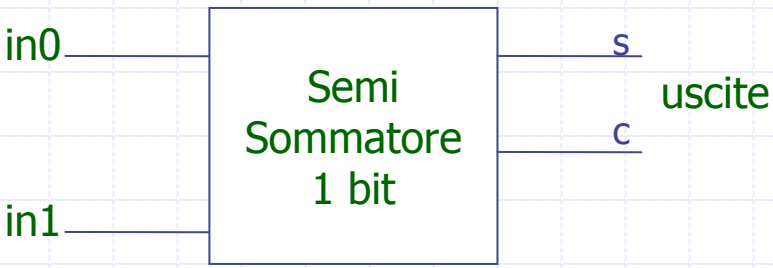
PSPICE – simulazione sommatori, comparatori

Davide Piccolo

Il *SOMMATORE*

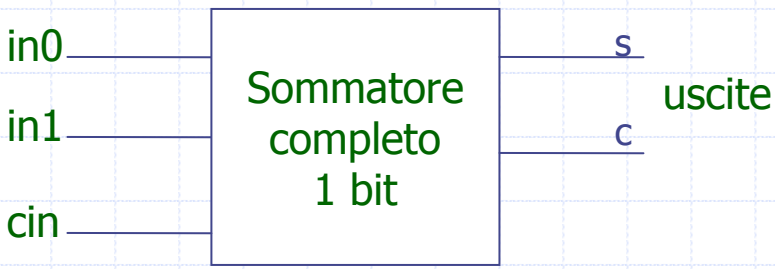
- Il *sommatore* è un circuito che effettua la soma bit per bit di cifre binarie. Può essere definito su di un numero arbitrario N di bit.
- Si distingue in *semisommatore* e *sommatore completo* a seconda se contiene o meno il trattamento del "riporto" (Carry flag)



In1	In0		S	C
0	0		0	0
0	1		1	0
1	0		1	0
1	1		0	1

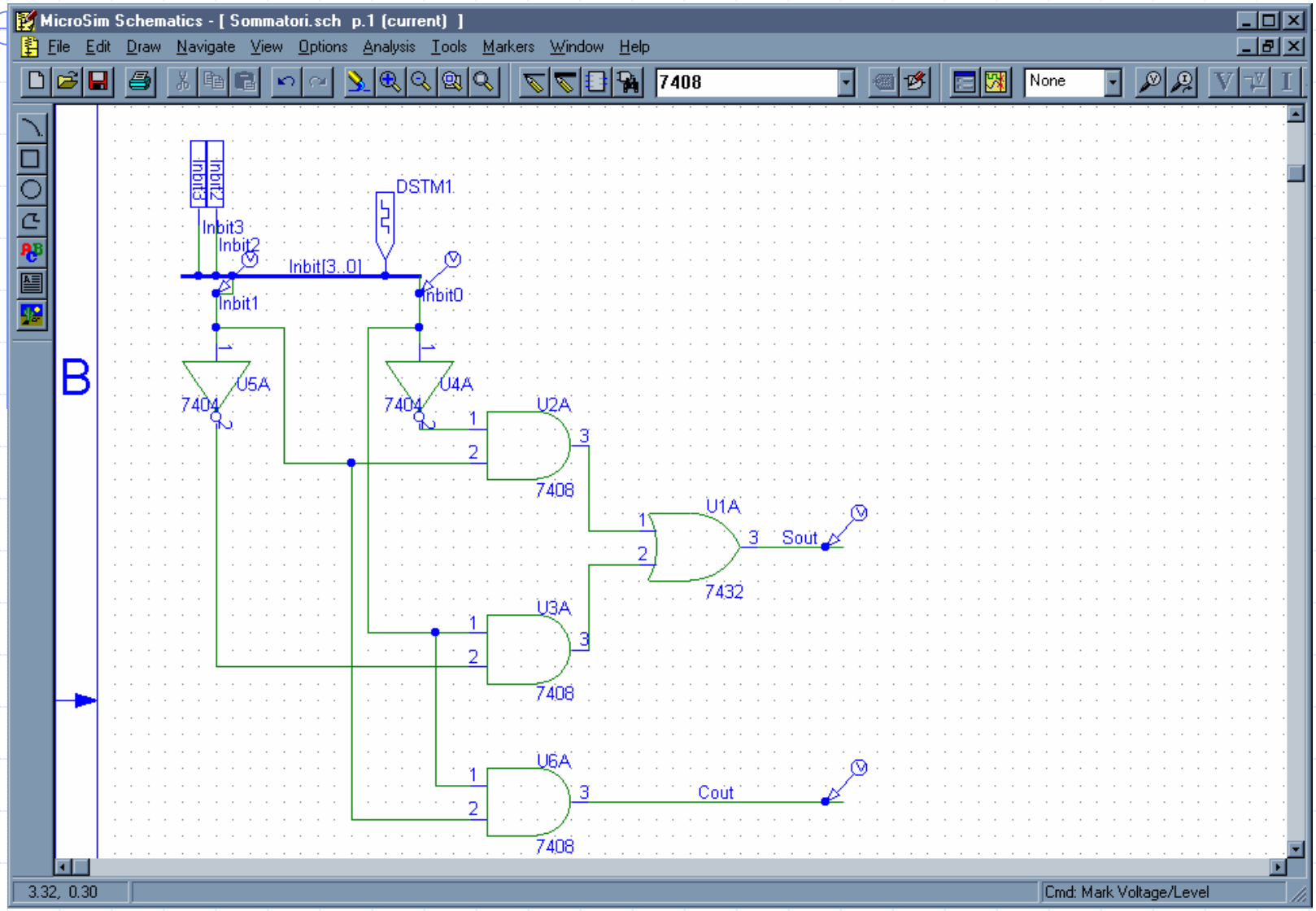
Il *SOMMATORE completo*

- Il *sommatore* completo deve tener conto del *carry flag* che gli arriva dagli stadi di somma precedenti
- Solo i bit *meno significativi* possono essere sommati con un semisommatore.

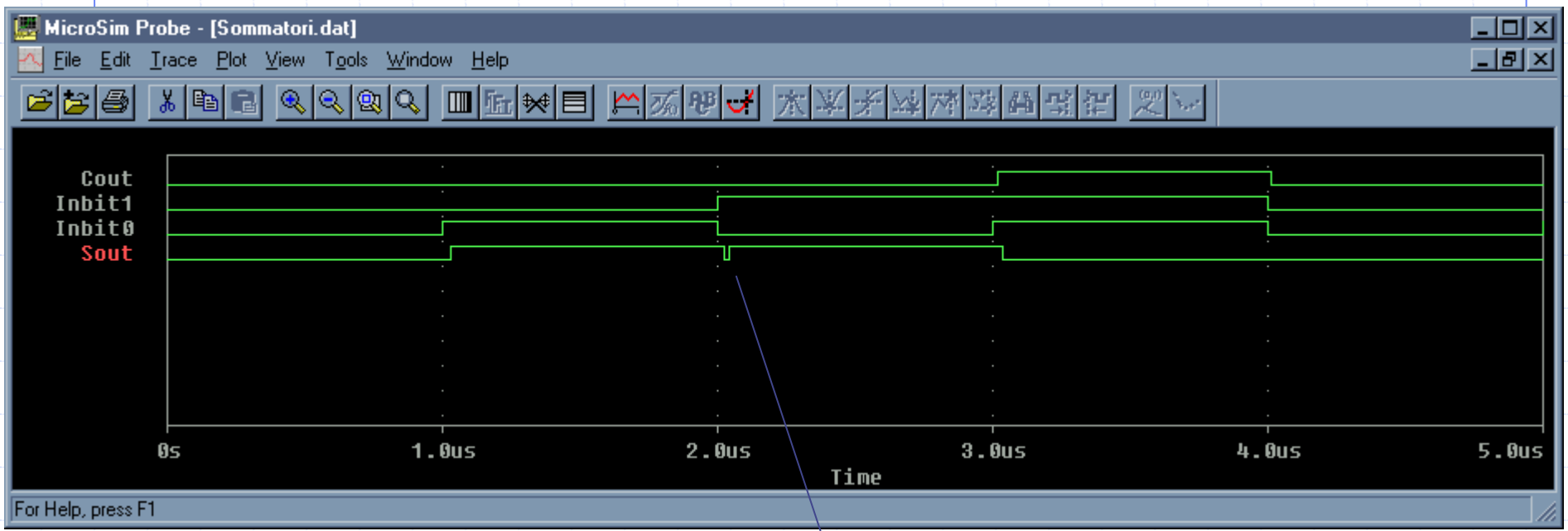


In1	In0	Cin		S	C
0	0	0		0	0
0	1	0		1	0
1	0	0		1	0
1	1	0		0	1
0	0	1		1	0
0	1	1		0	1
1	0	1		0	1
1	1	1		1	1

