COMP2611: Computer Organization

Tutorial 1: Programs and Numbers

Overview

- You will learn the following in this tutorial:
  - the compilation process of computer programs into machine instructions.
  - the conversion between binary, decimal and hexadecimal numbers.
  - the computer numerical unit prefix.

Programs and Numbers

Computer programs

- the compilation process

Number bases

- the introduction of different bases

Conversion between binary and decimal

- conversion methods and exercises

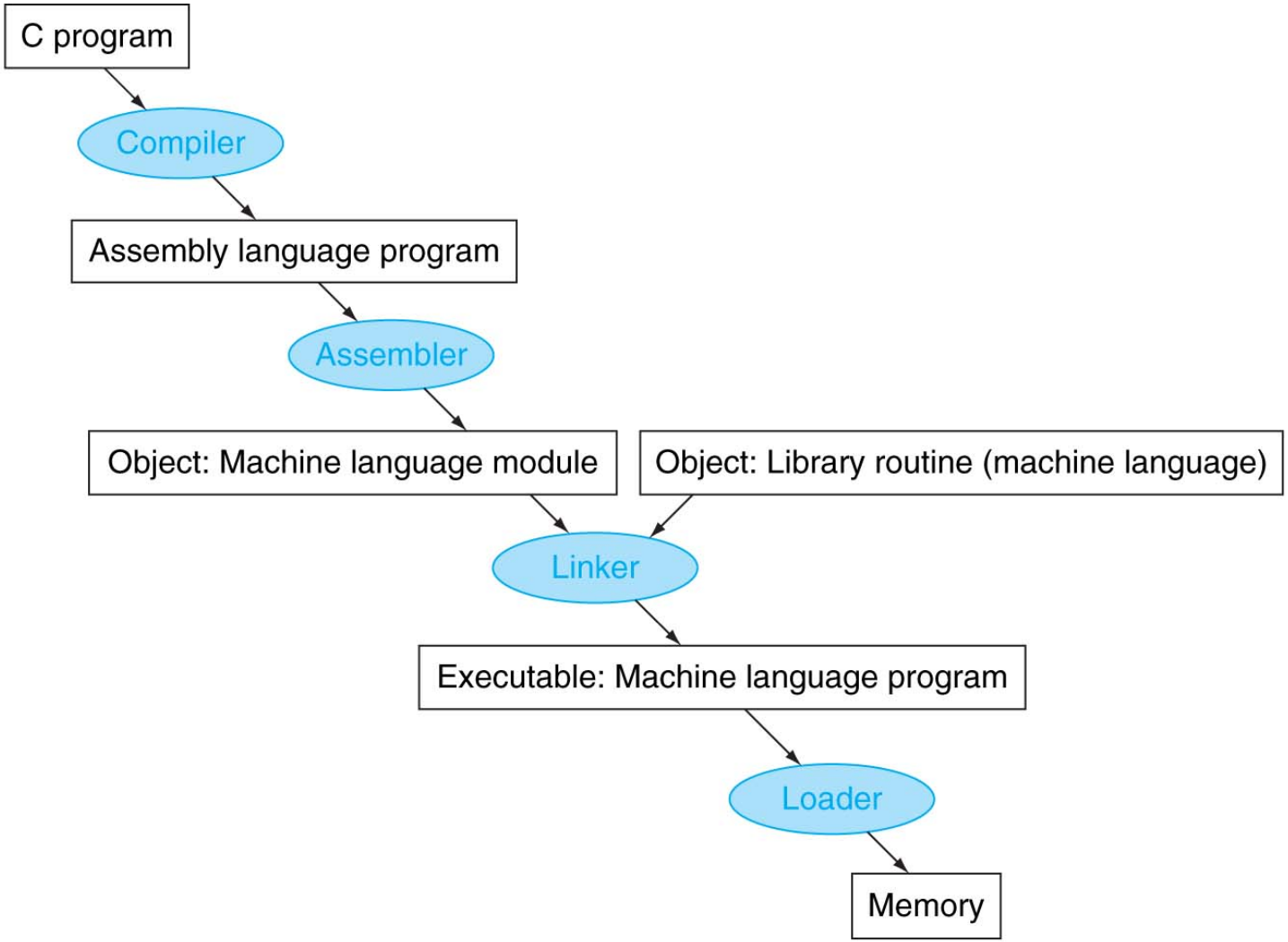
Conversion between binary and hexadecimal

- conversion methods and exercises

Computer numerical unit prefix

Exercises

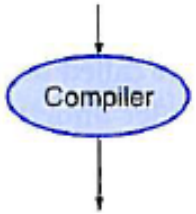
Computer programs



Programs for MIPS processor

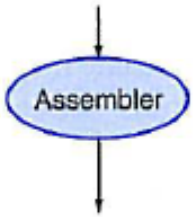
High-level language program (in C)

```
swap(int v[], int k)
{int temp;
   temp = v[k];
   v[k] = v[k+1];
   v[k+1] = temp;
}
```



