Introduction to Computing

Lectured by: Dr. Pham Tran Vu

t.v.pham@cse.hcmut.edu.vn

http://www.cse.hcmut.edu.vn/~ptvu/i2c

Course Details

- Number of credits: 4
- Study time allocation per week:
  - 4 lecture hours for theory
  - 3 lecture hours for lab work
  - 8 hours for self-study
- Reference:
  - Computing, 3rd ed., Goeffrey Knott & Nick Waites, 2000

Assessment

- Mid-term exam: 30%
- Writing report: 20%
- Presentation: 10%
- Tutorial + Lab work: 10%
- Final exam: 30%

Course Outline

- Fundamental concepts
- Hardware
- Operating systems and Networking
- Databases
- Programming
- Applications and social issues

Lecture 1: Fundamental Concepts

History of computer

Number systems

Data representation

Computer logic

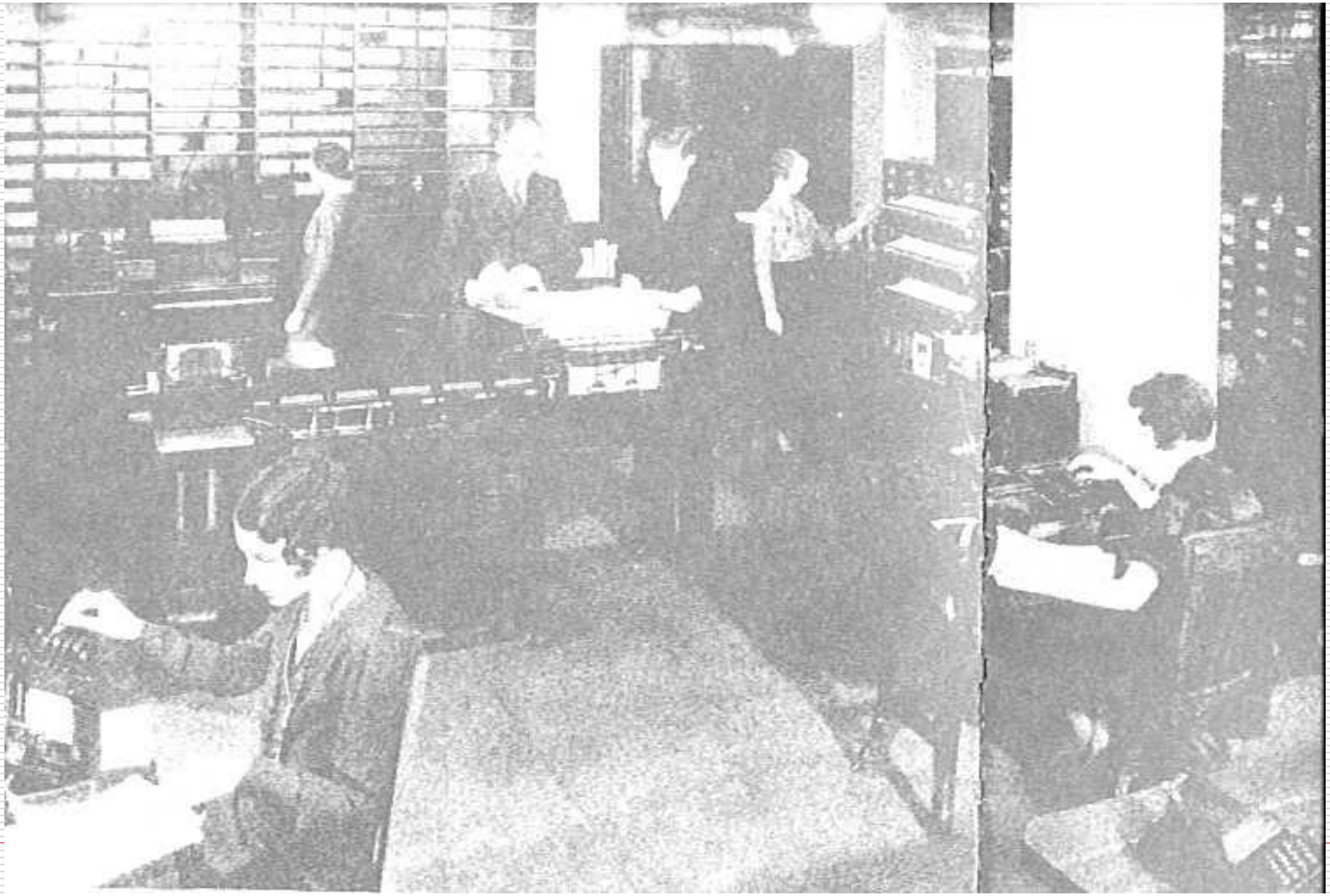
Extra reading: History of computer - http://www.computersciencelab.com/ComputerHistory/History.htm

Computer History

- Computer
  - A job title for people who do calculations
  - A machine for calculation
- Today's computer
  - Digital
  - Programmable

