

마이크로컨트롤러 기초(#514112)

#2. 자료의 표현과 연산

한림대학교
전자공학과 이선우

주요 학습 내용

- ▶ 컴퓨터 시스템의 데이터 표현 방식과 연산에 대해 학습
 - ▶ 진법 (2진수, 16진수)
 - ▶ 기초 연산 방법
 - ▶ 실수 표현 방법
 - ▶ 문자 표현 방법

진법

▶ n진법의 수

- ▶ $a_3a_2a_1a_0.b_1b_2$ 로 표시되는 수 →

$$a_3 \times n^3 + a_2 \times n^2 + a_1 \times n^1 + a_0 \times n^0 + b_1 \times n^{-1} + b_2 \times n^{-2}$$

▶ 예제

MSB(Most Significant Bit)

LSB(Least Significant Bit)

- ▶ 2진수: 0,1로 표현, 101010.01 → 42.25
- ▶ 8진수: 0,...,7로 표현, 5432.1 → 2842.125
- ▶ 16진수: 0,...,9,A,...,F로 표현, 2FA → 762
- ▶ 10진수: 13.75의 다른 진법으로의 표현
 - ▶ 2진수: 1101.11
 - ▶ 8진수: 15.6

정수와 실수의 표현방법

- ▶ 보수: 음의 정수를 나타내는 방법
- ▶ 1의 보수 (1's complement)
 - ▶ k자리로 표현되는 2진수 N의 1의 보수는 $(2^k-1)-N$ 으로 정의.
 - ▶ 2진수 N의 1의 보수: $0 \rightarrow 1, 1 \rightarrow 0$
 - ▶ 예: $101010 \rightarrow 010101$
- ▶ 2의 보수 (2's complement)
 - ▶ k자리로 표현되는 2진수 N의 2의 보수는 2^k-N 으로 정의.
 - ▶ 2진수 N의 2의 보수: $0 \rightarrow 1, 1 \rightarrow 0$ 로 바꾼 후, 1을 더함.
 - ▶ 예: $101010 \rightarrow 010101+1 \rightarrow 010110$

정수의 표현: 양수/음수 표시 방법

- ▶ +/-기호 대신 부호비트 사용
- ▶ N bit로 수를 표시할 때 1bit는 부호, N-1 bit는 크기 정보 표시
- ▶ 0: 양수, 1:음수
- ▶ 3가지 방식
 - ▶ Signed magnitude
 - ▶ Signed 1's complement
 - ▶ Signed 2's complement
- ▶ 예제 #1: -7를 5bit으로 표시
 - ▶ Signed magni.: 1 0111
 - ▶ 1's complement: 1 1000
 - ▶ 2's complement: 1 1001
- ▶ 8bit로 정수 표시할 때 표현 가능한 수의 범위
 - ▶ Signed mag.: +127~-127
 - ▶ 1's comp.: +127~-127
 - ▶ 2's comp.: +127 ~-128
- ▶ 2's complement 방식 채택

실수의 표현

- ▶ 부동소수점에 의한 실수의 표현
- ▶ 가수부와 지수부로 나누어 표현
- ▶ 예: 1234.56 $\rightarrow 0.\underline{123456} \times \underline{10^4}$
- ▶ IEEE 표준안 (단정도) : 32bit [1bit(sign):8bit(E:지수부):23bit(M:가수부)]
가 표시하는 숫자 = $1.M \times 2^{E-127}$
- ▶ 예제: $1.001010000 \times 2^{-87}$ 을 IEEE 표준 형식으로 표현하라.
 - ▶ 지수부: $-87+127=40=00101000$, 가수부: $00101000\dots$
 $\rightarrow 0\ 00101000\ 001010000\dots0$ (23bit)
- ▶ IEEE 표준안:
 - ▶ 배정도(64bit): 1(S):11(E):52(M)

문자의 표현

▶ ASCII Code

- ▶ American Standard Code for Information Interchange (ASCII)
 - ▶ 7bit→128개 문자 (프린트 가능한 94개의 문자, 34개의 제어 문자)
 - ▶ 94개의 프린트 가능 문자: 0~9 숫자, A~Z, a~z
- ▶ EBCDIC(Extended BCD Interchanged Code): IBM의 메인 프레임 등에서 사용됨

문자의 표현

▶ Unicode

- ▶ **유니코드**(Unicode)는 전 세계의 모든 문자를 컴퓨터에서 일관되게 표현하고 다룰 수 있도록 설계된 산업 표준이다.
- ▶ In Wikipedia
(<http://ko.wikipedia.org/wiki/%EC%9C%A0%EB%8B%88%EC%BD%94%EB%93%9C>)
- ▶ 최신판은 2009년 10월에 공개된 유니코드 5.2
- ▶ UTF-8
 - ▶ 유니코드를 위한 가변 길이 문자 인코딩 방식의 하나
 - ▶ 한 문자 표시 위해 1byte~4byte 사용

정수의 연산: 덧셈&뺄셈

▶ 양수의 덧셈

▶ $0+0=0, 0+1=1, 1+1=10$

▶ 양수-양수 (=양수+음수; 결과>0)

▶ Ex.:

+13: 0 1101

-7: 1 1001 (2's compl.)

Sum: 0 0110, carry=1

▶ 양수-양수 (=양수+음수; 결과<0)

▶ Ex: 7-13

+7: 0 0111

-13: 1 0011

sum: 1 1010 (= -6), C=0

▶ 양수+양수 (No overflow)

▶ Ex.; 6+7

+6: 0 0110

+7: 0 0111

sum: 0 1101 (= +13)

정수의 연산

- ▶ 양수+양수(오버플로우 있는 경우)

▶ Ex: 13+7

+13: 0 1101

+7: 0 0111

sum: 1 0100 (= -12), C=0

- ▶ 표현가능한 범위를 넘음 (overflow)

- ▶ 음수+음수 (오버플로우 없음)

▶ Ex.: -6 + (-7)

-6: 1 1010

-7: 1 1001

sum: 1 0011 (= -13), C=1 → 정상

- ▶ Overflow 있는 경우

▶ Ex.: -13+(-7)

-13: 1 0011

-7: 1 1001

sum: 0 1100 (= 12), C=1

실수의 연산

▶ 부동소수점 덧셈/뺄셈

- ▶ 1단계: 지수 조정
- ▶ 2단계: 가수 +/- 실행
- ▶ 3단계: 결과 정규화

▶ Ex.

- ▶ $0.1011 \times 2^4 + 0.1110 \times 2^6$
- ▶ Step 1: $0.1011 \times 2^4 \rightarrow 0.001011 \times 2^6$
- ▶ Step 2: $0.1110 + 0.001011 = 1.000011$
- ▶ Step 3: 0.1000011×2^7

▶ 곱셈

- ▶ 1단계: 가수를 곱한다. 결과는 배정도(double precision) 가짐
- ▶ 2단계: 지수는 더함
- ▶ 3단계: 정규화

▶ Ex.

- ▶ $(0.101 \times 2^4) \times (0.11 \times 2^6)$
- ▶ 1: $0.101 \times 0.11 = 0.01111$
- ▶ 2: $4 + 6 = 10$
- ▶ 3: 0.11111×2^9

다른 이진 코드

▶ Gray code

- ▶ 인접한 연속적인 수들을 서로 한 비트만 틀려지게 나타내는 이진 코드 방법
- ▶ 특정한 용도로 이용

▶ BCD (Binary Coded Data) code

- ▶ 4bit의 2진수로 십진수의 각 자리 수 표시하는 방법
- ▶ 예: 13d → 0001 0011bcd

10진수	Gray code
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101
10	1111