S2 Aritmética Digital

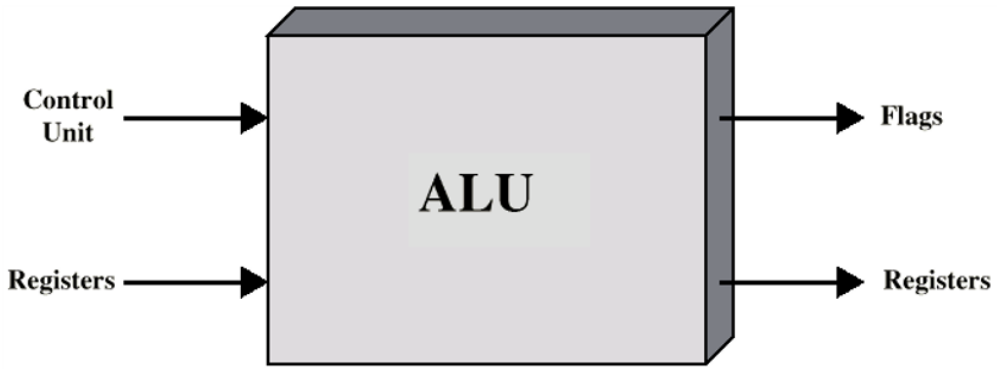
Lenin G. Falconí

2024-11-09

Outline

Unidad Aritmética Lógica (ALU)

- Está encargada de realizar las operaciones lógicas y aritméticas sobre los datos
- Está conformada de dispositivos electrónicos que permiten el almacenamiento de dígitos binarios y ejecutar operaciones Booleanas
- La ALU se interconecta por señales de control, utiliza 2 registros y emite flags



Representación de Números Enteros

- No se dispone de signos +/- para representar los números
- No se dispone de un punto decimal
- Un número entero queda representado por un conjunto de 0s y 1s
- Por tanto, un dígito ha de ser usado para representar el signo
- Una secuencia de n dígitos binarios se interpreta como un entero A sin signo

$$A = \sum_{i=0}^{n-1} 2^i a_i$$

Representación Signo - Magnitud I

- El bit más significativo de la izquierda (LMSB) se considera el signo
- $0 \rightarrow$ positivo
- $1 \rightarrow$ negativo
- Si la palabra tiene n dígitos:
  - El n - simo bit es el signo
  - Los $n-1$ bits son la magnitud

$$A = \begin{cases} \sum_{i=0}^{n-2} 2^i a_i & \text{if } a_{n-1} = 0 \\ -\sum_{i=0}^{n-2} 2^i a_i & \text{if } a_{n-1} = 1 \end{cases}$$

- El 0 tiene una representación doble como 0^+ y 0^-

Representación Signo - Magnitud II

$$+18 = 00010010$$

$$-18 = 10010010$$

Definition (Tarea)

Escribir una función en python que permita dado un número binario de 8 bits obtener su negativo usando el criterio de signo magnitud

Complemento a base disminuida $r-1$ I

Dado un número A en base r de n dígitos, el complemento a $r-1$ de A es:

$$(r^n - 1) - A$$

- Caso Decimal $r = 10$ y $r - 1 = 9$. Para un número de n dígitos se tiene: $(10^n - 1) - A$. Donde $(10^n - 1) = 999\ldots9 \rightarrow$ n 9s
  Ejemplo: El complemento a nueve de 546700 es
  $999999 - 546700 = 453299$
- Caso Binario $r = 2$ y $r - 1 = 1$, entonces $1111\ldots11 - A$. El resultado es la inversión del número
  Ejemplo: Sea $A = 1011000$, el complemento a 1 es
  $1111111 - 1011000 = 0100111$

Representación en Complemento a r

El complemento a r de un número A_r de n dígitos es el complemento a $r^n - A$ i.e.
$r^n - A = (r^n - 1) - A + 1$

- Caso Binario: Consiste en obtener el complemento a 1 o inversión del número binario y sumar 1

Sistema de Complemento a 2 I

- Utiliza el bit más significativo como signo
- Dispone de una sola representación para el 0
- Si se define un número de n bits como la secuencia $A = a_{n-1}a_{n-2}\ldots a_2a_1a_0$, entonces el bit del signo es el dígito a_{n-1} y el número A en complemento a 2 se representa por:

$$A = -2^{n-1}a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i$$

- El rango: -2^{n-1} hasta $2^{n-1} - 1$
- Se obtiene de invertir cada bit de la cadena A y luego sumar 1
- Extensión de la longitud en bits: Si el número A de n dígitos se ha de representar en m dígitos, donde $m \geq n$ entonces, se añade las posiciones faltantes a la izquierda y se rellenan con el valor del bit del signo original.

