

L'esponente è in forma complementata a due; il valore effettivo si ricava come segue:

real*4	EspEff = ESPONENTE – 127
real*8	EspEff = ESPONENTE – 1023
real*10	EspEff = ESPONENTE – 16383

Si riportano tre procedure in MASM X64 per l'estrazione della mantissa e dell'esponente di "reali" real*4 (DWORD), real*8 (QWORD) e real*10 (TBYTE):

```
.data
MSKD          DWORD          7FFFFFFh
MSKQ          QWORD          0FFFFFFFFFFFFFFh

.code
;-----D_EXTRACT-----
;---Sintassi >>> invoke64 D_EXTRACT,ADDR NUMERO <<< (NUMERO WORD)
;---USCITA  MANTISSA con segno in ecx, punto decimale dopo il bit 23 , esponente con segno in bl
;---Segno in complemento a 2
D_EXTRACT PROC FRAME
.endprolog

                mov     ecx,[rcx]
                mov     ebx,ecx
                mov     eax,MSKD
                and     ecx,eax
                bts     ecx,23
                not     eax
                and     ebx,eax
                btr     ebx,31
                jnc     @F
                neg     ecx
@@:
                clc
                rol     ebx,9
                sub     ebx,127
                ret

D_EXTRACT ENDP

;-----Q_EXTRACT-----
;---Sintassi >>> invoke64 Q_EXTRACT,ADDR NUMERO <<< (NUMERO QWORD)
;---USCITA  MANTISSA con segno in rcx, punto decimale dopo il bit 52 , esponente con segno in bx
;---Segno esponente in complemento a 2

Q_EXTRACT PROC FRAME
.endprolog

                mov     rcx,[rcx]
                mov     rbx,rcx
```

```

mov     rax,MSKQ
and     rcx,rax
bts     rcx,52
not     rax
and     rbx,rax
btr     rbx,63
jnc     @F
neg     rcx
@@:
clc
rol     rbx,12
sub     rbx,1023
ret

```

Q_EXTRACT ENDP

-----T_EXTRACT-----

----Sintassi >>> invoke64 T_EXTRACT,ADDR NUMERO <<< (NUMERO TBYTE)

----USCITA MANTISSA senza segno in rcx, punto decimale dopo il bit 63 , esponente con segno in bx

----Segno ESPONENTE in complemento a 2, segno MANTISSA, 0 per +, 1 per -, al 16 bit di ebx

T_EXTRACT PROC FRAME

.endprolog

SYMBOL xxx1

```

xor     rbx,rbx
mov     bx,WORD PTR [rcx+8]
mov     rcx,QWORD PTR [rcx]
shl     ebx,1
shr     bx,1
sub     bx,16383
ret

```

T_EXTRACT ENDP

RADICE QUADRATA

L'operazione è fatta su una MANTISSA $X > 0$ formata da una sequenza di cifre binarie $x_j \equiv (0 \text{ o } 1)$ con indice j decrescente fino a 0. Ciascuna cifra ha un peso determinato da un fattore pari a 2^j dipendente dalla sua posizione j ; i termini così composti sono additivi. Essa fornisce Y , equivalente sequenza di bit, tale che $X=Y^2$:

MANTISSA $X=x_k x_{k-1} x_{k-2} \dots x_2 x_1 x_0$ RADICEq $Y=y_i y_{i-1} y_{2-2} \dots y_2 y_1 y_0$

dove "i" è la metà intera per difetto dell'indice che individua il bit più significativo ($x_k=1$) della mantissa; in altri termini "i" è l'intero tale che:

$$2^{2*i} \leq X \leq 2^{2*(i+1)} \quad \text{oppure} \quad 2^i \leq Y < 2^{i+1}$$

Infatti, il quadrato della seconda relazione da la prima.

Le cifre di X rappresentano, per il momento, un intero privo di decimali e di esponente.

Trovato "i" si attribuisce alla cifra y_i di Y il valore 1, sia x_k sia y_i sono le cifre più significative rispettivamente di X e Y e valgono 1.

