**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN 01 MÔN KIẾN TRÚC MÁY TÍNH VÀ HỢP NGỮ**

Đề tài:

**BIỂU DIỄN VÀ TÍNH TOÁN SỐ HỌC TRONG MÁY TÍNH**

Người thực hiện:

Nguyễn Hoàng Thái Dương (18120336)

Trần Thanh Quang (18120230)

Trương Trọng Lộc (18120197)

Giảng viên hướng dẫn:

Thầy ThS. Lê Viết Long

Thành phố Hồ Chí Minh – Tháng 03, 2020

MỤC LỤC

[MỤC LỤC 2](#_Toc40444798)

[1. THÀNH VIÊN VÀ PHÂN CHIA CÔNG VIỆC: 4](#_Toc40444799)

[2. BIỂU DIỄN SỐ NGUYÊN LỚN 128 BIT 4](#_Toc40444800)

[2.1. Thiết kế cấu trúc dữ liệu QInt: 4](#_Toc40444801)

[2.2. Chuyển đổi giữa các hệ cơ số: 4](#_Toc40444802)

[2.2.1. Ý tưởng chuyển đổi: 4](#_Toc40444803)

[2.2.2. Nhóm hàm SetBit, GetBit: 4](#_Toc40444804)

[2.2.3. Chuyển đổi từ chuỗi nhị phân (string) sang QInt 5](#_Toc40444805)

[2.2.4. Chuyển đổi từ chuỗi thập phân (string) sang QInt 5](#_Toc40444806)

[2.2.5. Chuyển đổi từ chuỗi thập lục phân (string) sang QInt 6](#_Toc40444807)

[2.2.6. Chuyển đổi từ chuỗi ở hệ cơ số 2, 10, 16 (string) sang QInt 6](#_Toc40444808)

[2.2.7. Chuyển đổi từ QInt sang chuỗi nhị phân (string) 6](#_Toc40444809)

[2.2.8. Chuyển đổi từ QInt sang chuỗi thập phân (string) 6](#_Toc40444810)

[2.2.9. Chuyển đổi từ QInt sang chuỗi thập lục phân (string) 7](#_Toc40444811)

[2.2.10. Chuyển đổi từ QInt sang chuỗi ở hệ cơ số 2, 10, 16 (string) 7](#_Toc40444812)

[2.2.11. Chuyển đổi từ chuỗi ở hệ cơ số a sang chuỗi ở hệ cơ số b (a, b = 2, 10, 16) 7](#_Toc40444813)

[2.3. Các nhóm toán tử: 7](#_Toc40444814)

[2.3.1. Toán tử cộng: 7](#_Toc40444815)

[2.3.2. Toán tử trừ: 7](#_Toc40444816)

[2.3.3. Toán tử nhân: 7](#_Toc40444817)

[2.3.4. Toán tử chia: 7](#_Toc40444818)

[2.3.5. Toán tử so sánh và gán: 7](#_Toc40444819)

[2.3.6. Toán tử AND, OR, XOR, NOT: 7](#_Toc40444820)

[2.3.7. Toán tử dịch trái, dịch phải: 7](#_Toc40444821)

[2.3.8. Toán tử xoay trái, xoay phải: 7](#_Toc40444822)

# THÀNH VIÊN VÀ PHÂN CHIA CÔNG VIỆC:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ và tên** | **Công việc** |
| 1 | 18120336 | Nguyễn Hoàng Thái Dương |  |
| 2 | 18120230 | Trần Thanh Quang |  |
| 3 | 18120197 | Trương Trọng Lộc |  |

# BIỂU DIỄN SỐ NGUYÊN LỚN 128 BIT

## Thiết kế cấu trúc dữ liệu QInt:

* Số nguyên lớn QInt bao gồm 128 bit được lưu trữ bởi mảng *data* gồm 4 phần tử kiểu số nguyên. Ở phương thức khởi tạo mặc định, tất cả 128 bit này đều là bit 0.
* Vì 1 phần tử kiểu số nguyên có kích thước 4 bytes nên ta sử dụng 4 phần tử để biểu diễn (thỏa yêu cầu 16 bytes như đề bài).

## Chuyển đổi giữa các hệ cơ số:

### Ý tưởng chuyển đổi:

Đầu vào sẽ là một chuỗi String ở dạng nhị phân / thập lục phân / thập phân. Sau đó ta lưu chuỗi String này vào QInt bằng các hàm chuyển đổi. Từ QInt ta chuyển sang lại dạng chuỗi ở các dạng nhị phân / thập lục phân / thập phân. Như vậy, ta có thể chuyển từ chuỗi String từ hệ cơ số a sang chuỗi String ở hệ cơ số b dựa vào trung gian là QInt.

