PSPICE – simulazione codificatori e decodificatori, MUX - DEMUX

Davide Piccolo

Per le dispense delle lezioni:

http://people.na.infn.it/~piccolo/lezioniLaboratorio

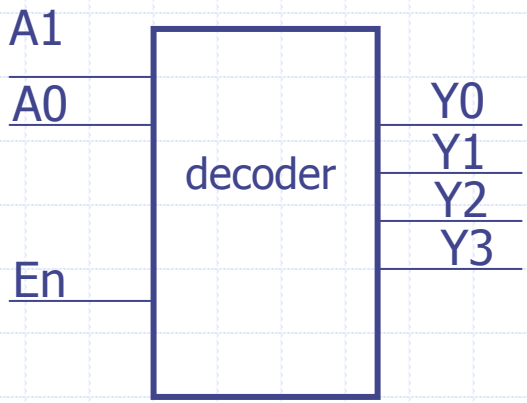
Il circuito decodificatore (decoder)

- Il circuito decodificatore è un circuito combinatorio che attiva una particolare uscita in funzione di una data parola binaria in ingresso.
- Un decoder presenta quindi un numero di linee di ingresso pari al numero di bit necessari a decodificare le possibili uscite, ed un numero di linee di uscita pari al numero di possibili combinazioni definite dai bit di ingresso (es. 3 linee in ingresso, 8 linee in uscita)
- Un ulteriore linea chiamata enable (En) viene impiegata per abilitare o disabilitare il funzionamento del decoder.

… decoder

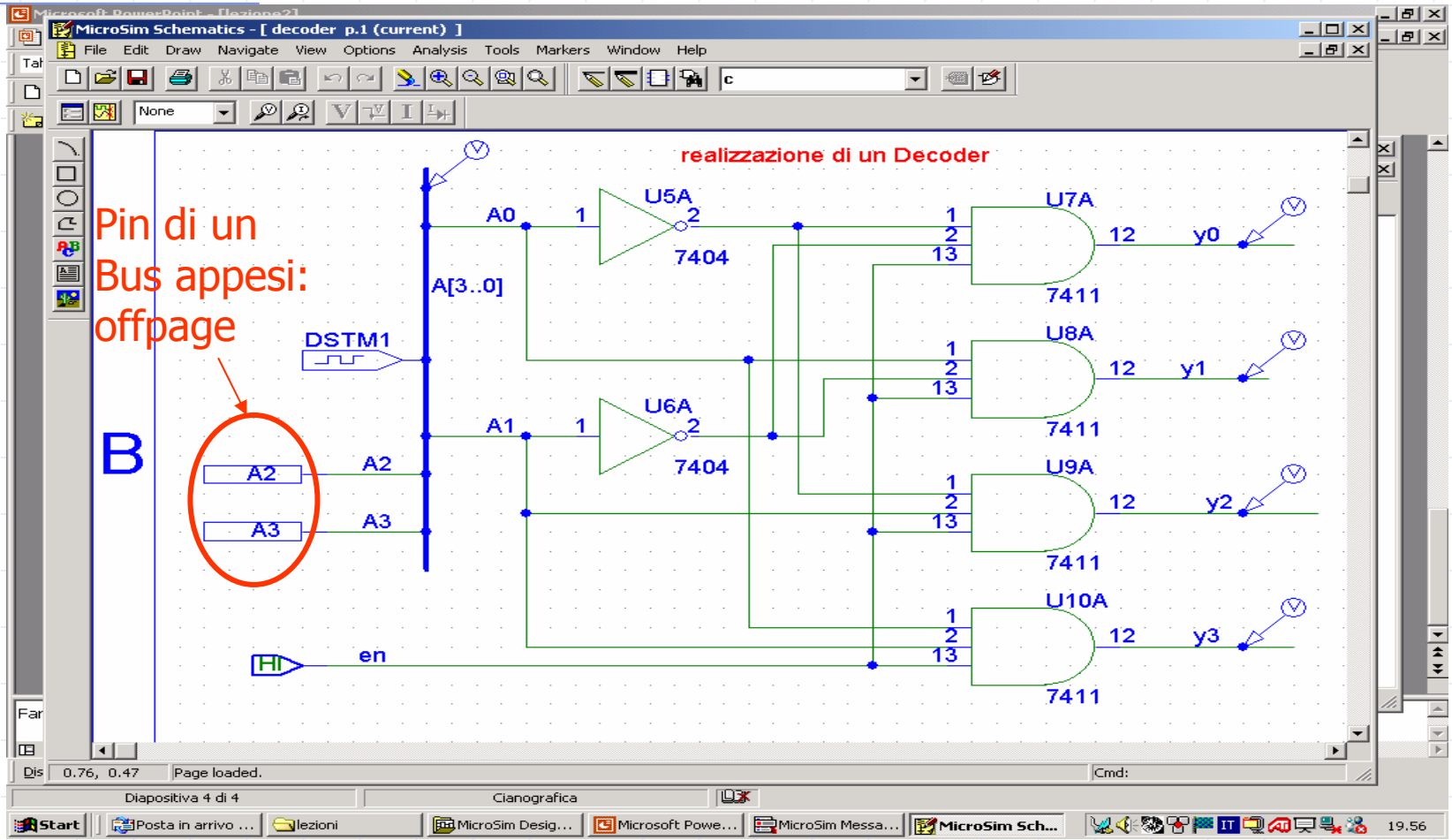
Es di decoder a 2 bit

A1	A0	En	Y0	Y1	Y2	Y3
0	0	1	1	0	0	0
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	1
0	0	0	0	0	0	0
0	1	0	0	0	0	0
1	0	0	0	0	0	0
1	1	0	0	0	0	0

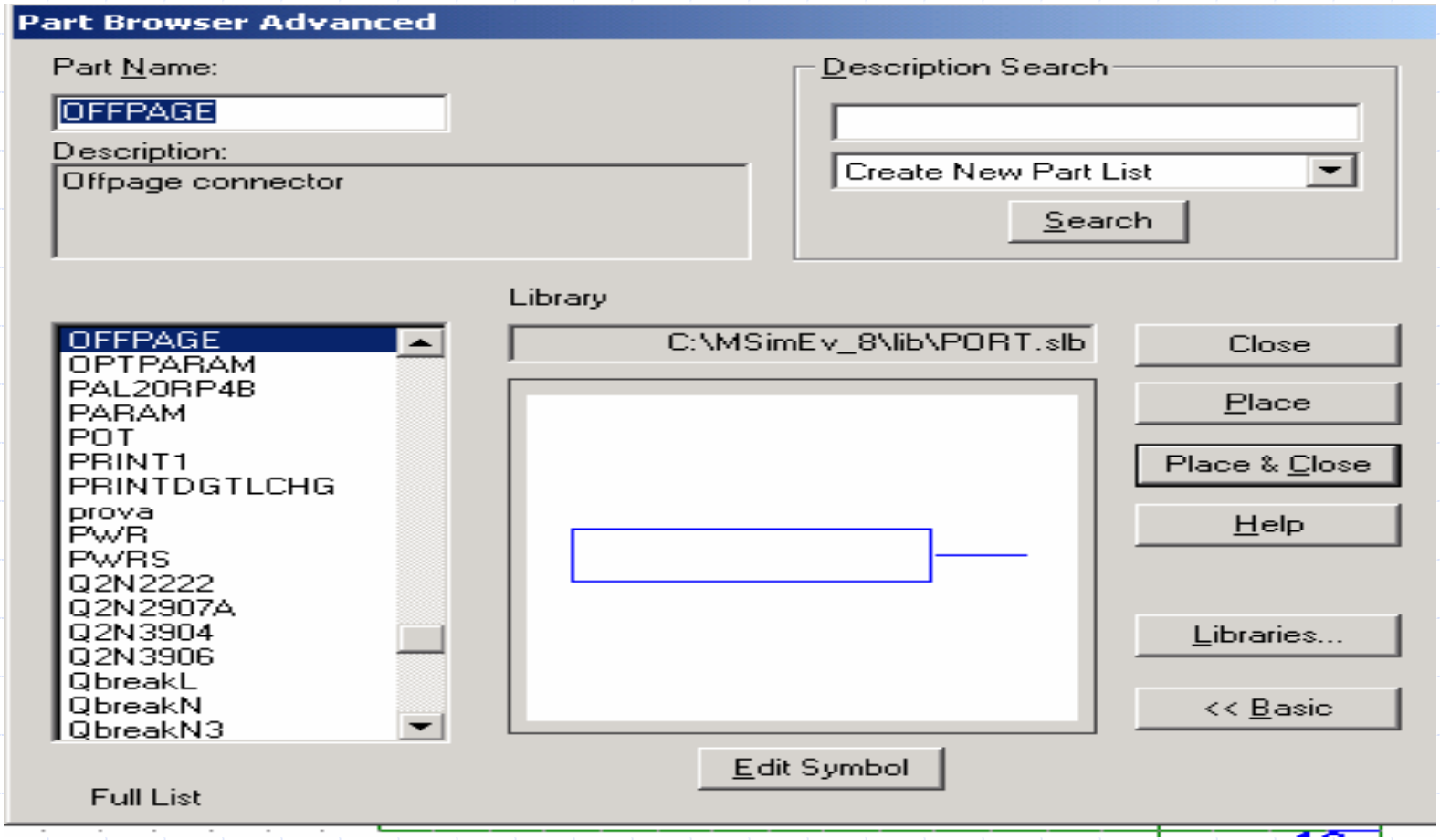


$$Y0 = \overline{A0}\ \overline{A1}\ En$$
$$Y1 = A0\ \overline{A1}\ En$$
$$Y2 = \overline{A0}\ A1\ En$$
$$Y3 = A0\ A1\ En$$