Implementazione Del semisommatore

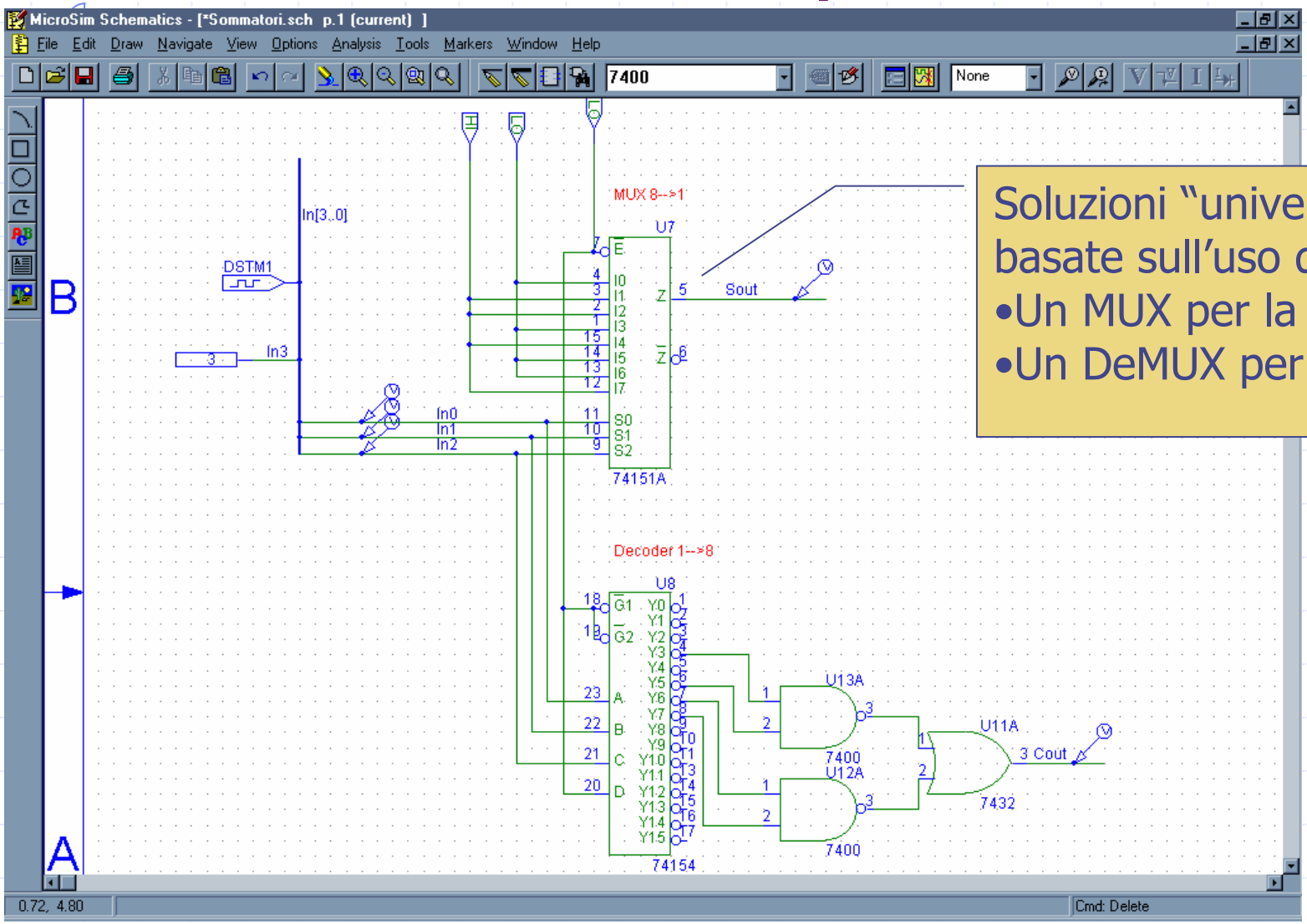


Risultato Della Simulazione



Da notare la spike derivante dai ritardi relativi…

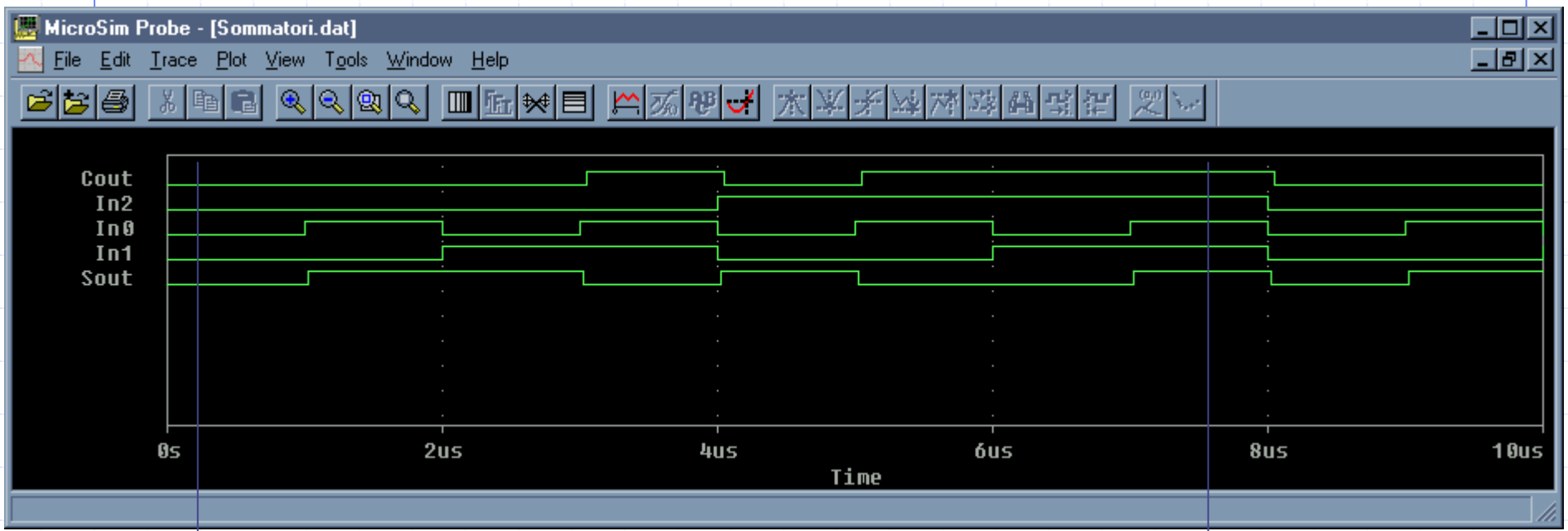
Implementazione Del sommatore Completo



Soluzioni "universali" basate sull'uso di:

