**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

HUỲNH VĂN TÚ -

BÙI THỊ CẨM NHUNG – 171264

NGUYỄN NGỌC BĂNG TÂM – 1712747

**BIỂU DIỄN VÀ TÍNH TOÁN SỐ HỌC**

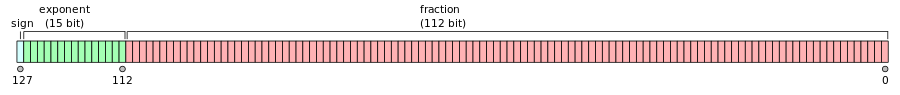
**TRÊN MÁY TÍNH**

**MÔN: KIẾN TRÚC MÁY TÍNH VÀ HỢP NGỮ**

**TP. HCM, 2019**

# **Chương 1. MỞ ĐẦU**

* 1. **Mô tả đồ án**
* Thiết kế kiểu dữ liệu QInt - số nguyên lớn có dấu với độ lớn 16 bytes, tức 128 bits.
* Thiết kế kiểu dữ liệu QFloat - số chấm động có độ chính xác cao với độ lớn 128 bits có cấu trúc như sau:

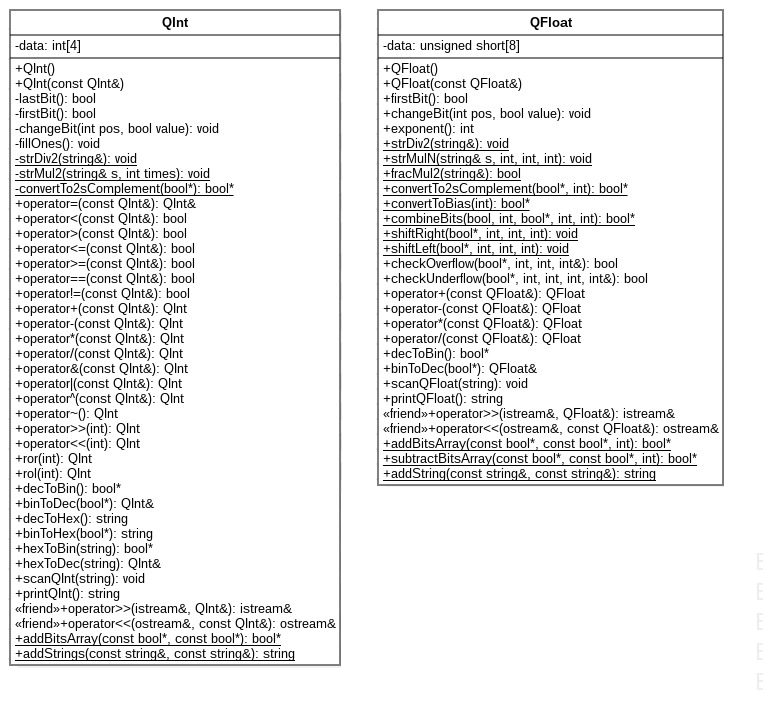


* 1. **Đánh giá mức độ hoàn thành**

Nhìn chung, nhóm đã hoàn thành 100% chức năng được yêu cầu. Cụ thể:

|  |  |  |
| --- | --- | --- |
| **Chức năng** | **QInt** | **QFloat** |
| Hàm nhập | x | x |
| Hàm xuất | x | x |
| Hàm chuyển đổi nhị phân sang thập phân | x | x |
| Hàm chuyển đổi thập phân sang nhị phân | x | x |
| Hàm chuyển đổi nhị phân sang thập lục phân | x |  |
| Hàm chuyển đổi thập phân sang thập lục phân | x |  |
| Các toán tử: “+”, “-” , “\*”, “/” | x | x |
| Các toán tử so sánh và phép gán: “<”, “>”, “==”, “<=”, “>=”, “=” | x |  |
| Các toán tử: AND “&”, OR “|”, XOR “^”, NOT “~” | x |  |
| Các toán tử: dịch trái “<<”, dịch phải “>>”, xoay trái “rol”, xoay phải “ror” | x |  |
| Chương trình minh họa (console + tham số dòng lệnh) | x | x |