Questa prima componente di Y, vale $Y_1 = y_i * 2^i$.

a) Se per caso $X = 2^{2*i}$ allora $Y = y_i * 2^i = Y_1$ e il calcolo finisce; altrimenti:

1) il valore di Y, da trovare, si compone intanto del valore corrispondente alla cifra y_i , cioè Y_1 , più un residuo r_1 comunque minore dell'intervallo che contiene Y:

$$r_1 < 2^{i+1} - 2^i = 2^i \quad Y = Y_1 + r_1 \quad \text{da cui} \quad X = (Y_1 + r_1)^2$$

$$X = Y_1^2 + 2 * Y_1 * r_1 + r_1^2 = Y_1^2 + (2 * Y_1 + r_1) * r_1$$

Quindi esaurito il ruolo della cifra $x_k = y_i = 1$ di X il rimanente è:

$$X_1 = X - Y_1^2 = (2 * Y_1 + r_1) * r_1$$

Se r_1 , di Y, è uguale al valore corrispondente alla successiva cifra $y_{i-1} = 1$ di peso 2^{i-1} , il valore ottenuto dalla precedente X_1 ponendo $r_1 = 2^{i-1}$: $(2 * Y_1 + 2^{i-1}) * 2^{i-1}$, diventa uguale, cosa naturalmente improbabile, al rimanente di X, cioè a X_1 e il calcolo sarebbe finito (punto a)).

Altrimenti, dal confronto tra $(2 * Y_1 + 2^{i-1}) * 2^{i-1}$ e $X_1 = X - Y_1^2$ si deduce che se il primo valore predomina sul secondo significa che se la cifra y_{i-1} è posta a 1, il suo valore 2^{i-1} sarebbe eccessivo rispetto a r_1 pertanto $y_{i-1} = 0$.

Invece se, $X_1 = X - Y_1^2 > (y_i * 2^{i+1} + 2^{i-1}) * 2^{i-1}$, allora $y_{i-1} = 1$, e Y riduce il suo residuo a r_2 più piccolo di r_1 , minore questa volta di 2^{i-1} :

$$y_{i-1} = \begin{cases} 0 & \text{se } (2 * Y_1 + 2^{i-1}) * 2^{i-1} > X_1 = (2 * Y_1 + r_1) * r_1 \\ 1 & \text{se } (2 * Y_1 + 2^{i-1}) * 2^{i-1} < X_1 = (2 * Y_1 + r_1) * r_1 \\ 1 & \text{se } (2 * Y_1 + 2^{i-1}) * 2^{i-1} = X_1 = (2 * Y_1 + r_1) * r_1 \rightarrow \text{FINE CALCOLO} \end{cases}$$

$$Y = y_i * 2^i + y_{i-1} * 2^{i-1} + r_2 = Y_2 + r_2 \quad \text{con} \quad Y_2 = Y_1 + y_{i-1} * 2^{i-1} \quad \text{e} \quad r_2 < 2^{i-1}$$

Se $y_{i-1} = 0$, cioè $Y_2 = Y_1$, allora $r_1 = r_2$

$$\text{continuando} \quad X = (Y_2 + r_2)^2 = Y_2^2 + 2 * Y_2 * r_2 + r_2^2 = Y_2^2 + (2 * Y_2 + r_2) * r_2$$

Il rimanente di X diventa:

$$X_2 = X - Y_2^2 = (2 * Y_2 + r_2) * r_2$$

2) Si riprende il discorso dal punto 1) questa volta con Y_2 al posto di Y_1 per determinare la cifra y_{i-2} :

$$y_{i-2} = \begin{cases} 0 & \text{se } (2 * Y_2 + 2^{i-2}) * 2^{i-2} > X_2 = (2 * Y_2 + r_2) * r_2 \\ 1 & \text{se } (2 * Y_2 + 2^{i-2}) * 2^{i-2} < X_2 = (2 * Y_2 + r_2) * r_2 \\ 1 & \text{se } (2 * Y_2 + 2^{i-2}) * 2^{i-2} = X_2 = (2 * Y_2 + r_2) * r_2 \quad \rightarrow \text{ FINE CALCOLO} \end{cases}$$

