Lección Anterior

Operandos

Tipo y tamaño de operandos

Tipos comunes

IEEE 754

Representación Interna

Conversión

Ejemplo

Práctica

Tipos de operaciones

Control de Flujo

Instrucciones de control de flujo

Direccionamiento en Control de Flujo

Dirección relativa

Codificación

Referencias

Operandos

Operandos: Tipo y tamaño de operandos

El tipo de operandos normalmente se encuentra, en la mayoría de los casos, en el código de operación (opcode). Alternativamente los datos son anotados mediante etiquetas (*tags*) que son interpretadas por el HW (mnemónico).

Byte (8 bits).

Half-Word (16 bits).

Word (32 bits) / punto flotante de precisión simple (SP-FP).

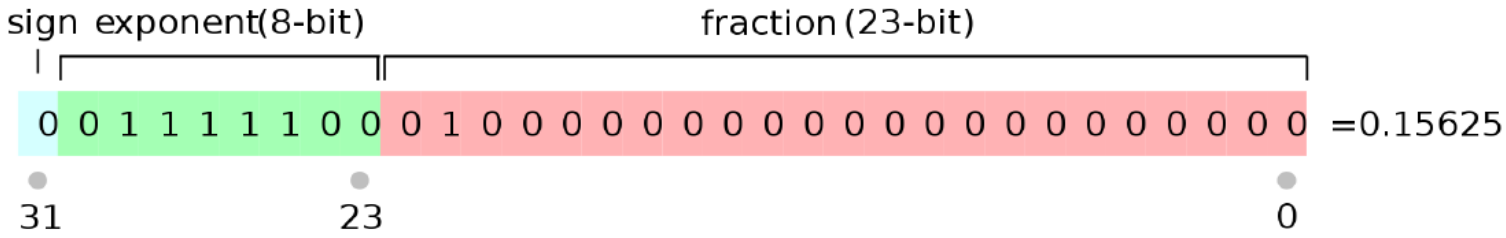
DoubleWord (64 bits) / punto flotante de precisi ón doble.

Enteros: Típicamente en representación complemento a 2.
Flotantes: IEEE 754*.
Caracteres: ASCII.
Decimal: BCD.

Conversión de Decimal a Punto Flotante IEEE 754

IEEE 754

¿Cómo se representan decimales?
Ejemplo: Representar $0{,}10_{10}$, en base 2: $0{,}0\overline{0011}$



El exponente es desplazado mediante un sesgo para poder representar exponentes negativos.

1. Convertir valor decimal a binario (solo número ignorar el signo).
2. Colocar el número de la forma: numero$\times 2^0$.
3. Denotar el número de la forma $1, a_1 a_2 \ldots \; a_i \times 2^n$ (se corre la coma n espacios).
4. Determinar el signo: 0 si N es mayor que 0, 1 si N es menor que 0.
5. Determinar el exponente como $E = n + 127$, luego pasar a binario.
6. Determinar la mantisa F como $F = a_1 a_2 \ldots \; a_i$.
7. Escribir el número según IEEE, completando con ceros a la derecha los 23 bits de la mantisa.

IEEE 754

Ejemplo: Convertir el número 22,625 a flotante

1. Paso 1: Convertir 22_{10} a binario: 10110_2. Convertir la parte decimal $0{,}625_{10}$ a binario: $0{,}625_{10} = 101_2$.
2. Paso 2: Por lo tanto $22{,}625_{10} = 10110{,}101_2$ en notación científica sería $22{,}625_{10} = 10110{,}101_2 \times 2^0$.
3. Paso 3: Se normaliza a $1{,}0110101_2 \times 2^4$. Por lo tanto $n = 4$.
4. Paso 4: Se obtiene el signo (0).
5. Paso 5: Se obtiene el exponente $E = 127 + n \rightarrow E = 127 + 4 = 131_{10} = 10000011_2$.
6. Paso 6: Se obtiene la mantisa $F = 01101010000000000000000_2$.
7. Paso 7: Se coloca en notación IEEE Signo + Exponente + Mantisa $= 01000001101101010000000000000000_2$.

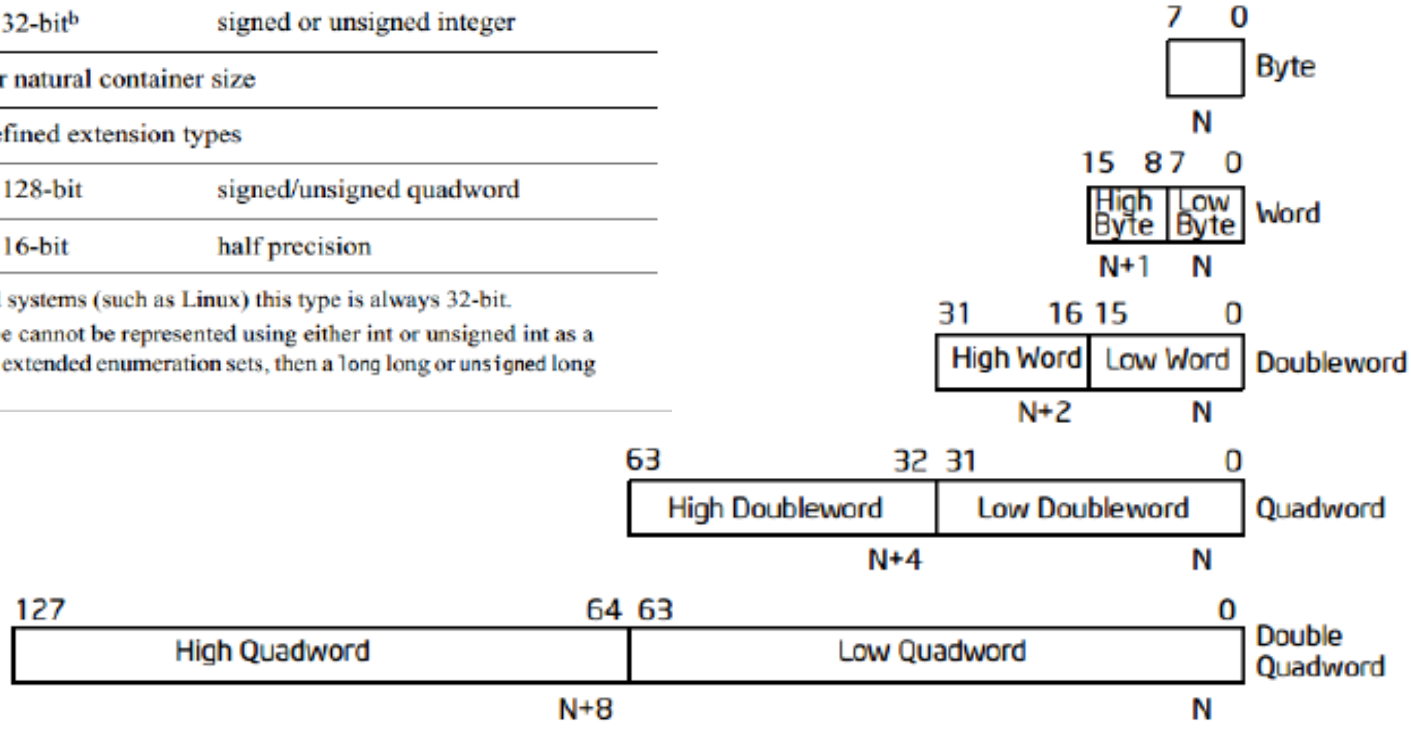
Operandos

Operandos: Ejemplo x86

Type	A32	A64	Description
int/long	32-bit	32-bit	integer
short	16-bit	16-bit	integer
char	8-bit	8-bit	byte
long long	64-bit	64-bit	integer
float	32-bit	32-bit	single-precision IEEE floating-point
double	64-bit	64-bit	double-precision IEEE floating-point
bool	8-bit	8-bit	Boolean
wchar_t[a]	16-bit unsigned	16-bit unsigned	short (compiler dependent)
	32-bit unsigned	32-bit unsigned	int (compiler dependent)
void* pointer	32-bit	64-bit	addresses to data or code
enumerated types	32-bit	32-bit[b]	signed or unsigned integer
bit fields	not larger than their natural container size		
ABI defined extension types			
__int128/__uint128	128-bit	128-bit	signed/unsigned quadword
__f16	16-bit	16-bit	half precision

a. Environment-dependent. In GNU-based systems (such as Linux) this type is always 32-bit.

b. If the set of values in an enumerated type cannot be represented using either int or unsigned int as a container type, and the language permits extended enumeration sets, then a long long or unsigned long long container may be used.



Codificación

Al tener un tamaño de palabra finito y determinado para las instrucciones, es vital maximizar su uso para poder describir cada instrucción.

Respecto al tamaño de instrucción existen 3 formas comunes de longitud de instrucción:

Variable $\rightarrow$ CISC.

Fija $\rightarrow$ RISC.

Híbrido.

Dirección relativa

Name	Examples	How condition is tested	Advantages	Disadvantages
Condition code (CC)	80x86, ARM, PowerPC, SPARC, SuperH	Tests special bits set by ALU operations, possibly under program control.	Sometimes condition is set for free.	CC is extra state. Condition codes constrain the ordering of instructions since they pass information from one instruction to a branch.
Condition register	Alpha, MIPS	Tests arbitrary register with the result of a comparison.	Simple.	Uses up a register.
Compare and branch	PA-RISC, VAX	Compare is part of the branch. Often compare is limited to subset.	One instruction rather than two for a branch.	May be too much work per instruction for pipelined execution.

Codificación

Ejemplo

Un set de instrucciones de **16 bits**, está constituido por instrucciones de **0, 1 y 2 operandos**,los operandos tienen un tamaño de 6 bits, si en él ya existen 5 instrucciones de 2 operandos y 33 instrucciones de 0 operandos. ¿Cuál es el número máximo de instrucciones de 1 operando que se pueden codificar con dicho ancho de palabra y como sería la codificación del ISA?

A	B	Y
0	0	0 op
0	1	1 op
1	0	2 op
1	1	X

Se necesita diferenciar entre si es un operando de 0, 1 o 2 operandos. Para identificarlos se usan 2 bits.

$$1\text{operandos} \rightarrow 6\text{bits} \rightarrow \mathbf{256\,instrucciones}$$

15 14	13 8	7 0
# Op	1 operandos (6 bits)	Id inst. (**256 instrucciones**)

$$0\text{operandos} \rightarrow 0\text{bits} \rightarrow 33\text{instrucciones}$$

15 14	13 0
# Op	Id inst. (14 bits o 16384 instrucciones)

$$2\text{operandos} \rightarrow 12\text{bits} \rightarrow 5\text{instrucciones}$$

15	14 3	2 0
# Op	2 operandos (12 bits $= 2 \times 6$ bits)	Id inst.