- Un MUX per la somma
- Un DeMUX per il Carry

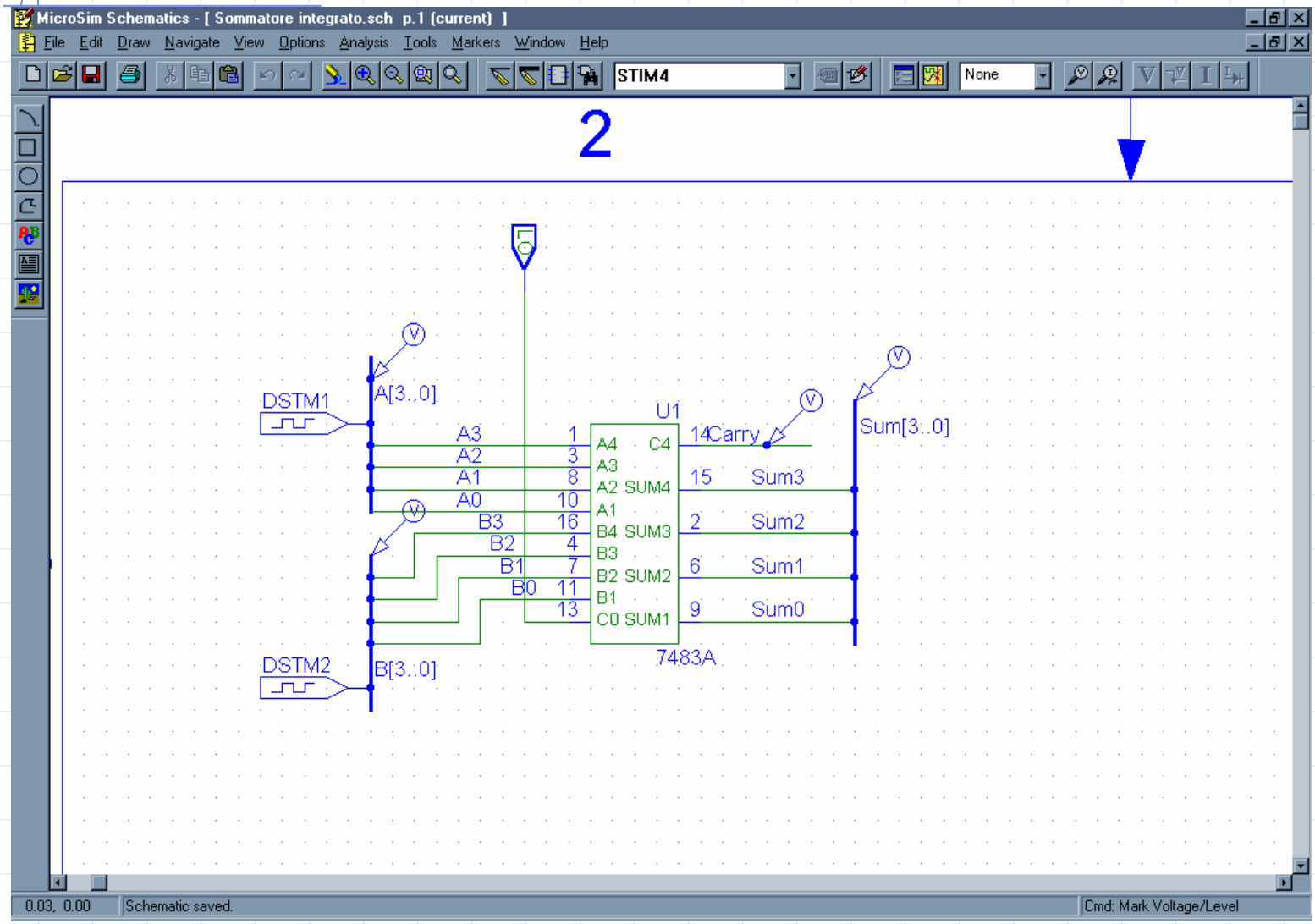
Risultato Della Simulazione



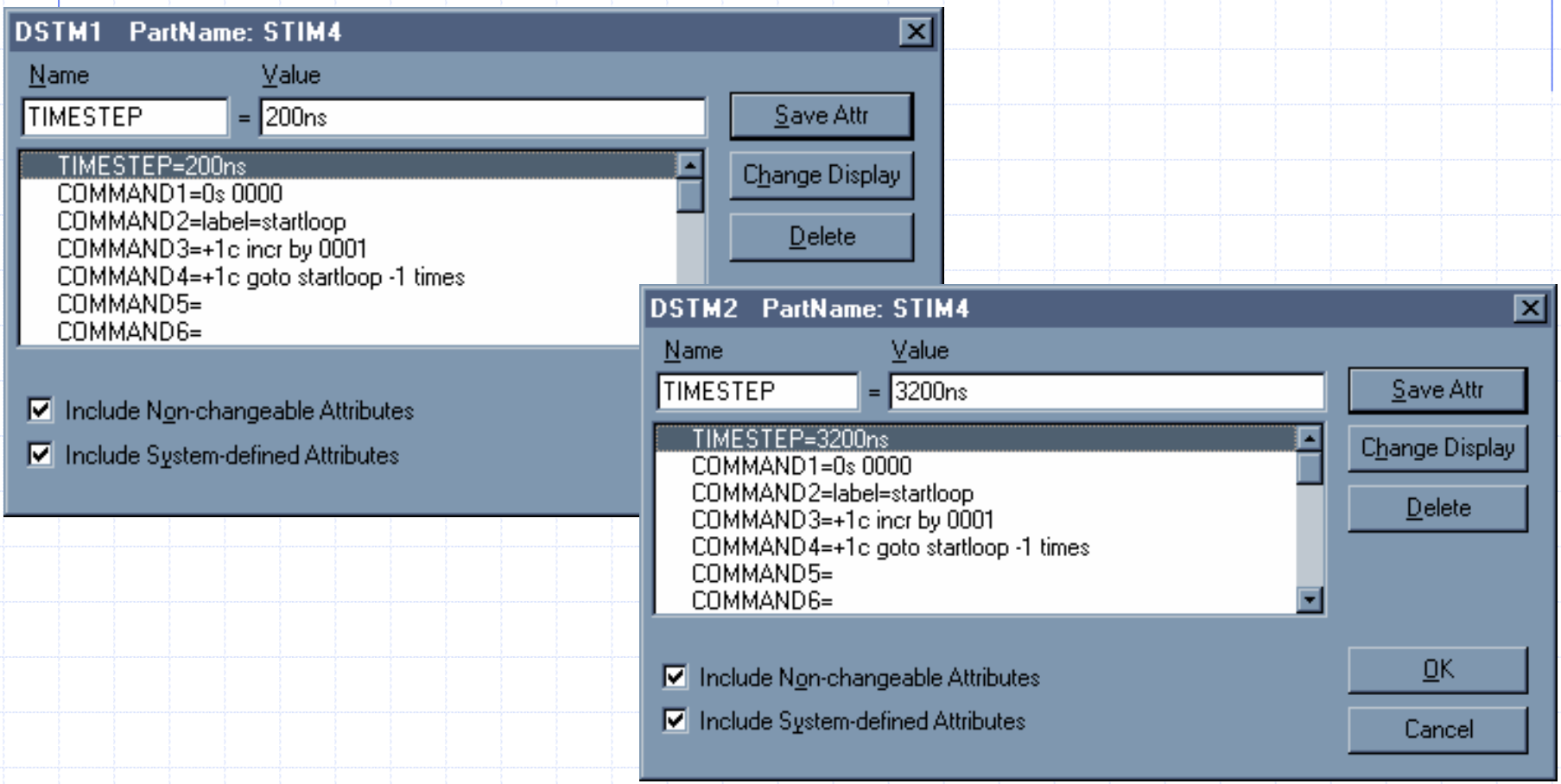
In=0,0
Cin=0

In=1,1
Cin=1

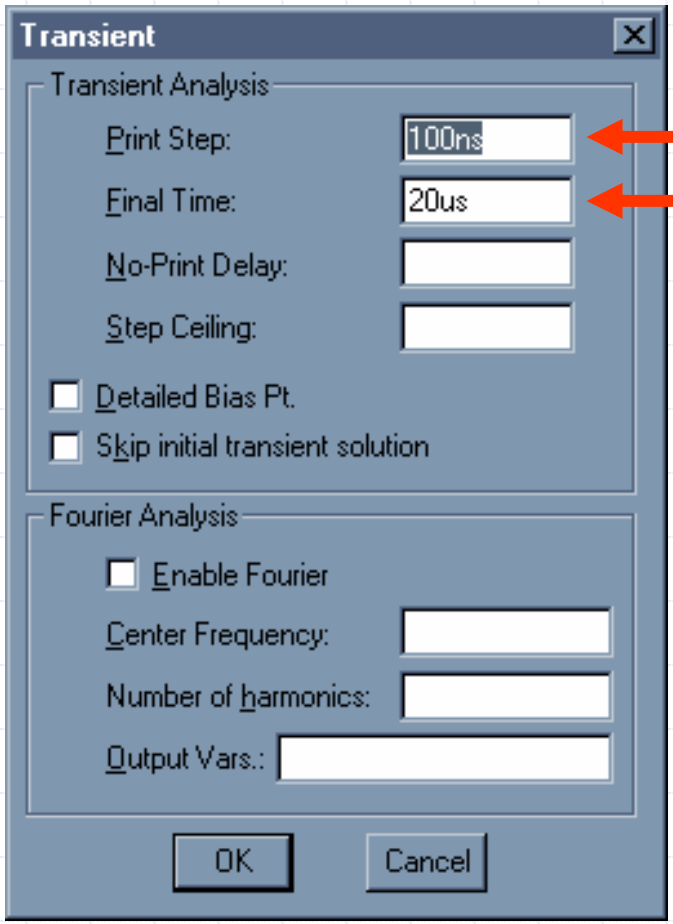
Sommatore integrato



Stimoli opportuni...