.....

$$Y = y_i * 2^i + y_{i-1} * 2^{i-1} + y_{i-2} * 2^{i-2} + r_3 = Y_3 + r_3 \quad \text{con} \quad Y_3 = Y_2 + y_{i-2} * 2^{i-2} \quad \text{e} \quad r_3 < 2^{i-2}$$

$$X_3 = X - Y_3^2 = (2 * Y_3 + r_3) * r_3$$

Iterando si individuano tutte le cifre $y_i y_{i-1} y_{i-2} \dots y_2 y_1 y_0$, oltre la macchina non consente di andare e il residuo finale $r_{i+1} < 2^0 = 1$.

Se la mantissa è scritta in forma normale, $0 \leq X < 2$ in binario $0 \leq X < 10_b$, cioè il punto decimale è posto dopo la prima cifra più significativa, anche la radice $0 \leq Y < 10_b$ e il punto decimale verrebbe a trovarsi subito dopo la cifra "i"; e il residuo finale passa a $r_{i+1} < 2^{-i}$.

La presenza di una parte esponenziale: $X * 2^E$, una volta trovata la radice Y di X come descritto nella procedura sopra riportata, a questa deve aggiungersi una parte esponenziale corrispondente alla metà della parte esponenziale "E" di X : $Y * 2^{E/2}$.

Però in questa eventualità bisogna distinguere due casi: il primo è quello in cui E sia pari e allora non c'è nulla da aggiungere a quanto detto; nell'altro caso la metà di "E" non è intera in tal caso i seguenti passaggi risolvono il problema:

se E è dispari $Y * 2^{E/2} = Y * 2^{\text{int}(E/2)+1/2} = Y * 2^{\text{int}(E/2)} * 2^{1/2} = \sqrt{2} * Y * 2^{\text{int}(E/2)}$
dove $\text{int}(\cdot)$ è l'intero immediatamente più piccolo dell'argomento; tutto si riduce a un fattore costante pari alla radice di 2.

LOGARITMO

LOGARITMO IN BASE 2 DI UN NUMERO

$$\lg_2(X * 2^E) = E + \lg_2(X) \quad \text{dove} \quad X * 2^E > 0$$

La Mantissa è positiva; nel caso di una mantissa negativa o più in generale per il logaritmo di un numero complesso, il calcolo, come è noto, è riducibile al precedente e a una funzione trigonometrica inversa.

Si fissa l'attenzione sul calcolo di $\lg_2(X)$.

SINTESI DEL CALCOLO

Si fattorizza X con il prodotto di fattori che appartengono a una successione convergente a 1.

In questo modo il $\lg_2(X)$ diventa la somma di logaritmi sempre più piccoli quanto più l'argomento si avvicina a 1. Questa complicazione del problema, il calcolo di un logaritmo trasformato in una somma di logaritmi (da calcolare), diventa una soluzione se l'elemento "i^{mo}" della successione di

logaritmi ha una composizione particolare fatta da una cifra binaria (0 o 1) e un fattore esponenziale del tipo 2^{-i} , almeno quando la cifra binaria è 1; in questo modo la cifra binaria diventa la i^{ma} cifra del $\lg_2(X)$.

Si ricava che i termini della successione dei fattori di X devono avere una forma del tipo $2^{(2^{-i})}$ il cui logaritmo in base 2 è 2^{-i} richiesto.

SVOLGIMENTO

Si introducono le seguenti tre successioni, la seconda e la terza delle quali sono interlacciate tra loro e dipendono dalla prima che è autonoma:

$$\{b_n\} \equiv \begin{cases} b_0 = 2 \\ b_n = 2^{(2^{-n})} = 2^{(1/2^n)} \end{cases} \equiv \begin{cases} b_0 = 2 \\ b_n = \sqrt{b_{n-1}} \end{cases} \quad \{x_n\} \equiv \begin{cases} x_0 = X \\ x_n = \frac{x_{n-1}}{a_{n-1}} \end{cases} \quad \{a_n\} \equiv \begin{cases} 1 & \text{se } b_n > x_n \\ b_n & \text{se } b_n \leq x_n \end{cases}$$

La successione $\{a_n\}$ è una fattorizzazione di X.

Infatti, siano noti gli a_n da 0 fino a un k quanto si vuole grande e sia anche noto x_k , allora procedendo a ritroso sulla successione $\{x_n\}$ a partire da x_k :

$$\begin{aligned} x_{k-1} &= a_{k-1} * x_k & x_{k-2} &= a_{k-2} * a_{k-1} * x_k & x_{k-3} &= a_{k-3} * a_{k-2} * a_{k-1} * x_k & \dots \dots \\ \dots \dots & & x_{k-k} &= x_0 = X = a_0 * a_1 * a_2 * \dots \dots * a_{k-2} * a_{k-1} * x_k \end{aligned}$$

Per verificare le altre caratteristiche, necessarie, per il calcolo in questione, si comincia con il constatare che:

$$\lim_{n \rightarrow \infty} b_n = 1$$

Infatti, se il limite scritto è vero, $\forall \epsilon$ reale positivo deve esistere un $v(\epsilon)$ reale dipendente da ϵ tale che per $\forall n > v(\epsilon)$ e n intero

$$|b_n - 1| < \epsilon \quad |2^{(2^{-n})} - 1| < \epsilon$$

Le potenze con base maggiore di uno, nel nostro caso 2, e esponente positivo sono sempre maggiori di uno; pertanto:

$$\begin{aligned} 2^{(2^{-n})} - 1 < \epsilon &\rightarrow \frac{1}{2^n} < \lg_2(1 + \epsilon) \rightarrow 2^n > \frac{1}{\lg_2(1 + \epsilon)} \rightarrow \\ \rightarrow n > \lg_2 \left[\frac{1}{\lg_2(1 + \epsilon)} \right] &= -\lg_2[\lg_2(1 + \epsilon)] = v(\epsilon) \end{aligned}$$

Quindi esiste $v(\epsilon)$ tale che per gli interi più grandi di esso la successione si discosta da 1 meno dell' ϵ scelto.

Si osserva che se $\epsilon < 1$ $\lg_2[\lg_2(1 + \epsilon)]$ diventa negativo e $v(\epsilon)$ diventa positivo.

Non solo la successione $\{b_n\}$ converge a 1 ma anche il $\lim_{n \rightarrow \infty} a_n = 1$ e $\lim_{n \rightarrow \infty} x_n = 1$

Per la prima si costata che $1 \leq a_n \leq b_n$, anzi è l'uno o l'altro valore di contenimento; cioè:

$\{a_n\}$ è sempre compresa tra $\{1\}$ e $\{b_n\}$ entrambe convergenti a 1, quindi $\{a_n\}$ converge a 1.

L'altra successione, $\{x_n\}$, ha il seguente andamento:

$1 \leq x_0 < 2$ $b_0 = 2$ inoltre per $k=1,2,3,\dots,i$ con $b_k > x_k$ per $k < i$ e $b_i \leq x_i$ x_k resta costante in quanto

$$a_{k-1} = 1 \text{ poi per } k = i + 1 \quad x_{i+1} = \frac{x_i}{a_i} = \frac{x_i}{b_i} \neq x_0$$

in dettaglio:

$$x_0 < b_0 = 2$$

$$x_1 = x_0 < b_1$$

$$x_2 = x_1 = x_0 < b_2$$

⋮

⋮

$$x_{i-1} = x_{i-2} = \dots = x_0 < b_{i-1}$$

$$x_i = x_{i-1} = \dots = x_0 \geq b_i \quad \text{in questo caso} \quad b_i \leq x_i = \dots = x_0 < b_{i-1}$$

$$\text{ma } \sqrt{b_{i-1}} = b_i \rightarrow b_{i-1} = b_i^2 \quad \text{cioè} \quad b_i \leq x_i = \dots = x_0 < b_i^2$$

$$\text{poi } x_{i+1} = \frac{x_i}{b_i} \text{ e } b_i \leq x_i < b_i^2 \rightarrow \frac{b_i}{b_i} \leq \frac{x_i}{b_i} < \frac{b_i^2}{b_i} \equiv 1 \leq \frac{x_i}{b_i} < b_i \rightarrow 1 \leq x_{i+1} < b_i$$

Quindi $\{x_{i+1}\}$ è compresa tra $\{1\}$ e $\{b_i\}$ entrambe convergenti a 1, pertanto anche $\{x_{i+1}\}$ converge a 1.

In ridondanza, ricordando che $b_i = b_{i+1}^2 \rightarrow 1 \leq x_{i+1} < b_{i+1}^2$
resta da dimostrare che anche $\{b_n^2\}$ converge a 1.

$$\lim_{n \rightarrow \infty} b_n^2 = \lim_{n \rightarrow \infty} (2^{(2^{-n})})^2 = \lim_{n \rightarrow \infty} 2^{(2^{-(n-1)})} = \lim_{n \rightarrow \infty} b_{n-1} = \lim_{n \rightarrow \infty} b_n = 1$$

In definitiva:

$$\lim_{n \rightarrow \infty} x_n = 1$$

Quanto provato permette di affermare che:

se $1 \leq x_0 < 2$ comunque si sceglie un reale $\varepsilon > 0$ esiste un intero $v(\varepsilon)$ tale che per ogni $k > v(\varepsilon)$

lo sviluppo: $X = x_0 = a_0 * a_1 * a_2 * \dots * a_{k-1} * x_k$ ha x_k che verifica $|x_k - 1| < \varepsilon$

In definitiva:

$$\lg_2(x_0) = \sum_{i=0}^{k-1} \lg_2(a_i) + R_k$$

con $R_k = \lg_2(x_k)$ il quale in modulo può diventare, scegliendo k , piccolo quanto si vuole in quanto x_k si "approssima" a 1 quanto si vuole e il suo logaritmo in egual modo si "approssima" a zero.

Un esame della relazione:

$$\lg_2(X) \cong \sum_{i=0}^{k-1} \lg_2(a_i)$$

Ricordando che:

$$a_i = \begin{cases} 1 & \text{se } b_i > x_i \\ b_i = 2^{(2^{-i})} & \text{se } b_i \leq x_i \end{cases} \quad \lg_2(a_i) = \begin{cases} 0 & \text{se } b_i > x_i \\ 2^{-i} & \text{se } b_i \leq x_i \end{cases} \quad \text{con } x_i = \frac{x_{i-1}}{a_{i-1}}$$

equivale a dire che introdotta la cifra Y_i nel seguente modo:

$$Y_i = \begin{cases} 0 & \text{se } b_i > x_i \\ 1 & \text{se } b_i \leq x_i \end{cases}$$

il $\lg_2(X) \cong Y_0 Y_1 Y_2 Y_3 \dots \dots Y_k$

La stringa di cifre binarie $Y_0 Y_1 Y_2 Y_3 \dots \dots Y_0$ è la sequenza di 0 e 1 del logaritmo in base 2 di X .

In questo modo il logaritmo è ridotto al calcolo di radici quadrate successive di 2, in genere l'operazione di radice è implementata in macchina, e di comparazioni.

Il numero di radici successive di 2, necessarie, è pari al numero di cifre della MANTISSA X , o se si vuole al numero delle cifre significative che la macchina riserva per le MANTISSE, sono un numero limitato di valori e possono calcolarsi una volta per tutte e sistamarle in macchina al momento del caricamento del programma che usa la procedura del logaritmo.