Tema 1: Circuitos Digitales

1.4 Representación de Información

Miguel Ángel Otaduy

Universidad Rey Juan Carlos

Números Naturales

- Representación en base ‘a’, usando sólo cifras menores que ‘a’. Ej: decimal (10), binario (2), octal (8), hexadecimal (16)…
- Número natural en base ‘a’:

$$N = \alpha_n \dots \alpha_1 \alpha_0 = \sum_{i=0}^{n} \alpha_i a^i, \qquad 0 \le \alpha_i < a$$

- Ejemplos:

$$45 = 4 \cdot 10^1 + 5 \cdot 10^0$$
$$45 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \to 101101$$
$$45 = 5 \cdot 8^1 + 5 \cdot 8^0 \to 55$$
$$45 = 2 \cdot 16^1 + 13 \cdot 16^0 \to 2d$$

Transformación desde Decimal

- Algoritmo:

$$
\begin{array}{l}
N \\
i = 0 \\
while\ N > 0 \\
\qquad \alpha_i = N\ MOD\ a \\
\qquad N = N\ DIV\ a \\
\qquad i = i + 1
\end{array}
$$

  - DIV: división entera
  - MOD: resto

Binario ←→ Hexadecimal

- En un procesador los valores están almacenados en binario, pero es común representarlos en hexadecimal. La conversión es inmediata e intuitiva, y la representación es más eficiente
- Binario → Hexadecimal
  - Agrupar de 4 en 4 y convertir

$$011110110011 \rightarrow 0111\ \ 1011\ \ 0011 \rightarrow 7b3$$

- Hexadecimal → Binario
  - Convertir cada dígito a binario de 4 bits

$$a5c \rightarrow 1010\ \ 0101\ \ 1100 \rightarrow 101001011100$$

Enteros con Signo y Magnitud

- Se añade un bit para indicar el signo
  - 0: positivo; 1: negativo
- Operaciones aritméticas (suma/resta)
  - Sumar números del mismo signo / Restar números de signo contrario: se suman (sin signo) y se pone el mismo signo
  - Restar números del mismo signo / Sumar números de signo contrario: se resta el menor al mayor (sin signo), y se aplica el signo apropiado

Enteros con Exceso

- El número N en exceso a M se representa como N+M en binario puro
- Con n bits se usa exceso a $2^{n-1}-1$.
  De esa manera se puede representar el rango entre $-(2^{n-1}-1)$ y 2^{n-1} como el rango entre 0 y $2^{n}-1$
- Requiere conversión para pasar de entero con signo a entero sin signo

Enteros en Complemento a 2

- Representación del número negativo –N en binario de n bits en complemento a 2 → $2^n – N$
- El bit de mayor peso indica el signo, pero es parte del número, no es un símbolo

- Las operaciones aritméticas (suma y resta) se tratan como sumas sin signo
- Ejemplo: a + (-b), a>0, b>0 → a + (2^n - b)

  Si a > b → a – b, porque 2^n no entra en n bits

  Si a < b → 2^n – (b – a), negativo en comp. a 2

Coma Fija

- Se representan números reales con parte entera de tamaño fijo p y parte fraccionaria de tamaño fijo r:

$$N = \alpha_{p-1} \ldots \alpha_0 \quad . \quad \beta_1 \ldots \beta_r = \sum_{i=0}^{p-1} \alpha_i \, a^i + \sum_{i=1}^{r} \beta_i \, a^{-i}$$

- Ventaja: aritmética igual que con enteros

Coma Fija

- 0.N de decimal a binario de r bits:

$$
\begin{aligned}
&N = int(0.N \cdot 2^r) \\
&for\ i = r\ to\ 1 \\
&\qquad \beta_i = N\ MOD\ 2 \\
&\qquad N = N\ DIV\ 2
\end{aligned}
$$

Coma Flotante

- Mantisa (m) y Exponente (e) en base ‘a’:

$$N = m \cdot a^e, \qquad 1 < \, |m| < a$$

- Ejemplos en decimal:

  342 → $3.42 * 10^{2}$

  0.0057 → $5.7 * 10^{-3}$

  -12.456 → $-1.2456 * 10^{1}$

  -8.023 → $-8.023 * 10^{0}$

Estándar IEEE754

- Precisión simple (32 bits)
  - Mantisa como signo-magnitud. 1 bit de signo y 23 de la parte fraccionaria de la mantisa (el ‘1’ de la parte entera es implícito; no se almacena).
  - Exponente de 8 bits como exceso a 127.

- Precisión doble (64 bits)
  - Mantisa como signo-magnitud. 1 bit de signo y 52 de la parte fraccionaria de la mantisa.
  - Exponente de 11 bits como exceso a 1023.

- Números especiales para 0, NaN y ±inf.

Rango y Resolución

- Complemento a 2 de 32 bits
  - Rango: $[-2^{31}, 2^{31}-1]$
  - Resolución: 1

- IEEE754 simple
  - Rango: $\pm[2^{-126}, (2-2^{-23})*2^{127}]$
  - Resolución: depende del valor del exponente

- IEEE754 doble
  - Rango: $\pm[2^{-1022}, (2-2^{-52})*2^{1023}]$
  - Resolución: depende del valor del exponente

Precisión y Redondeo

- Al hacer operaciones en coma flotante se producen errores. Hay dos factores a tener en cuenta:
  - Precisión: es el error en el resultado. Cada operación (suma, multiplicación…) tiene un cierto número de bits de mantisa que se pueden considerar correctos. Al acumular operaciones se reduce la precisión.
  - Redondeo: el estándar IEEE754 establece también cómo redondear las operaciones.