Sistema de Complemento a 2 II

- Overflow: se produce cuando al sumar dos números A y B en complemento a 2, el resultado tiene signo opuesto.
- Resta: dados A y B y se desea obtener $A - B$, entonces

$$A - B = A + (-B)$$

Conversión de binario a decimal en complemento a 2

Consiste en aplicar la ecuación de un número en complemento a 2:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	0	0	0	0	1	1
-128	0	0	0	0	0	+2	+1

$$A = -2^{n-1} a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i$$

$$A = 1000\,0011_2 = -128 + 2 + 1 = -125$$

Representaciones con Signo I

- Los números negativos se representan por su complemento
- En un sistema de numeración binaria (e.g. ALU) se usa el complemento a 2
- Se asume que el 0 en la MSB es positivo

Por ejemplo -9 en una máquina de 8 bits puede representarse como:

Sistema	-9
Magnitud Signo	10001001
Complemento 1	11110110
Complemento 2	11110111

Suma de Enteros sin Signo I

- En binario se ha de considerar que:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

- La operación puede generar un acarreo.
- En general, dados dos números A_1 y A_2, se procede a sumar cada uno de los dígitos $z_i = a_{1_i} + a_{2_i}$. Si $z_i \leq r$, el resultado tiene un dígito dentro del sistema de numeración. Si $z_i \geq r$, entonces se resta la base y se desplaza el acarreo a la siguiente posición.

Ejemplo: Sume AAF a F3C

Suma de Enteros sin Signo II

CF	1		1	
A1		A	A	F
A2		F	3	C
Z	1	9	E	B

Resta de Enteros sin Signo

- En binario se ha de considerar que:

$$1 - 0 = 1$$

$$0 - 1 = 1$$

, $CF = 1$

- La operación puede generar un acarreo que se suma al sustraendo.

Suma de Enteros con signo

La operación de suma no se ve afectada esencialmente. Sin embargo, se debe notar que en este caso el bit más significativo se interpreta como signo.

Resta de Enteros con signo en Complemento a 2 I

- El complemento a 2 permite resolver la resta de enteros como una suma
- Obtener el complemento a 2 del sustraendo y sumar al minuendo
- Regla de Desbordamiento: Si el resultado de una suma de dos números de igual signo produce un signo opuesto
- Regla de extensión de bits: Si se desea representar un número de n bits en m bits donde $m \leq n$ se completa con copias del bit del signo las posiciones nuevas.

Por ejemplo, considere la operación restar $3A5_{16}$ de 592_{16}

1. Sea $M = 0592_{16}$ (Minuendo), usando un dígito más para signo
2. Sea $S = 03A5_{16}$ (Sustraendo), usando un dígito más para signo
3. Obtener el complemento a 2 del sustraendo i.e.

$$\bar{S} = FFFF_{16} - 03A5_{16} + 1 = FC5A + 1 = FC5B$$

Resta de Enteros con signo en Complemento a 2 II

4. Obtener la suma del complemento a 16 y el minuendo:

$$0592 + FC5B = 01ED$$

con $CF = 1$

Overflow

- Considere las operaciones (+5) + (+4), (-7)+(-6), usando el complemento a 2 para una representación en 4 bits.
- Al representar en 4 bits y utilizar el complemento a 2, se reserva el bit más significativo para el signo, entonces

La suma de (+5)+(+4):

	0	1	0	1
+	0	1	0	0
	1	0	0	1

La suma de (-7)+(-6):

	1	0	0	1
+	1	0	1	0
1	0	0	1	1

Representación en Coma Flotante I

- La representación en coma fija de complemento a 2 no permite representar números muy grandes o muy pequeños (e.g. el diámetro de la Tierra)
- La representación en coma flotante binaria se define por:

$$\pm S \times B^{\pm E}$$

- $\pm$: signo
- S: Mantisa o parte más significativa
- E: Exponente (representación sesgada)

Estándar IEEE 754 en Coma Flotante en 32 bits

- Se utiliza un bit para el signo: 0$\rightarrow$ positivo, 1 $\rightarrow$ negativo.
- El exponente está sesgado: $2^{k-1}-1$ y k es el número de bits del exponente. Por tanto, tiene un sesgo ($BIAS$) de 127 y un rango de -127 a +128
- La mantisa es normalizada: $\pm 1.bbb\ldots b \times 2^{\pm E}$. La mantisa almacena la parte fraccionaria.

S	Exp 8 Bits	Mantisa 23 Bits

Estándar IEEE 754 en Coma Flotante en 64 bits

Dada la siguiente representación de números en 64 bits, cuáles son los rangos del exponente?

S	Exp 11 Bits	Mantisa 52 Bits

- ¿Cuánto vale el sesgo?
- ¿Cuál es el rango del exponente?

Estándar IEEE 754 en Coma Flotante en 64 bits

Dada la siguiente representación de números en 64 bits, cuáles son los rangos del exponente?

S	Exp 11 Bits	Mantisa 52 Bits

- Sesgo: $2^{11-1} - 1 = 1023$
- ¿Cuál es el rango del exponente? En 11 bits podemos representar números desde 000 hasta 7FF, es decir 0 a 2047. Restando el sesgo de los extremos se tiene que el rango sería:

$[0, 2047] - 1023 = [-1023, 1024]$

Conversión de Decimal a IEEE754 en 32 bits I

Convertir en representación de coma flotante de 32 bits: -248.75
Procedimiento:

- Identificar Bit de signo
- Convertir a binario el valor sin signo
- Desplazar la como a la izquierda del 1 más significativo (normalización):$1.bbb\cdots \times 2^E$
- Ajustar con el sesgo el exponente: $127 + E$
- Convertir a binario el exponente sesgado
- Colocar en la mantisa la fracción: $1.bbb\ldots b$
- Completar con 0s los bits faltantes de la mantisa

Visita la calculadora en línea

Solución I

- Bit de signo: 1
- Convertir a binario el valor sin signo: 11111000.11
- Desplazar la como a la izquierda del 1 más significativo (normalización):$1.111100011 \times 2^{-7}$
- Ajustar con el sesgo el exponente: $127 + 7 = 134$
- Convertir a binario el exponente sesgado: 10000110
- Colocar en la mantisa la fracción: 1. 111100011
- Completar con 0s los bits faltantes de la mantisa

1	1	0	0	0	0	1	1	0	1	1	1	1	0	0	0	1	1	0	...	0
S	exponente								mantisa											

Resp: $C378C000_{16}$

Conversión de IEEE754 de 32 bits a Decimal

Sea s el valor del bit del Signo, m la mantisa y e el exponente, entonces, el decimal equivalente se da por:

$$(-1)^s \times 1.m \times 2^{e-127}$$

- Convierta $C378C000_{16}$ a decimal

Límites de representación en coma flotante 32 bits

1. La representación en coma flotante no puede representar a todos los números reales.
2. Números Negativos entre $-(2-2^{-23}) \times 2^{128}$ y -2^{-127}
3. Números Positivos entre 2^{-127} y (2-2^{-23})$\times$ 2^{128}\$

Definition (Tarea)

1. Consultar ¿cuáles son el número más grande y más pequeño que puede representar la IEEE754 de 32 bits?
2. Consultar cómo se representa 0^+, 0^-, $+\infty$, $-\infty$ en IEEE754
3. ¿Cuál es la condición NaN?

Límites de representación en coma flotante 32 bits

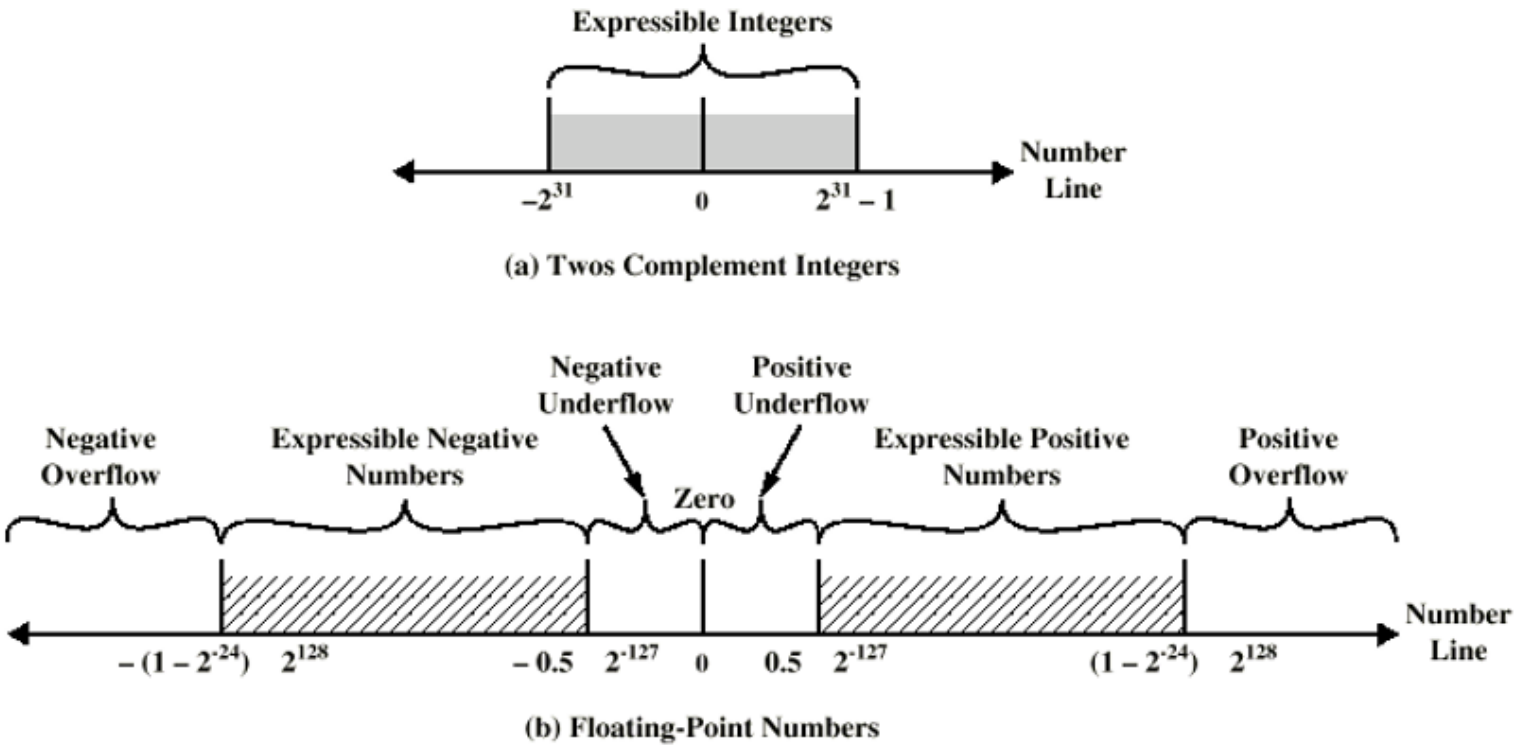


Figure: Comparación de números representables con 32 bits. Parte superior enteros en complemento a 2. Parte inferior Notación de Coma flotante

Ejercicios varios: I

1. Convierta -0.5 en IEEE 754 de 32 bits
2. Convierta +234.75 en IEEE 754 de 32 bits
3. Convierta $402DF854_{16}$ de IEEE 754 32 bits a decimal
4. En clase se estudió el algoritmo utilizado por los desarrolladores de Quake para obtener la raíz cuadrada inversa. En el vídeo se presenta el número hexadecimal 0x5F3759DF, el cual ocupa 32 bits. Suponga que el número es la representación en coma flotante de un número decimal X_{10}, ¿Cuál es el número?