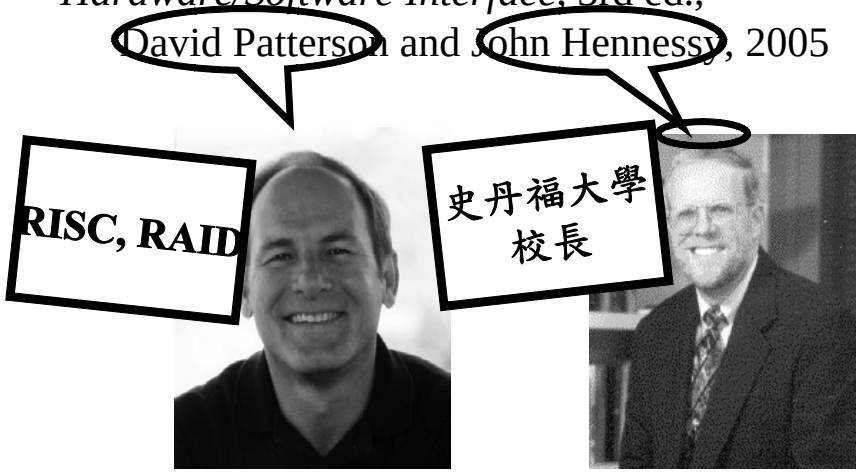
Digital Alpha	(v1, v3)	1992-97
HP PA-RISC	(v1.1, v2.0)	1986-96
Sun Sparc	(v8, v9)	1987-95
SGI MIPS	(MIPS I, II, III, IV, V)	1986-96
Intel	(8086,80286,80386, 80486,Pentium, MMX, SIMD, IA-64, ...)	1978-

Why Do Computer Architecture?

- RAPID CHANGES
- It is exciting!
- It has never been more exciting!
- It impacts every other aspect of electrical engineering and computer science

Text Book

Computer Organization and Design: The Hardware/Software Interface, 3rd ed.,
David Patterson and John Hennessy, 2005



Topics Covered

Computer Organization and Design: The Hardware/Software Interface,3rd ed.,
D. Patterson and J. Hennessy, 2005

Topic	Chapter
Introduction	1
Instructions: Language of the Machine	2
Arithmetic for Computers	3
Assessing and Understanding Performance	4
The Processor: Datapath and Control	5
Enhancing Performance with Pipelining	6
Exploiting Memory Hierarchy	7
Interfacing Processors and Peripherals	8

Prerequisite

- Prerequisite courses:
  - Logic design, assembly language and system programming

Expected Course Workload

- Learn MIPS instruction set
- Learn processor emulators and benchmarking
- One mid-term and one final examination
- Grade breakdown
  - Homework Assignments and Quiz 30%
  - Midterm Exam: 35%
  - Final Exam: 35%