1. **NỘI DUNG ĐỒ ÁN**
   1. **Sơ đồ UML**

****Sơ đồ UML thể hiện thiết kế của lớp QInt (128-bit) và QFloat (128-bit).

Hình 1. Sơ đồ UML của lớp QInt và QFloat

* 1. **Phạm vi biểu diễn**
     1. **Phạm vi biểu diễn của kiểu dữ liệu QInt (128-bit):**

|  |  |  |
| --- | --- | --- |
| **Số** | **Biểu diễn nhị phân** | **Giá trị** |
| Chuẩn max | 011111111…1111111 | 2127 - 1 |
| Chuẩn min | 100000000…0000000 | -2127 |

* + 1. **Phạm vi biểu diễn của kiểu dữ liệu QFloat (128-bit):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Số** | **Dấu** | **Mũ** | **Trị** | **Giá trị** |
| +0 | 0 | 0000…0000 | 0000…0000 | +0.0 |
| -0 | 1 | 0000…0000 | 0000…0000 | -0.0 |
| +INF | 0 | 1111…1111 | 0000…0000 | +INF |
| -INF | 1 | 1111…1111 | 0000…0000 | -INF |
| NAN | 0/1 | 1111…1111 | Dãy bit khác 0 | NAN |
| Chuẩn max (+) | 0 | 1111…1110 | 1111…1111 | (2 - 2-112) x 216383 |
| Chuẩn min (+) | 0 | 0000…0001 | 0000…0000 | 2-16382 |
| Chuẩn max (-) | 1 |  |  |  |
| Chuẩn min (-) | 1 |  |  |  |
| Không chuẩn max (+) | 0 | 0000…0000 | 1111…1111 | (1 - 2-112) x 2-16382 |
| Không chuẩn min (+) | 0 | 0000…0000 | 0000…0001 | 2-16494 |
| Không chuẩn max (-) | 1 |  |  |  |
| Không chuẩn min (-) |  |  |  |  |

QFloat (128 bit) ~ 1 + 15 + 112

- Số 0: 0 000…00 000…00 hoặc 1 000..00 000..00

- Số dạng chuẩn:

+ Số dương (+)

MAX: 0 111…10 111…11

= 1.111…11 x 2^16383

= ( 2^0 + 2^(-1) + 2^(-2) + … + 2^(-112) ) x 2^16383

= ( 2 - 2^(-112) ) x 2^16383

MIN: 0 000…01 000…00

= 2^(-16382)

+ Tương tự cho số âm.

Vậy: Phạm vi biểu diễn số chấm động lớn (128 bit) có dạng chuẩn là:

( - ( 2 - 2^(-112) ) x 2^16383, -2^(-16382)) U (2^(-16382), ( 2 - 2^(-112) ) x 2^16383)

- Số dạng không chuẩn:

+ Số dương (+)

MAX: 0 000…00 111…11

= 0.111..11 x 2^(-16382)

= ( 2^(-1) + 2^(-2) + 2^(-3) + … + 2^(-112) ) x 2^(-16382)

= ( 1 - 2^(-112) ) x 2^(-16382)

MIN: 0 000…00 000…01

= 0.000…01 x 2^(-16382)

= 2^(-16494)

+ Tương tự cho số âm.

Vậy: Phạm vi biểu diễn số chấm động lớn (128 bit) có dạng không chuẩn là:

( - ( 1 - 2^(-112) ) x 2^(-16382), -2^(-16494)) U (2^(-16494), ( 1 - 2^(-112) ) x 2^(-16382))

1. **THỬ NGHIỆM**
   1. **Thử nghiệm trên kiểu dữ liệu Qint (128-bit):**
   2. **Thử nghiệm trên kiểu dữ liệu QFloat (128-bit):**
2. **TÀI LIỆU THAM KHẢO**
   * + 1. Wikipedia, Floating-point arithmetic