

KHOA CÔNG NGHỆ THÔNG TIN.

MÔN HỌC: KIẾN TRÚC MÁY TÍNH.

Họ tên SV: TỪ VĂN TÂM

MSSV: 2311556012

LAB 2: BIỂU DIỄN SỐ HỌC TRÊN MÁY TÍNH

(SV thực hiện tại lớp)

Bài tập 1: Các đơn vị đo lường trong máy tính

Yêu cầu bài tập:

1.1. Tính giá trị theo đơn vị đo dung lượng của những tập tin sau (chỉ đổi từ đơn vị lớn => nhỏ):

Dung lượng tập tin	Đơn vị GB	Đơn vị MB	Đơn vị KB	Đơn vị Byte	Đơn vị bit
189 KB	-	-	189	193,536	1,548,288
6.7 MB	-	6,7	6,860,8	7,025,459,2	56,203,673,6
500 MB	-	500	512,000	524,288,000	4,194,304,000
1.2 GB	1.2	1228,8	1,258,291,2	1,288,490,89	1,030792151*10 ¹⁰

1.2. Từ thông số về tần số (*Frequency*) hoạt động của các đường truyền (hoặc thiết bị) dưới đây, hãy diễn giải lại số lần truyền (hoặc xử lý) dữ liệu trong thời gian 1 giây (*second*):

Thông số thiết bị	Số lần truyền / xử lý trong thời gian 1 second
RAM bus 400 MHz	400,000,000 lần truyền / 1 sec
RAM bus 1333 MHz	1,333,000,000 lần truyền / 1 sec
CPU có FSB 800 MHz	800,000,000 lần truyền / 1 sec
CPU 2.2 GHz	2,200,000,000 lần truyền / 1 sec

1.3. Tính băng thông (*Bandwith* - dung lượng dữ liệu truyền trong thời gian 1 second) của các thiết bị sau:

Thông số thiết bị	Độ rộng đường truyền	Dung lượng dữ liệu truyền trong thời gian 1 second
RAM bus 400 MHz	64 bit	3,200 MB/s
RAM bus 1333 MHz	64 bit	10,664 MB/s
CPU có FSB 800 MHz	64 bit	6,400 MB/s
CPU có FSB 1333 MHz	64 bit	10,664 MB/s
USB 2.0 (480MHz)	Serial	48 MB/s
HDD SATA-2 (3GHz)	Serial	300 MB/s
HDD SATA-3 (6GHz)	Serial	600 MB/s

Thông số thiết bị	Độ rộng đường truyền	Dung lượng dữ liệu truyền trong thời gian 1 second
Card mạng Fast Ethernet	Serial	100 MB/s
Card mạng Gigabit Ethernet	Serial	1000 MB/s
Cáp quang 15 “mê”	Serial	15 MB/s
CPU có QPI 4.8 GT/s (GT/s = Giga transfer / sec)	Lane = 1 bit	480MB/s
	Lane = 8 bits	4.8GB/s
	Lane = 16 bits	4.8GT/s
CPU có DMI 2.5 GT/s (GT/s = Giga transfer / sec)	Lane = 1 bit	250 MB/s
	Lane = 8 bits	4.8GB /s
	Lane = 16 bits	2.5GB/s

1.4. Giải thích các thông số thiết bị sau:

Thông số thiết bị	Thuật ngữ của thông số	Giải thích
Máy in 600 DPI	Dots per inch	600 điểm mực trong khoảng cách 1 inch
Máy in 1200 DPI	Dots per inch	1200 điểm mực trong khoảng cách 1 inch
HDD 7200 RPM	Revolutions Per Minute	là chỉ số dùng để tính toán vận tốc quay hoặc tần số quay xung quanh một trục cố định của một vật trong một khoảng thời gian
Máy ảnh 5 MP	Mega Pixel	là đơn vị đo độ phân giải của thiết bị quang (như là số điểm ảnh trong bức ảnh) đặc biệt trong camera số
Ảnh màu 24 bits	binary digit	là đơn vị nhỏ nhất dùng để biểu thị những thông tin trong máy tính và là đơn vị cơ bản dùng để đo lường
Màn hình 1080x1920	Full HD	là chuẩn Full HD với độ phân giải 1920x1080 (1920 số điểm ảnh ngang và 1080 điểm ảnh dọc)
Audio 192 Kbps	Kilo bit per second	là đơn vị đo tốc độ truyền dẫn dữ liệu, tương đương 1000 bit trên một giây
Video HD 720p	High Definition	là 1 màn hình HD với độ phân giải là 1280 x 720 pixels, tương đương khoảng 1 triệu điểm ảnh
Video Full HD 1080p	Full High Definition	là FHD có độ phân giải 1.920 × 1.080 pixels (điểm ảnh) với tỷ lệ khung hình 16:9
Video Ultra HD 4K	Ultra High Definition	là các mẫu TV 4K Ultra HD sẽ cho độ chi tiết gấp 4 lần độ phân giải Full HD 1080p

Bài tập 2: Chuyển đổi các hệ đếm

Yêu cầu bài tập:

2.1. Thực hiện chuyển đổi giá trị hệ Binary => Hexa và Decimal

Số hệ Binary	Số hệ Hexa	Số hệ Decimal	Số hệ Binary	Số hệ Hexa	Số hệ Decimal
1100 1010	CA	202	0011 1011	3B	59
0100 0001	41	65	0011 1100	3C	60
1010 1100	AC	172	1111 1111	FF	255

2.2. Thực hiện chuyển đổi giá trị hệ *Decimal* => *Binary* và *Hexa*

Số hệ Decimal	Số hệ Binary	Số hệ Hexa	Số hệ Decimal	Số hệ Binary	Số hệ Hexa
65	0100 0001	41	128	1000 0000	80
97	01100001	61	192	1100 0000	C0
150	10010110	96	224	1110 0000	E0
168	10101000	A8	240	1111 0000	F0

2.3. Chuyển đổi địa chỉ IP: **192.168.1.129** thành dãy 32 số Nhị phân (4 *octet*):

192	168	1	129
11000000	10101000	00000001	10000001

Bài tập 3: Biểu diễn và tính toán số âm dưới dạng bù 2:

Yêu cầu bài tập:

3.1. Biểu diễn các số âm dưới đây dưới dạng số bù 2:

Số hệ Decimal	Biểu diễn Binary	Biểu diễn bù 2	Số hệ Decimal	Biểu diễn Binary	Biểu diễn bù 2
-9	1000 1001	1111 0111	-56	1011 1000	1100 1000
-20	1001 0100	1110 1100	-129	1000 0001	1111 1111
-16	1001 0000	1111 0000	-200	1100 1000	1011 1000

3.2. Thực hiện các phép toán có số âm (số âm được biểu diễn dạng bù 2):

Phép cộng	Số Binary hoặc bù 2	Phép cộng	Số Binary hoặc bù 2	Phép cộng	Số Binary hoặc bù 2
16	0001 0000	-5	1111 1011	18	0001 0010
-20	1110 1100	7	0000 0111	-30	1110 0010
KQ: -4	1111 1100	KQ:	0000 0010	KQ:	1000 1100

Hướng dẫn:

- Biểu diễn các số dưới dạng bit:
 - Biểu diễn **nhị phân** số **+16**: **0001 0000**
 - Biểu diễn **nhị phân** số **-20**: **1001 0100**
 - Biểu diễn **bù 2** của số **-20**: **1110 1100**
- Khi đưa số cho CPU tính toán, số âm **-20** được đưa vào dưới **dạng bù 2**, số dương **+16** giữ nguyên mẫu.
 - Biểu diễn **nhị phân** số **16**: **0001 0000**
 - Biểu diễn **bù 2** của số **-20**: **1110 1100**
 - Kết quả tính **16 - 20**: **1111 1100**

- Xử lý kết quả sau tính toán:
 - o Nếu kết quả tính toán trả về **số âm** (bit đầu là **1**): phải **tính bù 2** cho *kết quả* này để có giá trị **số âm biểu diễn dạng Binary**:
 - Minh họa: số bù 2: **1111 1100** => viết dạng *Binary*: **1000 0100** => giá trị là **-4**)
 - o Nếu kết quả tính toán trả về **số dương** (bit đầu là **0**) thì *kết quả* đó cũng là số biểu diễn dạng *Binary*.
 - o Với hệ xử lý **8 bit**, nếu kết quả tính nhiều hơn 8 bit thì **loại bỏ** bit cao nhất (bit thứ 9).

Bài tập 4: Các phép tính logic trên số nhị phân (bit).

Yêu cầu bài tập:

4.1. Tính kết quả của các phép tính *Logic* sau:

AND	1010 1010 1111 1111
KQ	1010 1010

AND	1010 1010 0000 0000
	0000 0000

OR	1010 1010 1111 1111
	1111 1111

OR	1010 1010 0000 0000
	101010

4.2. Dùng phép **XOR** để mã hóa dữ liệu.

- Giả sử: ta có 16 bit dữ liệu gốc **X = 1110 0101 0011 1011**
- Ta mã hóa dữ liệu trên bằng khóa 4 bit **K = 1010** dùng thuật giải *XOR*.
- Ta được dữ liệu đã mã hóa **Y = X ^ K**

X	1110 0101 0011 1011
K	1010 1010 1010 1010
Y = X ^ K	0100 1111 1001 0001

Yêu cầu: hãy dùng thuật giải **XOR** để **giải mã** dữ liệu mã hóa **Y** trong 2 trường hợp:

- Giải mã dùng khóa **K** đúng (giống với khóa đã dùng mã hóa dữ liệu **X**).
- Giải mã dùng khóa **K** sai (không giống với khóa đã dùng mã hóa dữ liệu **X**).

Trường hợp khóa K sai	
Y	0100 1111 1001 0001
K_T	1100 1100 1100 1100
X = Y ⊕ K	1000 0011 0101 1101

Trường hợp khóa K đúng	
Y	0100 1111 1001 0001
K_F	1010 1010 1010 1010
X = Y ⊕ K	1110 0101 0011 1011