Assembly language program (for MIPS)

```
swap:
      muli $2, $5,4
      add  $2, $4,$2
      lw   $15, 0($2)
      lw   $16, 4($2)
      sw   $16, 0($2)
      sw   $15, 4($2)
      jr   $31
```



Binary machine language program (for MIPS)

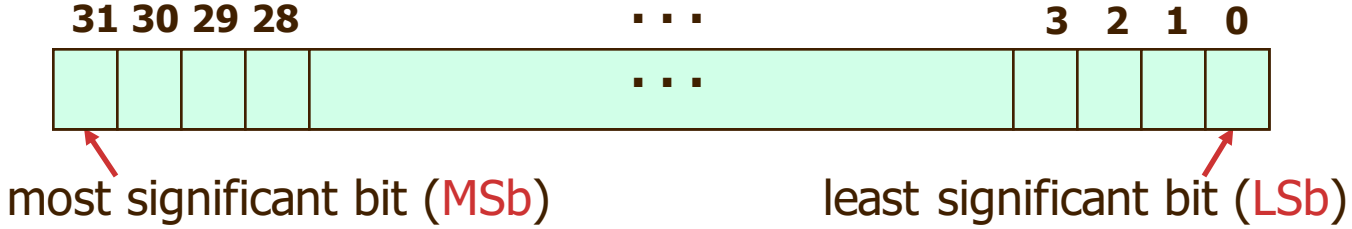
```
00000000101000010000000000011000
00000000000011000000110000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

Programs and Numbers

- Computer programs
  - the compilation process
- Number bases
  - the introduction of different bases
- Conversion between binary and decimal
  - conversion methods and exercises
- Conversion between binary and hexadecimal
  - conversion methods and exercises
- Computer numerical unit prefix
- Exercises

Base, Representation and Value

- Numbers can be represented in any base
  - Human: decimal (base 10, has 10 digits 0,1,…,9);
  - Computer: binary (base 2, has 2 digits, 0,1)
- Positional Notation: value of the `i`th digit `d` is `d x Base`^{i}
  - $1001_2 = (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)_{10} = 9_{10}$
- Bits are grouped and numbered 0, 1, 2, 3 … from right to the left:
  - Byte: a group of 8 bits
  - Word: a group of 32 or 64 bit



- Value of the 32-bit binary numbers =

$$(b_{31} \times 2^{31}) + (b_{30} \times 2^{30}) + \ldots + (b_1 \times 2^1) + (b_0 \times 2^0)$$

Hexadecimal Representation

- Hexadecimal (base 16) numbers are commonly used
- To avoid reading and writing long binary numbers

Conversion to hexadecimal

- Since base 16 is a power of 2, we can simply convert by replacing each group of four bits by a single hexadecimal digit, and vice versa

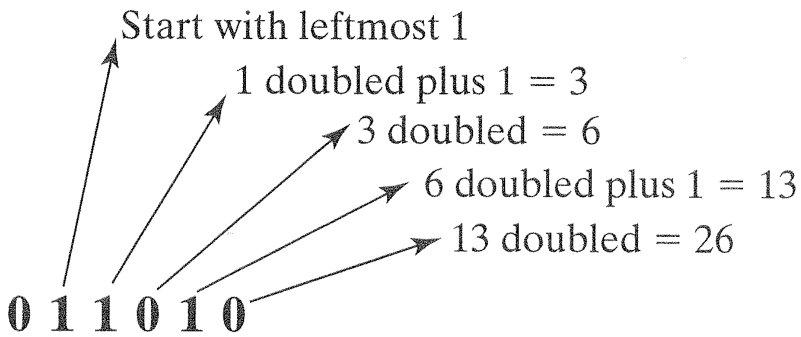
Example of hexadecimal-to-binary conversion:

- 0_{hex} - 9_{hex} for 0000_2 - 1001_2
- a_{hex} - f_{hex} for 1010_2 – 1111_2
- i.e. $0000\ 1010\ 0000\ 0101\ 0000\ 1100\ 0000\ 0110_2$
  = 0 a 0 5 0 c 0 6_{hex}
  = 0x0a050c06 # 0x to indicate it is a hexadecimal
  = 168102918_{10}

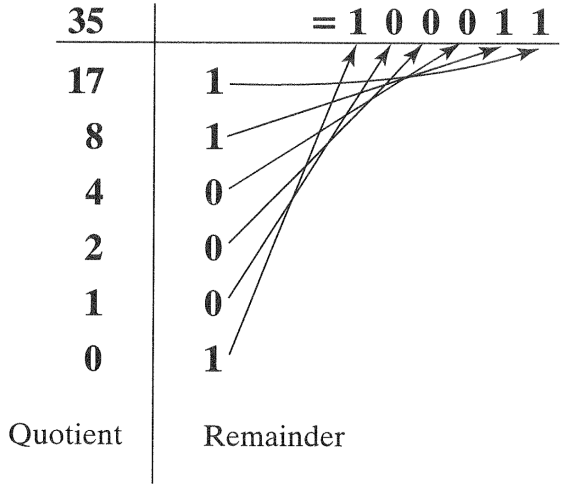
Simple recipes: Converting Binary <-> Decimal

- Binary to decimal:
  - Double and add method:
    - For each position *i* starting at the leftmost 1,
      - Double the sum, add bit *i* to the sum

    Ex: 011010



- Decimal to binary:
  - Successive divisions by 2:
    - Inverse of double and add method
    - In each step the remainder of the division is the next bit of the sequence
- Remark:
  - useful powers of 2 to memorize



2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}
1	2	4	8	16	32	64	128	256	512	1024

Programs and Numbers

- Computer programs
  - the compilation process
- Number bases
  - the introduction of different bases
- Conversion between binary and decimal
  - conversion methods and exercises
- Conversion between binary and hexadecimal
  - conversion methods and exercises
- Computer numerical unit prefix
- Exercises

Converting positive decimal integer to binary

- We keep on dividing the decimal integer by 2 until the quotient is 0. The remainder at each step corresponds to a digit of the integer in base 2, from the Least Significant Digit (LSD) to the Most Significant Digit (MSD).

Question 1: Convert $37_{(10)}$ to the binary format.

Converting positive binary integer to decimal

- The value represented by the i-th bit d of a positive binary integer is in fact $d \times 2^i$. Note that the Least Significant bit is the 0-th bit.