### Nhóm hàm SetBit, GetBit:

* Lưu ý về việc lưu các dãy bit. Ta xét ví dụ sau: Giả sử ta có dãy 8 bit. Khi đó, đối với dạng chuỗi (string), vị trí từng bit sẽ được đánh số thứ tự từ trái sang phải. Tuy nhiên, khi lưu trữ dãy bit, ta đánh số từ phải sang trái. Giả sử ta lưu dãy 8 bit này vào 1 mảng gồm 2 phần tử có kích thước 4 bit, ta lưu như sau:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Vị trí bit (dạng string) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Vị trí bit (khi lưu trữ) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Mảng 2 phần tử (mỗi phần tử 4 bit) | A[0] | | | | A[1] | | | |
| Vị trí bit ở mỗi phần tử trong mảng | 3 | 2 | 1 | 0 | 3 | 2 | 1 | 0 |

* Để dễ hiểu, ta xét ví dụ, dãy 8 bit 1100 1011 ở dạng chuỗi (string), khi đó phần tử str[0] = ‘1’ sẽ tương ứng với vị trí bit thứ 7 khi lưu trữ. Khi đó bit này sẽ được lưu trữ vào A[1] ở vị trí thứ 3. Sở dĩ là A[1] vì bit lưu trữ ở vị trí thứ 7 chia cho kích thước một phần tử là 4, lấy phần nguyên ta được 1; ở vị trí thứ 3 trong A[1] thì 7 chia 4 lấy phần dư là 3. Từ đó, ta thấy với dãy 1100 1011 thì A[0] gồm dãy 4 bit 1011 và A[1] gồm dãy 4 bit 1100.
* Tổng quát để xác định vị trí bit lưu ở đâu trong QInt dựa vào vị trí bit khi lưu trữ, ta làm như sau:
  + Xác định *block* dựa vào vị trí bit khi lưu trữ bằng cách lấy vị trí đó chia cho kích thước của một *block*. Với QInt, mỗi *block* là kiểu số nguyên nên kích thước là 32 bit.
  + Xác định vị trí bit trong *block* bằng cách lấy vị trí bit khi lưu trữ chia lấy dư cho kích thước của một *block.*
* Sau khi xác định được vị trí bit trong QInt, ta tiến hành GetBit hoặc SetBit như sau:
  + Với GetBit, ta thực hiện phép AND với 1 tại bit đó.
  + Với SetBit, nếu là bit 1, ta thực hiện phép OR với 1 tại bit đó (vì OR với 1 sẽ ra 1); nếu là bit 0, ta thực hiện phép AND với 0 tại bit đó (vì AND với 0 sẽ ra 0)

### Chuyển đổi từ chuỗi nhị phân (string) sang QInt

Ta lần lượt duyệt qua từng phần tử của chuỗi string. Cứ mỗi lần duyệt ta gọi hàm SetBit để lưu vào QInt. Ở đây, ta duyệt từ đầu đến cuối của chuỗi string, nên khi SetBit vị trí sẽ là str.length() – 1 – i, với i là biến đếm duyệt chuỗi string, str.length() là độ dài chuỗi string (vì vị trí set bit là vị trị lưu trữ - sẽ được đánh số thứ tự ngược với chuỗi string như đã trình bày ở trên).

### Chuyển đổi từ chuỗi thập phân (string) sang QInt

* Đầu tiên ta xét chuỗi thập phân biểu thị số âm hay dương, nếu có dấu ‘-’ ta thực hiện xóa dấu này ra khỏi chuỗi và có lưu lại sự thay đổi này bởi biến *sign* kiểu bool để làm cơ sở xét dấu lúc sau.
* Ta thực hiện chuyển chuỗi thập phân sang QInt bằng cách chia chuỗi này cho 2, phần dư sẽ được SetBit vào QInt, còn phần nguyên ta tiếp tục chia cho 2. Lặp lại như vậy cho đến khi chuỗi bằng “0” hoặc chuỗi rỗng.
* Ta dựa vào tình trạng biến *sign*, nếu biến *sign* biểu thị số âm ta thực hiện chuyển sang dạng bù 2 (ta thực hiện đổi bit kết quả tìm được ở trên, sau đó cộng cho 1).