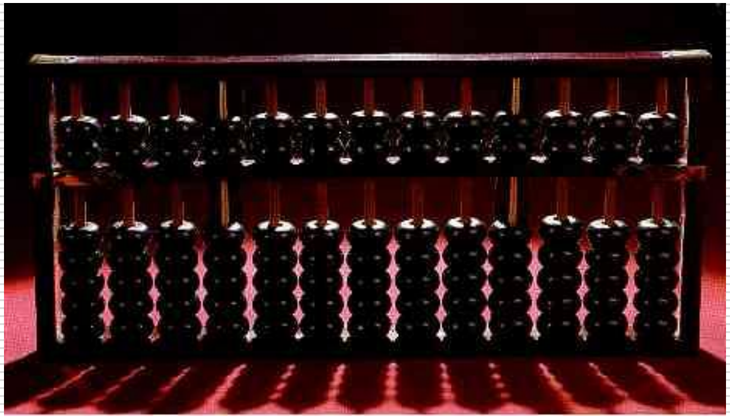
Computer History:

Computers were people



Computer History: Earliest Computers

- Abacus
  - 300 B.C. by the Babylonians



- Astronomical clock
  - By Al-Jazari in 1206
  - First programmable analog computer

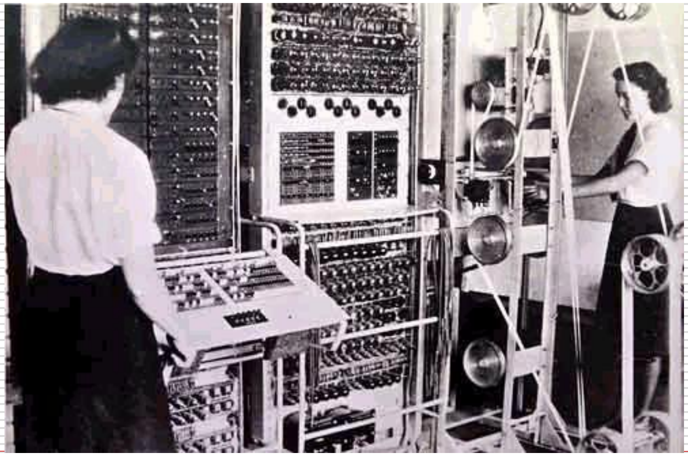
Analogue Computers

- Jacquard's Loom
  - 1801
  - Used punched cards
  - In textile industry

- Cambridge differential analyzer
  - 1938
  - Advanced analog computer

First Digital Computers (1)

- Z3
  - Completed in 1941 in Germany
  - World’s first functional program controlled digital computer
- Colossus
  - Built 1943 in UK
  - First totally electronic computing device

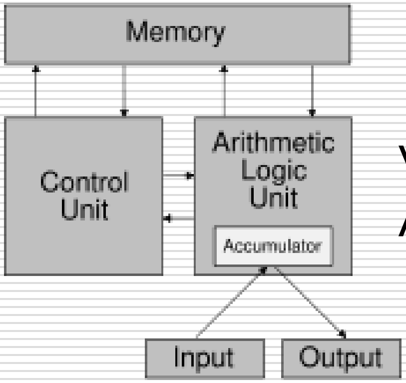


First Digital Computers (2)

- Havard Mark 1
  - Built 1944 by Harvard and IBM
  - First programmable digital computer in US
  - Electro-mechanical computer
  - 5 tons, 500 miles of wire, 8 feet tall, 51 feet long
  - 5 horse power electric motor
  - Run for 15 years

Today's Computers

- Totally digital
- Small in size
- Using Integrated Circuit (IC)
- Based on von Neumann architecture



Von Neumann Architecture

Computer Generations

Blaise Pascal (1642)

Charles Babbage (1830)

Mechanical

(1642 - 1945)

Herman Hollerith founded IBM (International Business Machine) - 1890

Von Neumann (1945)

Vacuum tube

(1945 - 1955)

Transistors

(1955 - 1965)

Intel 8080 (1974)
First integrated circuit processor

IBM 360 (1965)

IC

(1965 - 1980)

80x86 (1978)

?

(1980 - ????)

Digital Computer

- Computer instructions: to tell a computer to do something
- Computer programs: a set of computer instructions
- Machine code: understandable to computers
- Program languages: used to write computer programs

Number Systems (1)

- Base of a number system:
  - The number of different symbols used in the system
  - For examples: denary (decimal) system uses 10 symbols (0,1,2,3,4,5,6,7,8 and 9), hence has the base 10

Number Systems (2)

- Place value:
  - Each symbol has a weight
  - Its value (place value) is decided based on its position with a number
  - For example: in decimal system, each place value is a power of 10 (base)
    - $123_{10} = 1x10^2 + 2x10^1 + 3x10^0$
  - Fraction number:
    - $0.123_{10} = 1x10^{-1} + 2x10^{-2} + 3x10^{-3} = 0.1 + 0.02 + 0.003$

Binary System (1)

- Binary numbers are used in today's digital computers
- Use 2 symbol 0 and 1
- Each digit is know as binary digit or bit
- Base is 2 -> each place value is a power of 2

Place	4	3	2	1	0	-1	-2
Power	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}
Value	16	8	4	2	1	1/2	1/4

Binary System (2)

- Example
  - $11001_2 = 1x2^4 + 1x2^3 + 0x2^2 + 0x\ 2^1 + 1x2^0$

    $= 16 + 8 + 0 + 0 + 1 = 25_{10}$
- Fraction number:
  - $0.0111_2 = 0x2^{-1} + 1x2^{-2} + 1x2^{-3} + 1x\ 2^{-4} = 7/16_{10}$
- Binary numbers from 0 to 9 …
  - 0000->0001->0010->0011->0100->0101-> 0110->0111->1000->1001->…

Binary Arithmetic Operations

- Addition rules:
  - 0 + 0 = 0
  - 0 + 1 = 1
  - 1 + 0 = 1
  - 1 + 1 = 0 carry 1
- Examples
  - 0110 + 0011 = 1001 (6 + 3 = 9)
  - 0110 + 1110 = 10100 (6 + 14 = 20)

Octal and Hexadecimal Numbers (1)

- Binary numbers are used by digital computers but very confusing, especially large numbers
- It is necessary to present binary numbers in a way that is readable by programmers
- Decimal numbers are used naturally by human beings but are not readily converted to or from binary numbers

Octal and Hexadecimal Numbers (2)

- Octal and Hexadecimal numbers are used in preference to decimal numbers, as they are easily converted to and from binary numbers

Octal System

- Octal system has base of 8, using 0, 1, 2, 3, 4, 5, 6, 7 as symbols
- Each place value has the power of eight

Place	4	3	2	1	0	-1	-2
Power	8^4	8^3	8^2	8^1	8^0	8^{-1}	8^{-2}
Value	4096	512	64	8	1	1/8	1/64

Octal Coding

- Octal coding uses three bits at a time ($8=2^3$)

Binary	000	001	010	011	100	101	110	111
Octal	0	1	2	3	4	5	6	7

- To represent a binary number in octal format, a binary number can be split into groups of 3 bits, started from the right hand side
- Then, replace each group by a corresponding octal digit

Octal Coding Example

Binary	01110011	01	110	011
Octal	163	1	6	3
Decimal	115	$1x8^2$	$6x8^1$	$3x8^0$

Hexadecimal System

- Use 16 symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F
- Base 16
- To represent a hexadecimal symbol, a group of 4 bits is needed
- Similar to octal coding, a binary number can be converted to hexadecimal number by splitting the number into groups of 4 bits

Hexadecimal Coding Example

Binary	01110011	0111	0011
Hex	73	7	3
Decimal	115	$7x16^1$	$3x16^0$

- In practice, hexadecimal is used in preference to octal as computer memory is organised into groups of 8 bits, which is a multiple of 4

Number Base Conversions

- Conversions between binary and octal or hex are straight forward
- Conversions from binary, octal or hex to denary have been shown
- Conversions from denary to binary, octal or hex need some calculations

Denary to Binary (1)

- Integers: using successive divisions by the base

Denary	Divided by	Equals	Remainder	Binary	
1273	2	636	1	1	LSB
636	2	318	0	0	
318	2	159	0	0	
159	2	79	1	1	
79	2	39	1	1	
39	2	19	1	1	
19	2	9	1	1	
9	2	4	1	1	
4	2	2	0	0	
2	2	1	0	0	
1	2	0	1	1	MSB

Denary to Binary (2)

- Real numbers:
  - Integer part: using successive divisions by the base
  - Fractional part: using successive multiplications by the base

Denary to Binary (3)

- Example: 34.375_{10} ->100010.011_2
  - Convert the integer part (34) to binary

Denary	Divided by	Equals	Remainder	Binary	
34	2	17	0	0	LSB
17	2	8	1	1	↑
8	2	4	0	0	
4	2	2	0	0	
2	2	1	0	0	
1	2	0	1	1	MSB

Denary to Binary (4)

- Convert 0.375 to binary
  - Using successive multiplications
  - If there is a one (1) before the decimal point, take 1 for binary number
  - If not, take 0 for the binary number
  - Multiply the remainder by the base (2) again

Denary	Multiplied by	Equals	Binary	
0.375	2	0.75	0	MSB
0.75	2	1.5	1	
0.5	2	1	1	LSB

Denary to Binary (5)

- There is possible loss of precision when converting a decimal number into binary, when the factional part of a real number cannot be precisely converted to binary equivalent
- For example, when converting 0.425 into a binary number

Denary to Binary (6)

Denary	Multiplied by	Equals	Binary	
0.435	2	0.85	0	MSB
0.85	2	1.7	1	
0.7	2	1.4	1	
0.4	2	0.8	0	
0.8	2	1.6	1	
0.6	2	1.2	1	
0.2	2	0.4	0	
0.4	2	0.8	0	
0.8	2	1.6	1	
0.6	2	1.2	1	
0.2	2	0.4	0	Etc

Denary to Octal and Hexadecimal

- The same method can be applied to convert denary numbers to octal and hexadecimal
- For example, convert 1273_{10} to 2371_8

Denary	Divided by	Equals	Remainder	Octal	
1273	8	159	1	1	LSB
159	8	19	7	7	
19	8	2	3	3	
2	8	0	2	2	MSB