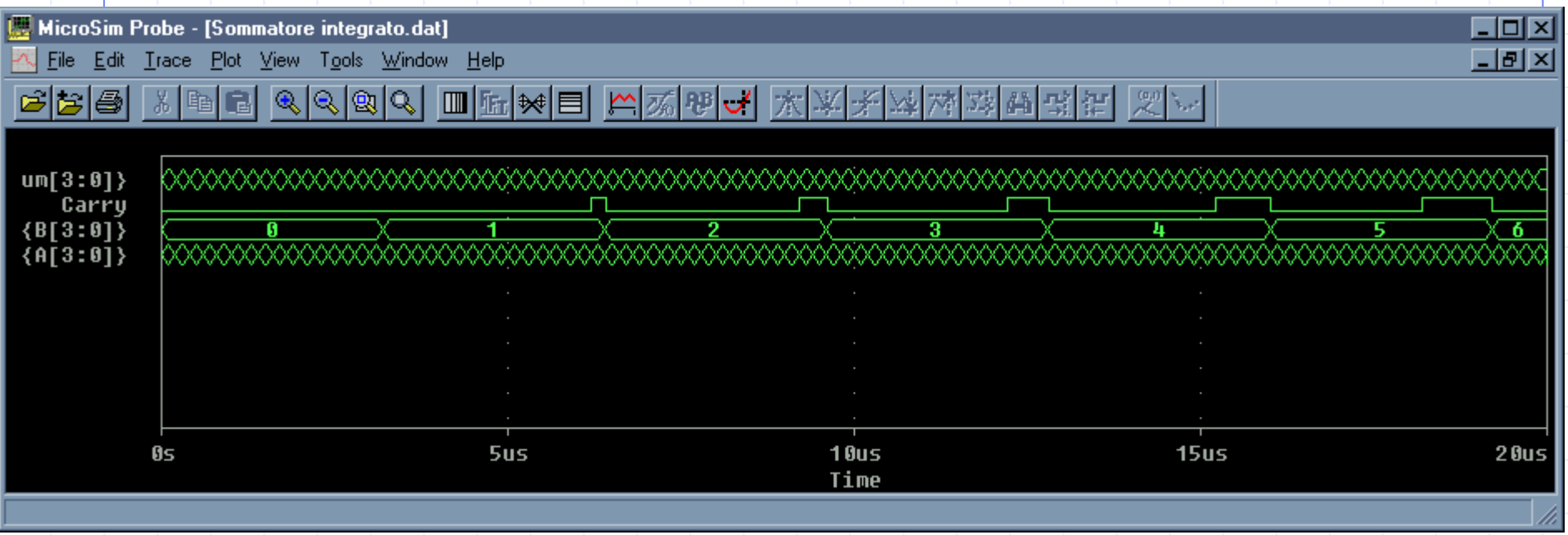


Tempo totale di simulazione



20us/100ns =
20000/100 =200 passi

Risultato della simulazione



I comparatori numerici

- I comparatori sono dei circuiti combinatori in grado di confrontare tra di loro due numeri binari

- All'ingresso del circuito arrivano i bit dei due numeri da confrontare e in uscita solo una di tre linee va allo stato logico alto a seconda che il primo numero in ingresso sia uguale, maggiore o minore del secondo numero in ingresso.

Mappe di karnaugh del comparatore a due bit

a = b

b2b1 \ a2a1	00	01	10	11
00	1	0	0	0
01	0	1	0	0
10	0	0	1	0
11	0	0	0	1

a > b

b2b1 \ a2a1	00	01	10	11
00	0	1	1	1
01	0	0	1	1
10	0	0	0	1
11	0	0	0	0

a < b

b2b1 \ a2a1	00	01	10	11
00	0	0	0	0
01	1	0	0	0
10	1	1	0	0
11	1	1	1	0

A = B

$$Y = \overline{A_1}\,\overline{B_1}\,\overline{A_2}\,\overline{B_2} + \overline{A_1}\,\overline{B_1}A_2B_2 + A_1B_1\overline{A_2}\,\overline{B_2} + A_1B_1A_2B_2 =$$

$$\overline{A_1}\,\overline{B_1}(\overline{A_2}\,\overline{B_2}+A_2B_2)+A_1B_1(\overline{A_2}\,\overline{B_2}+A_2B_2)$$

$$= (\overline{A_1}\,\overline{B_1}+ A_1B_1)\,(\overline{A_2}\,\overline{B_2}+A_2B_2) =$$

$$\overline{A_1 \oplus B_1} \bullet \overline{A_2 \oplus B_2}$$

A > B

$$Y = A_1\overline{A_2}\,\overline{B_1}\,\overline{B_2} + A_1\overline{A_2}B_1B_2 + \overline{A_2}\,\overline{B_2}$$

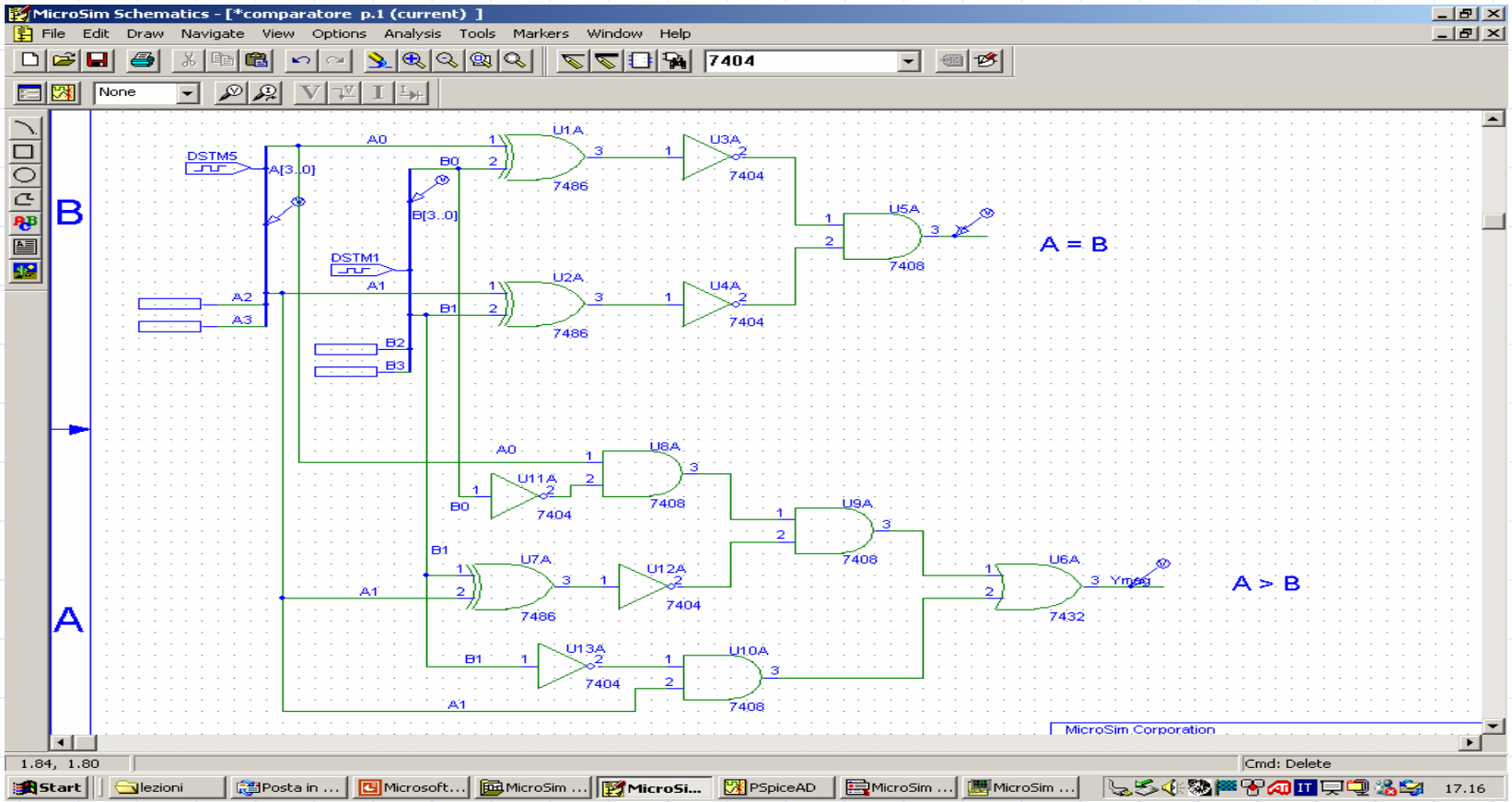
$$= A_1\overline{B_1}\bullet\overline{(A_2\oplus B_2)} + \overline{A_2}\,\overline{B_2}$$

A < B

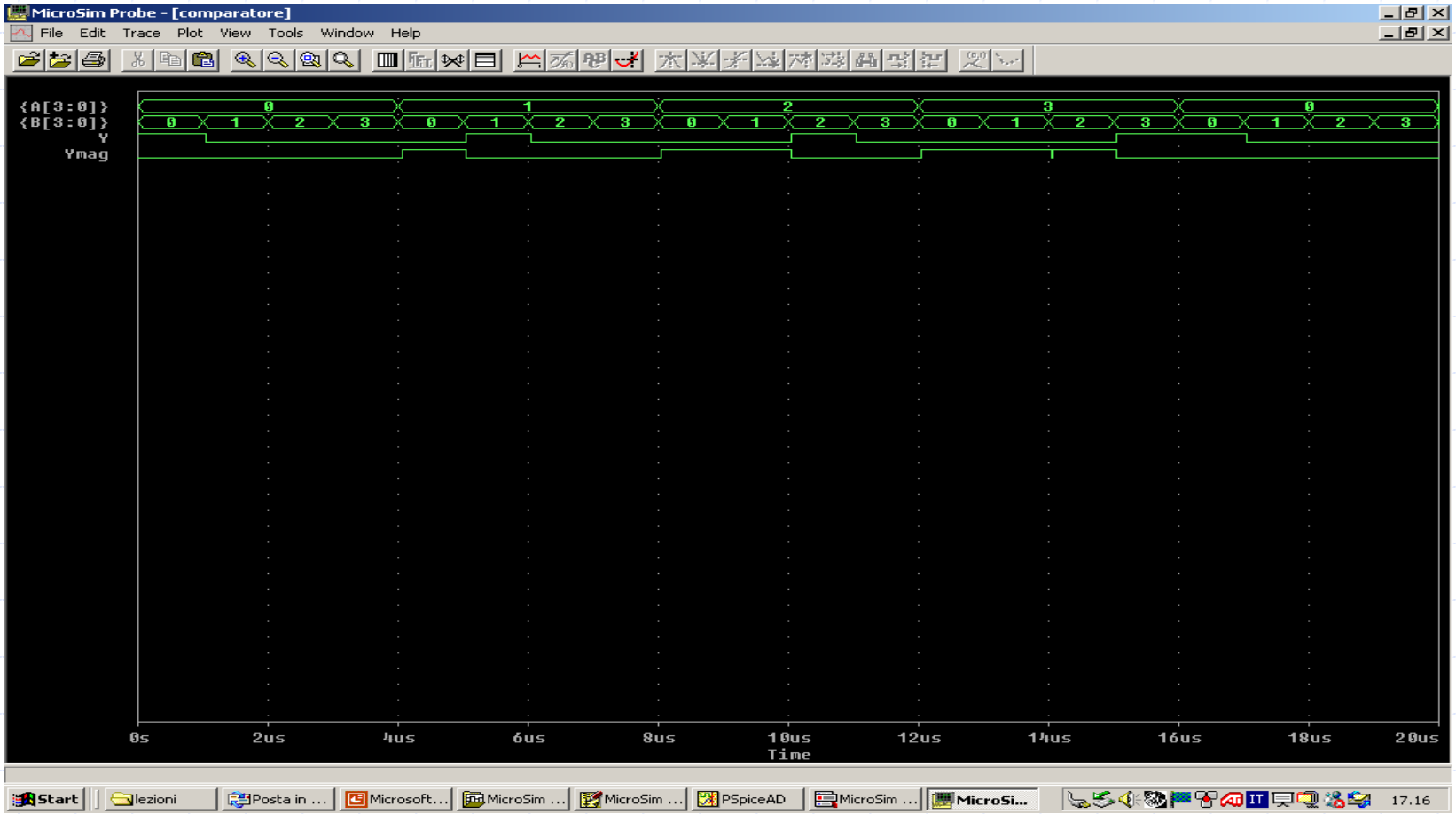
$$Y = \overline{A_1}A_2\overline{B_1}\,\overline{B_2} + \overline{A_1}A_2B_1B_2 + \overline{A_2}B_2$$

$$= \overline{A_1}B_1\bullet\overline{(A_2\oplus B_2)} + \overline{A_2}B_2$$

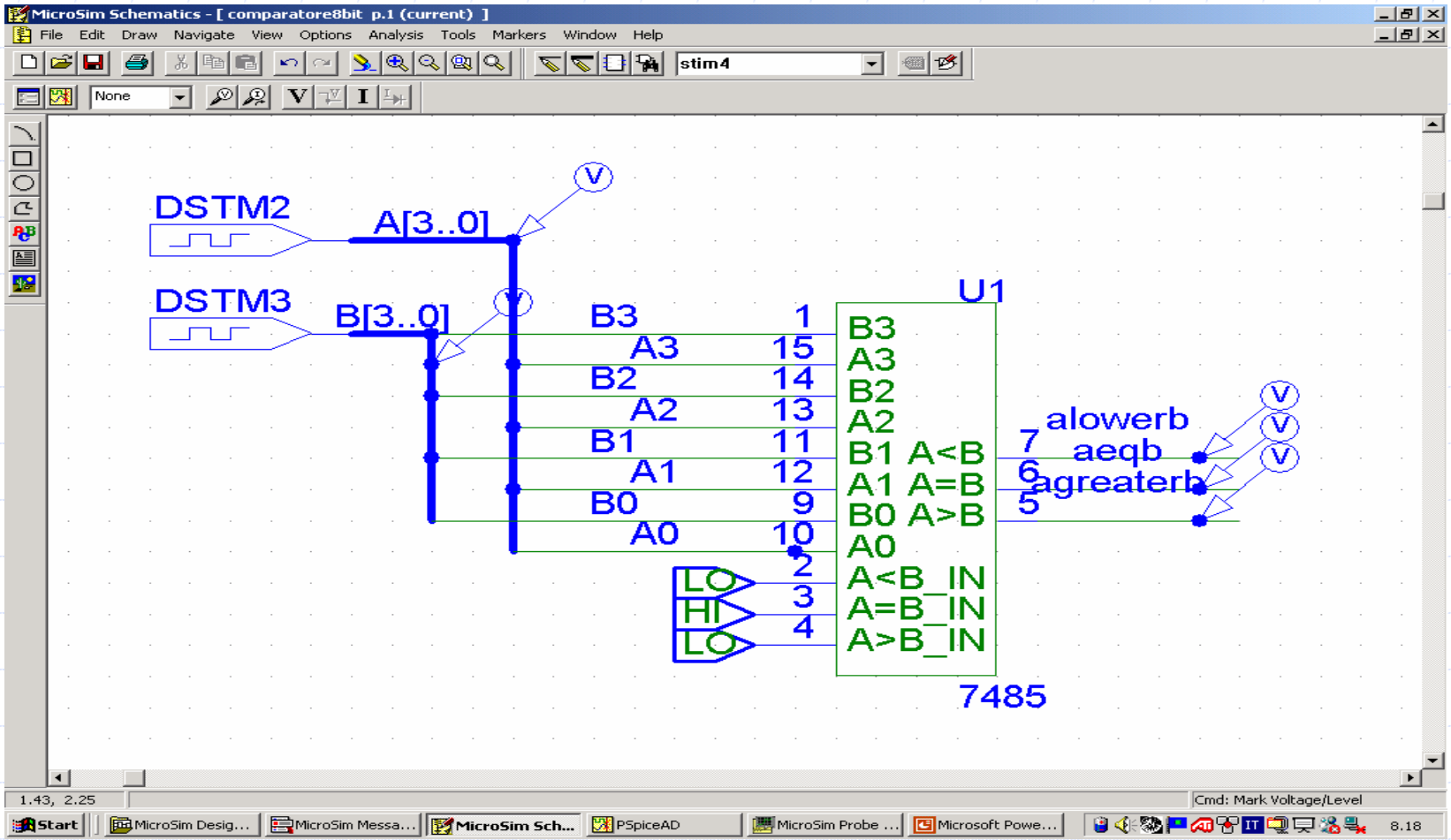
Schematico Comparatore



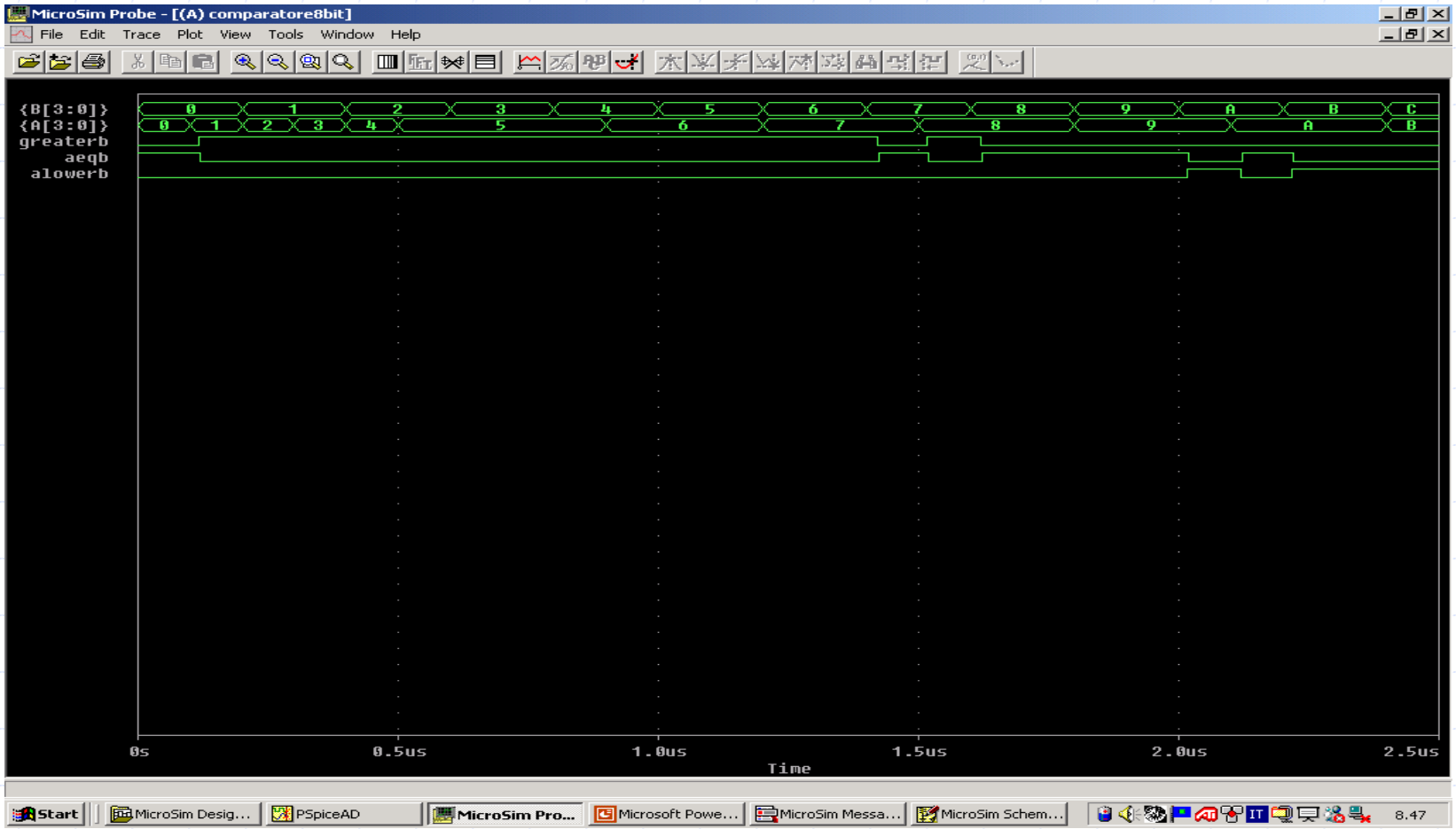
Simulazione del circuito



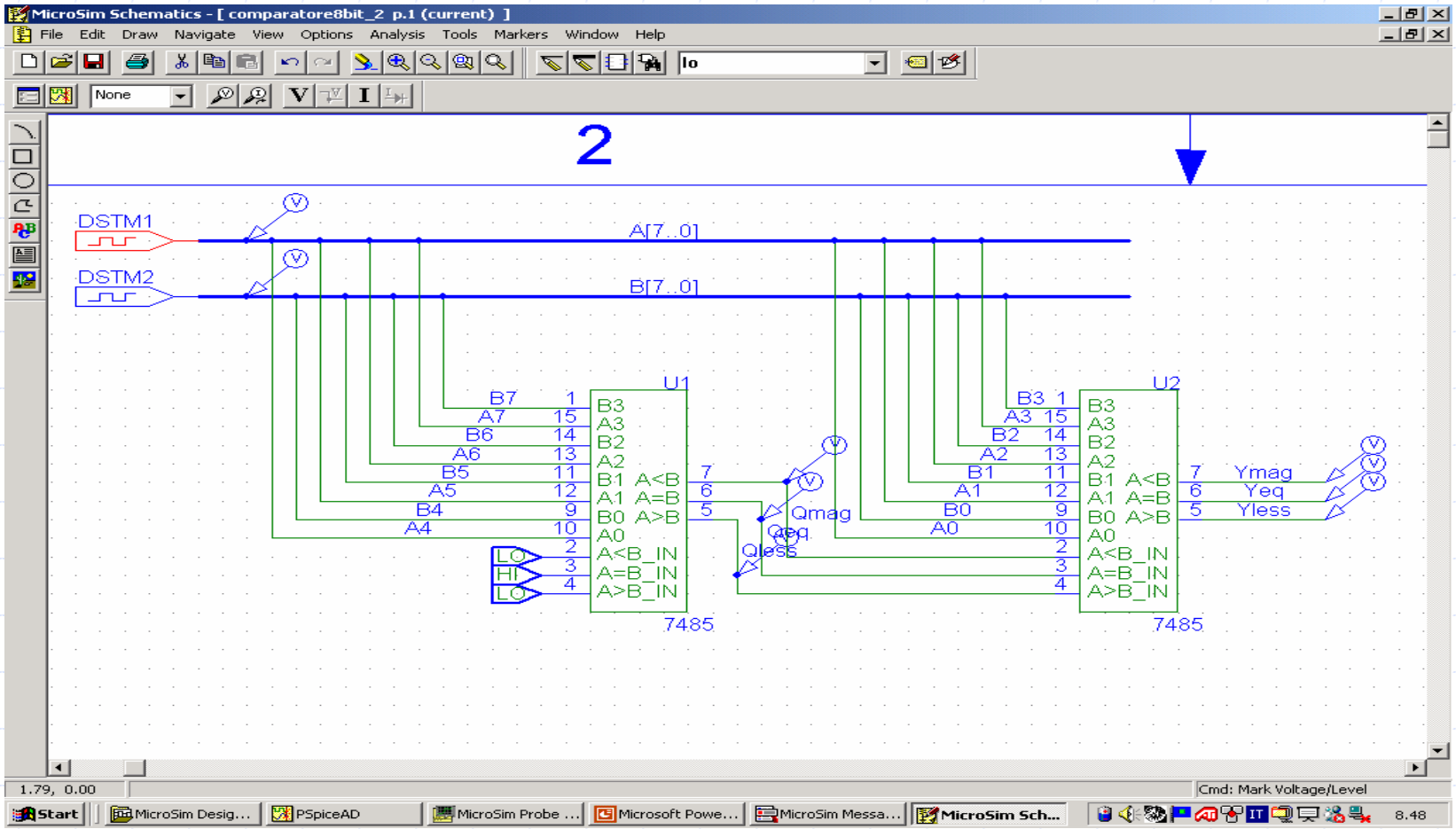
Comparatore a 4 bit (integrato 7485)



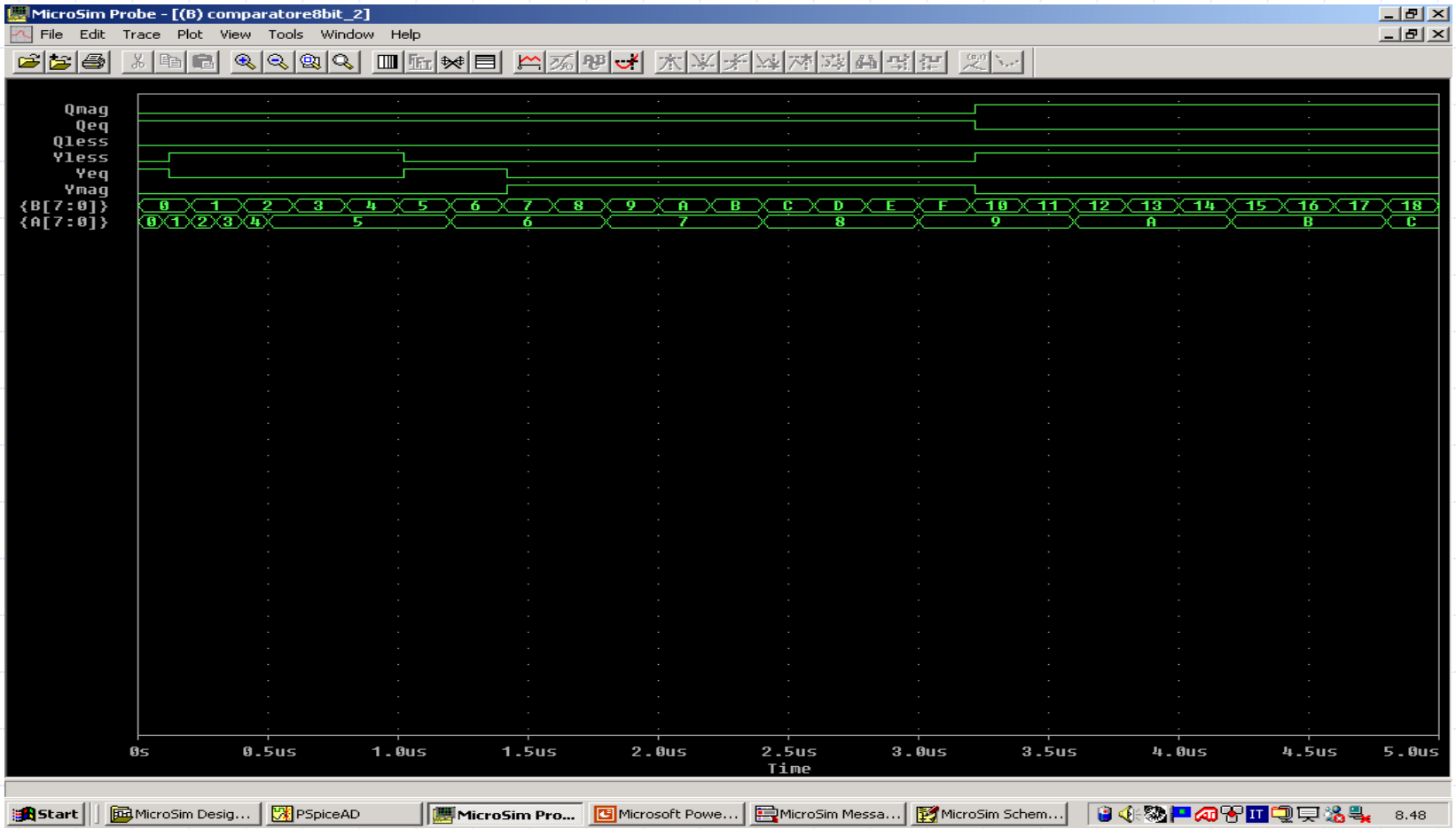
Simulazione circuito



Comparatore a 8 bit



Simulazione del circuito

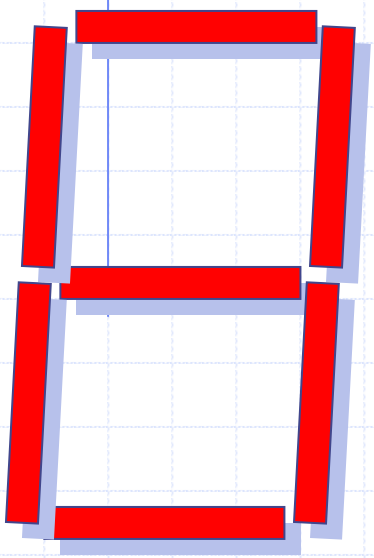


Decodifica per display 7 segmenti

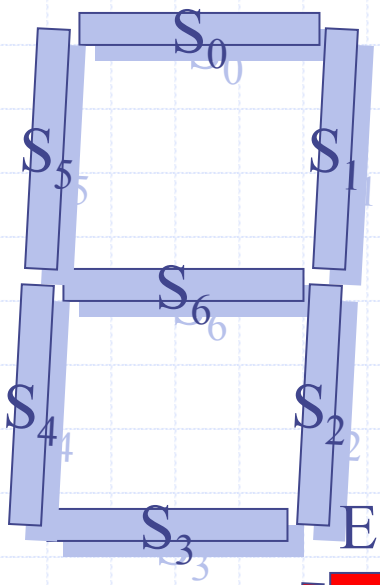
- Da codice BCD a pilotaggio segmenti
  - A catodo comune (accesi se H)
  - Ad anodo comune (accesi se L)

Decodifica per display LED/LCD: tabella della verità

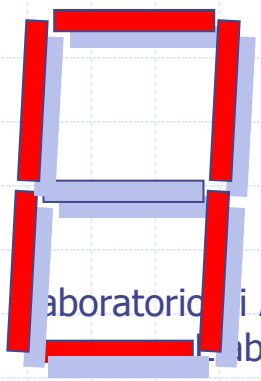
Tutto ON



Tutto OFF



Es. '0'



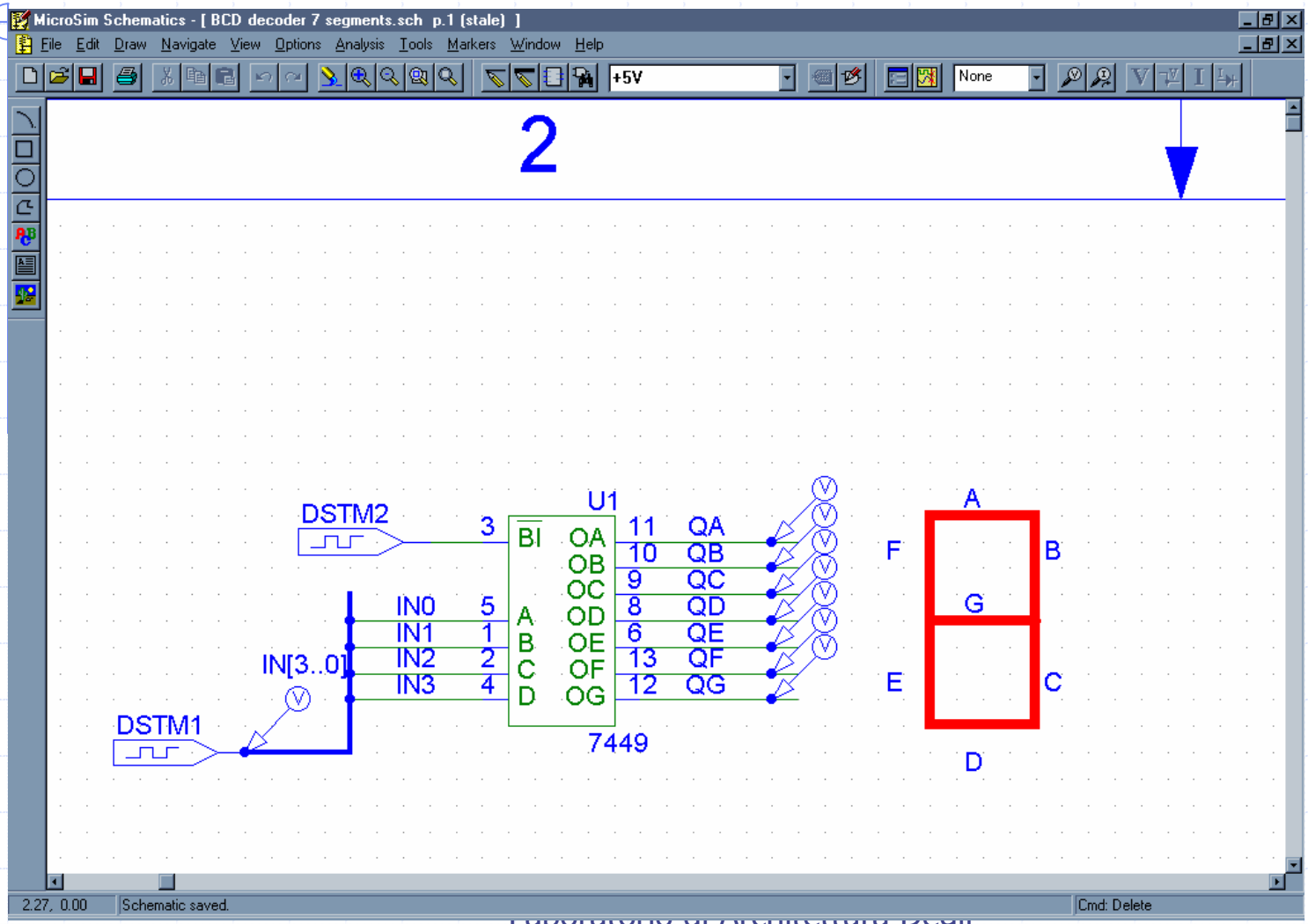
- A catodo comune (H)

Code	S_0	S_1	S_2	S_3	S_4	S_5	S_6	
0	H	H	H	H	H	H	L	
1	L	H	H	L	L	L	L	
2	H	H	L	H	H	L	H	
3	H	H	H	H	L	L	H	

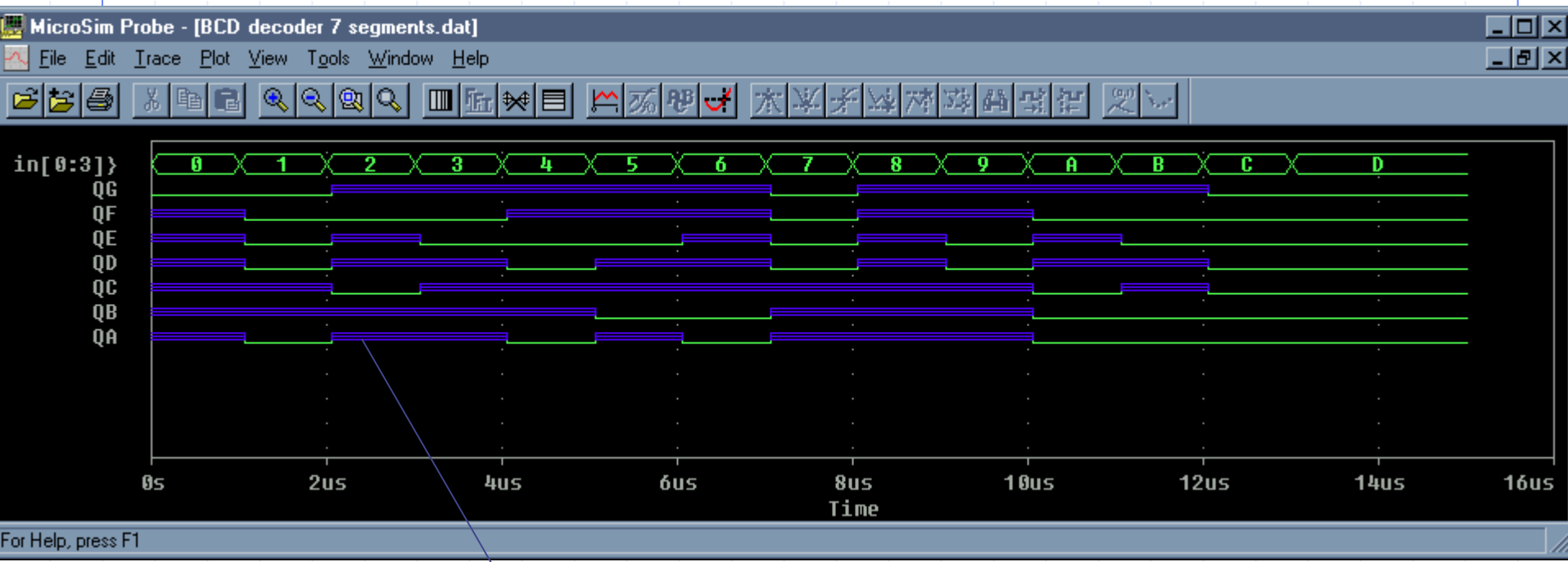
Logica della decodifica

- Si basa sulla semplice interpretazione della tabella della verità (in forma SOP o POS opportunamente minimizzata)
- Per la simulazione ci serviamo di un circuito preesistente (7449)

Lo schematic



Risultato di una simulazione



Da notare lo stato di alta impedenza dovuta ad uscite a collettore aperto mancanti di connessioni esterne

Esempi di esercizi di esame (I)

Si dispone di 4 pile, due di 1.5 V, un di 3 V ed una di 4.5 V. Associando lo stato logico 1 quando la pila è attivata e lo stato logico 0 quando non è attivata, progettare un circuito combinatorio che funga da sistema di controllo con le specifiche seguenti:

- Ha 4 uscite U0, U1, U2, U3
- Chiamando Vt la somma delle tensioni delle pile attivate, le uscite assumeranno i seguenti valori:
  - Vt>6 V U0=1
  - 4.5 V < Vt <= 6 V U1=1
  - 3V < Vt <= 4.5 V U2=1
  - 1.5 V <= Vt <= 3 V U3=1

Progettare il sistema con una rete basata su un decoder e simularne il funzionamento ricavandone la tabella di verità

Esempi di esercizi di esame (II)

Una bilancia ha una portata massima di 15 Kg e su di essa si possono trovare uno o più oggetti: A, B, C, D che pesano rispettivamente 7 Kg; 4 Kg; 3 Kg; 5 Kg. Associando lo stato logico 1 quando il peso è posto sulla bilancia, progettare un circuito digitale ad una uscita Y tale che Y=1 solo se il peso complessivo posto sulla bilancia supera la portata.

Realizzare il circuito mediante un multiplexer 4 a 1. Simulare e ricavare la tabella di verità completa del circuito.

Esempi di esercizi di esame (III)

Una compagnia di avventurieri si imbatte in un'idra. Per sconfiggerla il mago del gruppo (7 livello, modificatore di intelligenza = +7) può lanciare un incantesimo, il chierico (6 livello, modificatore di saggezza = +5) può lanciare un incantesimo "distruggere un essere vivente" mentre il ladro può usare l'oggetto "Freccia assassina superiore". A ciascun incantesimo e oggetto magico è associata una classe di difficoltà (CD) secondo la seguente tabella:

incantesimo = 10 + livello dell'incantatore+modificatore di caratteristica

oggetto = 17

Il gruppo può sferrare i suoi attacchi magici in una qualsiasi combinazione di incantesimi e oggetti magici. La CD complessiva è data dalla somma delle singole CD. Per difendersi, l'idra ha diritto ad un tiro salvezza che deve essere maggiore o uguale all'attacco magico: Nel caso in cui l'idra ottenga 28, progettare un circuito che stabilisca quando l'idra viene colpita dall'attacco utilizzando un'opportuna rete logica minimizzata riportando il disegno del circuito e la tabella di verità completa.