### Chuyển đổi từ chuỗi thập lục phân (string) sang QInt

* Do ở hệ thập lục phân, biểu diễn bởi 16 ký tự là 0 đến 9 và A đến F. Nên trước tiên ta thực hiện chuẩn hóa các ký tự đầu vào thành kí tự in hoa (nếu đầu vào là chữ thường).
* Điểm đặc biệt ở hệ thập lục phân là mỗi ký tự biểu diễn của nó sẽ được biễu diễn bởi 4 bit nhị phân (Vì mỗi ký tự của nó chạy từ 0 đến 15). Vì vậy ta sẽ duyệt lần lượt từng ký tự ở hệ thập lục phân và chuyển nó về dạng 4 bit nhị phân.
* Lưu ý khi chuyển, nếu gặp ký tự chữ ta phải chuyển đó về dạng số tương ứng rồi sau đó mới chuyển về dạng 4 bit. Cụ thể A là 10, B là 11, C là 12, D là 13, E là 14 và F là 15.

### Chuyển đổi từ chuỗi ở hệ cơ số 2, 10, 16 (string) sang QInt

Ta sử dụng câu lệnh *if* trên tham số đầu vào (đầu vào sẽ được truyền vào biểu thị hệ cơ số cần được chuyển đổi vào QInt). Sau đó dựa vào đầu vào ta lần lượt gọi các hàm đã trình bày ở [2.2.3](#_Chuyển_đổi_từ_5), [2.2.4](#_Chuyển_đổi_từ_6) và [2.2.5](#_Chuyển_đổi_từ_7) để lưu trữ vào QInt.

### Chuyển đổi từ QInt sang chuỗi nhị phân (string)

Ta thực hiện lấy từng bit trong QInt và cộng vào chuỗi kết quả. Sau đó ta loại bỏ các số 0 thừa ở đầu. Lưu ý giữ lại 1 bit 0 nếu 128 bit đều là 0.

### Chuyển đổi từ QInt sang chuỗi thập phân (string)

* Ta thực hiện lấy từng bit trong QInt theo đúng thứ tự lưu trữ (từ 0 đến 126, trừ bit dấu). Tại mỗi bit nếu giá trị bit là 1 ta thực hiện cộng với 2^i với i là số thứ tự của bit.
* Ta xét bit dấu, nếu bit dấu là 1, ta thực hiện phép trừ giữa 2^127 và kết quả tìm được ở trên, sau đó thêm ‘-’ vào trước chuỗi kết quả.

### Chuyển đổi từ QInt sang chuỗi thập lục phân (string)

* Đầu tiên, ta tạo kiểu dữ liệu *map* để lưu trữ các ký tự biểu diễn hệ thập lục phân ở dạng 4 bit. Sở dĩ chọn kiểu dữ liệu *map* vì nó dễ dàng truy xuất phần tử với index dạng chuỗi.
* Sau đó ta duyệt lần lượt bộ 4 bit trong QInt từ vị trí i = 0 và dựa vào chuỗi 4 bit đó ta cộng vào chuỗi kết quả ký tự ở hệ thập lục phân tương ứng thông qua kiểu dữ liệu *map*.
* Sau đó ta thực hiện xóa các số 0 thừa (nếu có) và thực hiện đảo ngược lại kết quả (vì i = 0 nên ta phải đảo ngược kết quả).

### Chuyển đổi từ QInt sang chuỗi ở hệ cơ số 2, 10, 16 (string)

Ta sử dụng câu lệnh *if* trên tham số đầu vào (đầu vào sẽ được truyền vào biểu thị hệ cơ số cần được chuyển đổi ra string từ QInt). Sau đó dựa vào đầu vào ta lần lượt gọi các hàm đã trình bày ở [2.2.7](#_Chuyển_đổi_từ_2), [2.2.8](#_Chuyển_đổi_từ_3) và [2.2.9](#_Chuyển_đổi_từ_4) để chuyển từ QInt sang string.

### Chuyển đổi từ chuỗi ở hệ cơ số a sang chuỗi ở hệ cơ số b (a, b = 2, 10, 16)

Ta sử dụng câu lệnh *if* trên tham số đầu vào (đầu vào sẽ được truyền vào biểu thị hệ cơ số a – hệ cơ số cần phải chuyển đổi và hệ cơ số b – sau khi chuyển đổi). Sau đó dựa vào đầu vào ta lần lượt gọi các hàm đã trình bày ở [2.2.6](#_Chuyển_đổi_từ) và [2.2.10](#_Chuyển_đổi_từ_1) để thực hiện việc chuyển đổi.

## Các nhóm toán tử:

### Toán tử cộng:

### Toán tử trừ:

### Toán tử nhân:

### Toán tử chia và lấy dư:

### Toán tử so sánh và gán:

### Toán tử AND, OR, XOR, NOT:

### Toán tử dịch trái, dịch phải:

### Toán tử xoay trái, xoay phải: