

ESAME CALCOLATORI ELETTRONICI

appello 6 luglio 2006

Un'interfaccia IFCMP opera come *coprocesore* per l'esecuzione di moltiplicazioni tra numeri complessi, rappresentati come coppie ordinate di numeri reali floating-point a 32 bit. L'interfaccia riceve dalla CPU PD-32 due operandi complessi $X = X_R + jX_I$, $Y = Y_R + jY_I$, dove $j = \sqrt{-1}$ è l'unità immaginaria, e trasmette alla CPU il risultato

$$Z = Z_R + jZ_I = (X_R Y_R - X_I Y_I) + j(X_R Y_I + X_I Y_R)$$

L'hardware dell'interfaccia utilizza **un singolo** modulo sequenziale FPMA (*Floating-Point Multiply/Add*, Fig. 1) per l'esecuzione dell'operazione $R = A \pm BC$, dove A, B, C, R sono numeri **reali** floating-point a 32 bit; oltre alla moltiplicazione, il modulo esegue una somma o una sottrazione a seconda che il suo ingresso ADD sia attivo o meno; l'inizio e il termine delle operazioni del modulo sono determinati dai segnali presenti rispettivamente all'ingresso BEGIN e all'uscita END (Fig. 2).

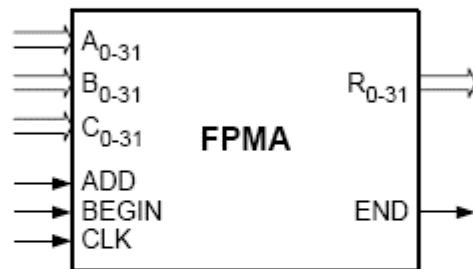


Fig. 1

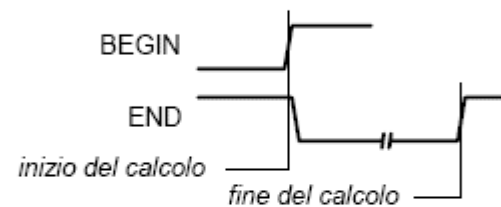


Fig. 2

Il software di pilotaggio è organizzato sotto forma di subroutine che riceve in ingresso:

- un intero $N > 0$;
- i puntatori a due tavole XTABLE, YTABLE, che contengono rispettivamente N numeri complessi $X[i]$ ed N numeri complessi $Y[i]$;
- il puntatore a una tavola ZTABLE nella quale, dopo l'esecuzione dei calcoli necessari, dovranno essere immagazzinati gli N risultati $Z[i] = X[i] \cdot Y[i]$.

Progettare l'hardware dell'interfaccia e codificare la relativa subroutine di pilotaggio, corredandola di adeguati commenti.

NOTE:

1. Il modulo FPMA **non** deve essere progettato.
2. Il valore 0, qualora necessario, è rappresentato in floating-point come 32 bit tutti uguali a zero.

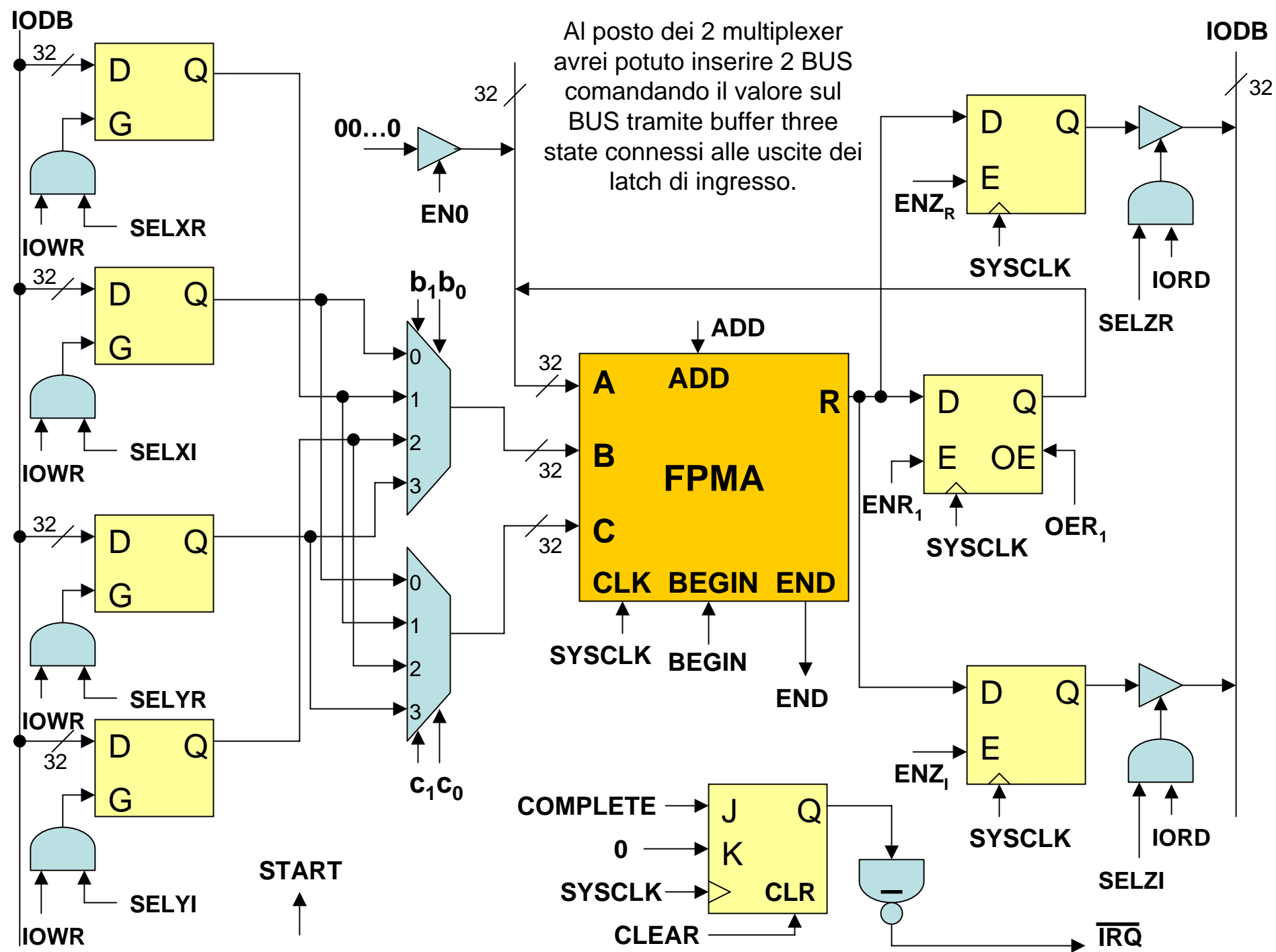
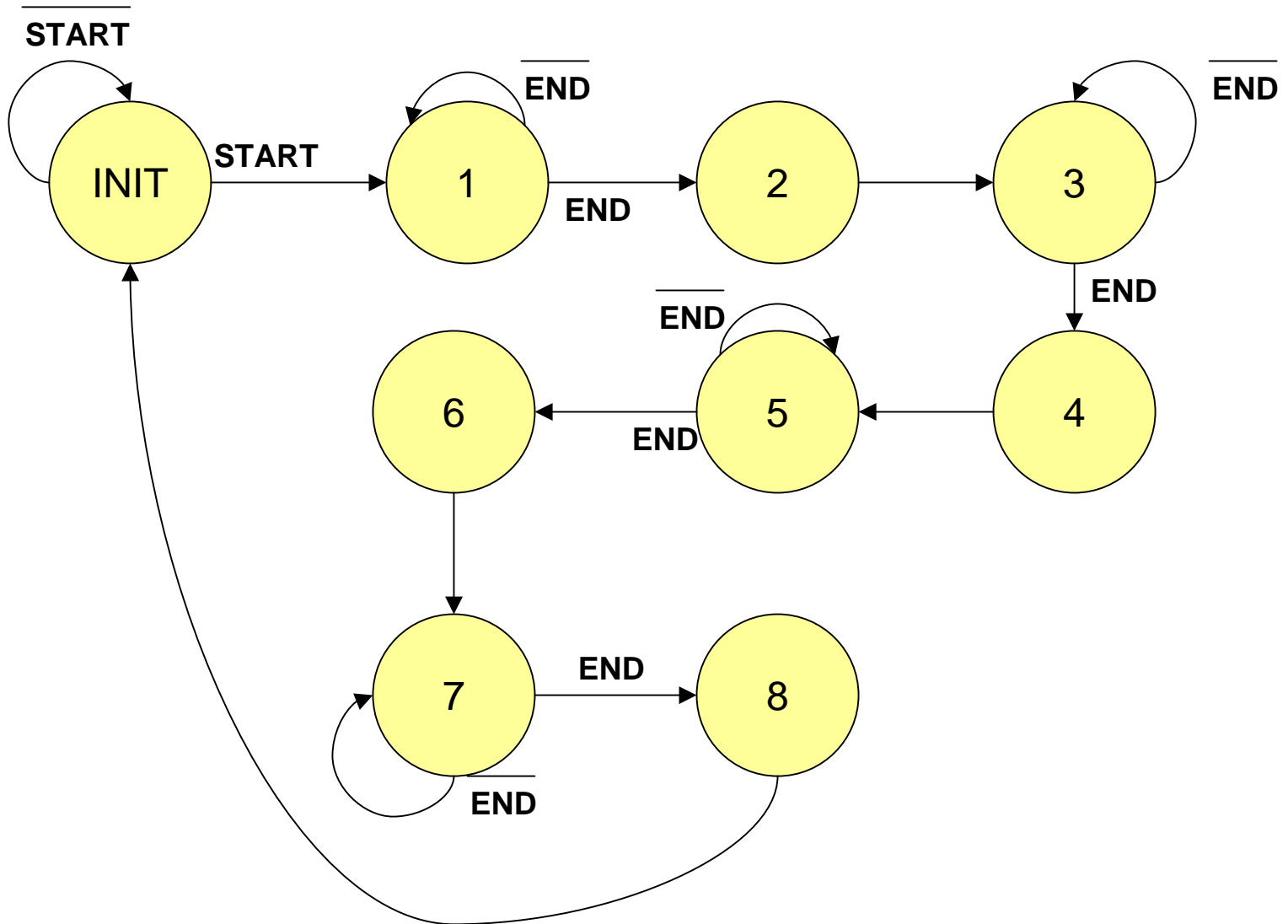


Diagramma di stato



[INIT] : tutti i segnali a zero e i segnali di controllo del MUX congelati allo stato 8

[STATO 1]:

$EN0 = 1, b_1b_0 = 01, c_1c_0 = 11, ADD = 1, BEGIN = 1$

[STATO 2]:

$EN0 = 1, b_1b_0 = 01, c_1c_0 = 11, ADD = 1, BEGIN = 0, ENR_1 = 1$

[STATO 3]:

$EN0 = 0, b_1b_0 = 00, c_1c_0 = 10, ADD = 0, BEGIN = 1, ENR_1 = 0, OER_1 = 1$

[STATO 4]:

$EN0 = 0, b_1b_0 = 00, c_1c_0 = 10, ADD = 0, BEGIN = 0, ENR_1 = 0, OER_1 = 1, ENZ_R = 1$

[STATO 5]:

$EN0 = 1, b_1b_0 = 01, c_1c_0 = 10, ADD = 1, BEGIN = 1, ENR_1 = 0, OER_1 = 0, ENZ_R = 0$

[STATO 6]:

$EN0 = 1, b_1b_0 = 01, c_1c_0 = 10, ADD = 1, BEGIN = 0, ENR_1 = 1, OER_1 = 0, ENZ_R = 0$

[STATO 7]:

$EN0 = 0, b_1b_0 = 00, c_1c_0 = 01, ADD = 1, BEGIN = 1, ENR_1 = 0, OER_1 = 1, ENZ_R = 0$

[STATO 8]:

$EN0 = 0, b_1b_0 = 00, c_1c_0 = 01, ADD = 1, BEGIN = 0, ENR_1 = 0, OER_1 = 0, ENZ_R = 0, ENZ_i = 1, COMPLETE = 1$