- Take the integer $ABCD_{(2)}$ as an example, it effectively corresponds to:

  $ABCD_{(2)} = (Ax2^3) + (Bx2^2) + (Cx2^1) + (Dx2^0)$

2's power	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}
value	2	4	8	16	32	64	128	256	512	1024

Exercises

Question 1: Convert the positive integer 10 $1001_{(2)}$ to the decimal format.

Programs and Numbers

- Computer programs
  - the compilation process
- Number bases
  - the introduction of different bases
- Conversion between binary and decimal
  - conversion methods and exercises
- Conversion between binary and hexadecimal
  - conversion methods and exercises
- Computer numerical unit prefix
- Exercises

Hexadecimal format

- The binary number system is verbose in that even small numerical values could require long strings of bits to represent.
- Hexadecimal number system is a number system that has a base of 16 (instead of 2).
- Under the hexadecimal system, there are 16 possible values for each digit, as shown in the table.

Decimal value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexa-decimal digit	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Converting binary number to hexadecimal

- We just group the binary number into groups of 4 bits, and then each group represents one digit of the corresponding hexadecimal number.
- The conversion can be made immediately by eye inspection.

Exercises

Question 1: Convert $0011\ 0010_{(2)}$ to the hexadecimal format.

Converting hexadecimal number to binary

- We just expand each digit of the hexadecimal number into 4 bits, and then the resulting bits from all the digits represent the corresponding binary number.
- Again, the conversion can be made immediately by eye inspection.

Question 1: Convert $A7_{(16)}$ to the binary format.

Programs and Numbers

Computer programs
- the compilation process

Number bases
- the introduction of different bases

Conversion between binary and decimal
- conversion methods and exercises

Conversion between binary and hexadecimal
- conversion methods and exercises

Computer numerical unit prefix

Exercises

Computer numerical unit prefix

- For representing a number of bits or bytes in computers, the unit prefix Kilo is often used to represent 2^{10} (equal to 1024) not 1000.
- For examples, 1 kilobytes = 1024 bytes and 1 kilobits = 1024 bits.
- Similarly, we have the following table for the common prefixes used for bytes and bits:

Unit prefix	Value
Kilo	2^{10} (or 1024)
Mega	2^{20} (or 2^{10} Kilo)
Giga	2^{30} (or 2^{10} Mega)
Tera	2^{40} (or 2^{10} Giga)

Programs and Numbers

- Computer programs
  - the compilation process
- Number bases
  - the introduction of different bases
- Conversion between binary and decimal
  - conversion methods and exercises
- Conversion between binary and hexadecimal
  - conversion methods and exercises
- Computer numerical unit prefix
- Exercises

Exercises

Question 1: Convert $176_{(10)}$ to the binary format.

Exercises

Question 2: Convert the positive integer $11\ 0100\ 1001_{(2)}$ to the decimal format.

Exercises

Question 3: Convert $1010\ 0011\ 1001\ 0111\ 0100_{(2)}$ to the hexadecimal format.

Exercises

Question 4: Convert $B12A3F01_{(16)}$ to the binary format.

Conclusion

- You have learnt:
  - the compilation process of computer programs into machine instructions.
  - the conversion between binary, decimal and hexadecimal numbers.
  - the computer numerical unit prefix.