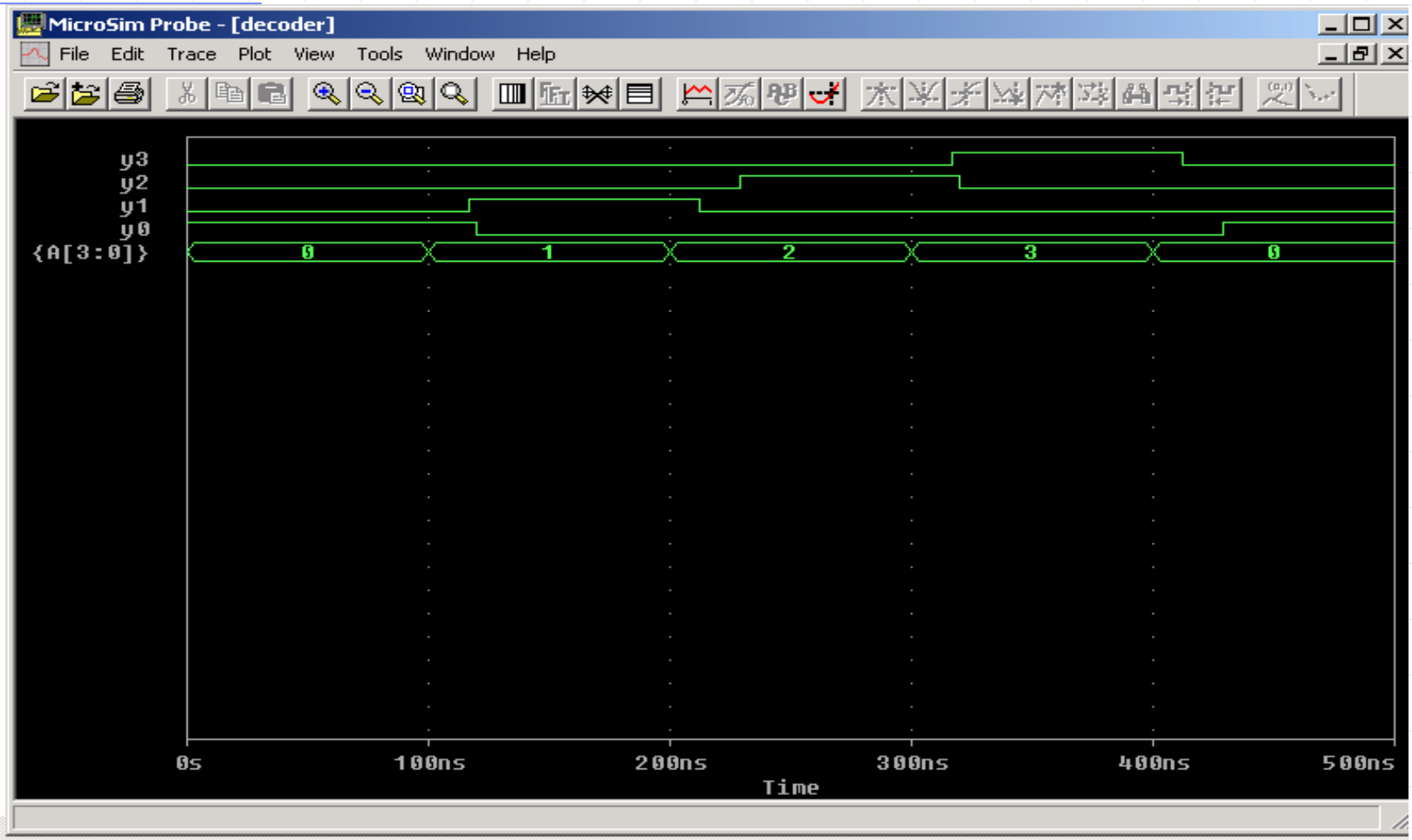
Schematico del decoder



Offpage component



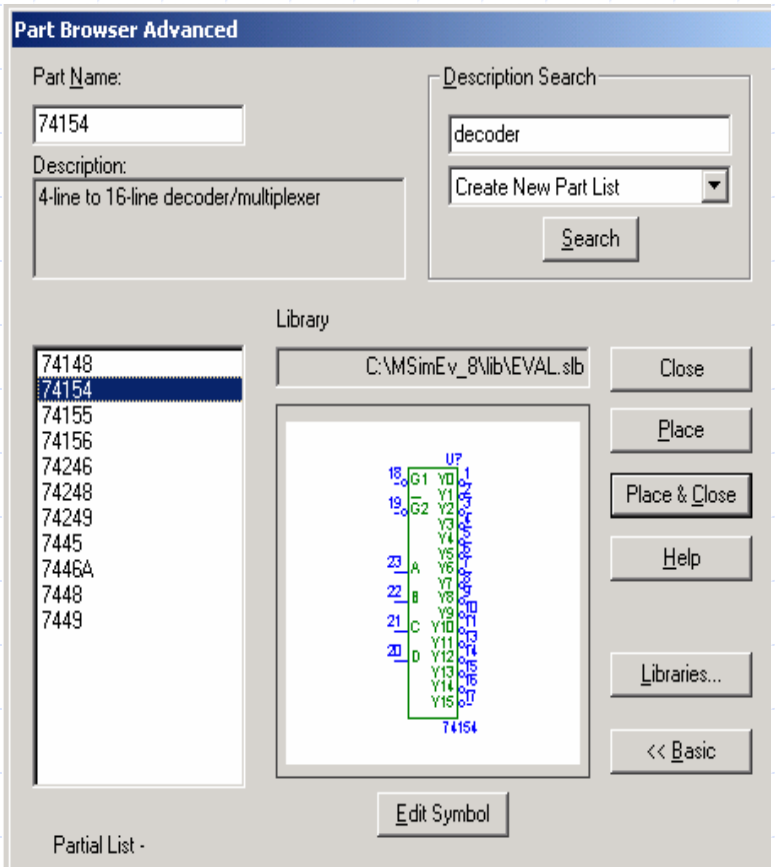
Simulazione del decoder

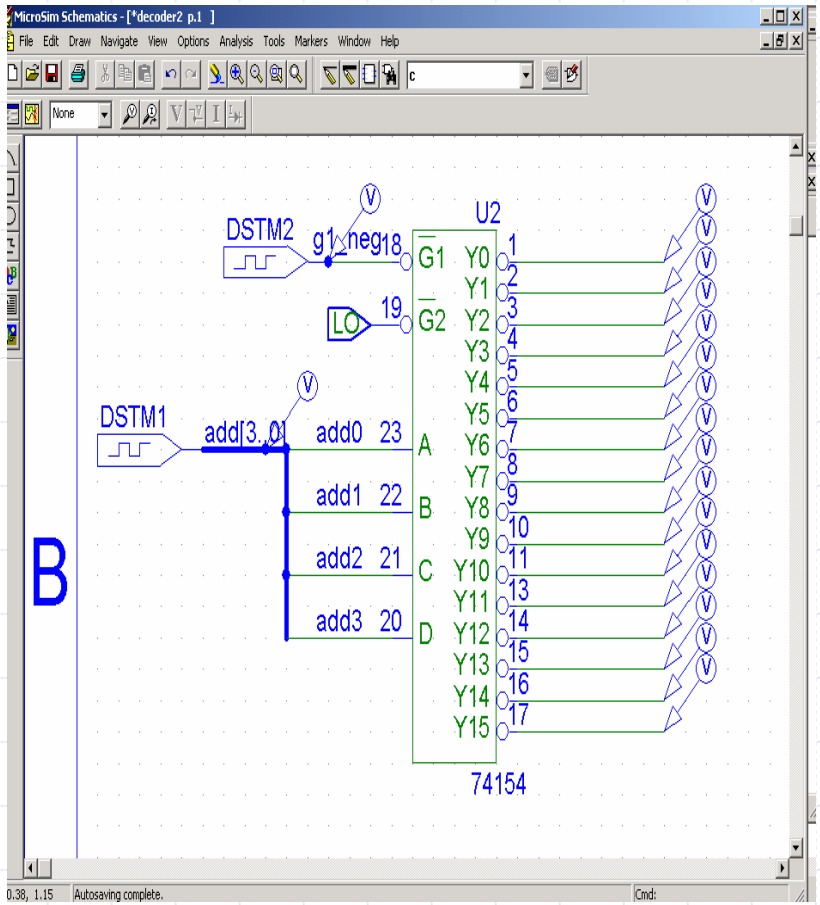


Il decoder realizzato con l'integrato 74154

- Il risultato precedente può essere ottenuto utilizzando un singolo integrato presente in commercio: Decodificatore da 4 a 16 linee, 74154
  - Due ingressi di abilitazione $\overline{G1}$ e $\overline{G2}$ entrambi attivi bassi
  - Quattro ingressi di selezione A, B, C, D
  - Sedici uscite attive basse
- Se $\overline{G1} = \overline{G2} = 0$ allora sarà bassa l'uscita corrispondente al codice in ingresso
- Se almeno uno tra $\overline{G1}$ e $\overline{G2}$ è alto allora tutte le uscite del circuito sono alte

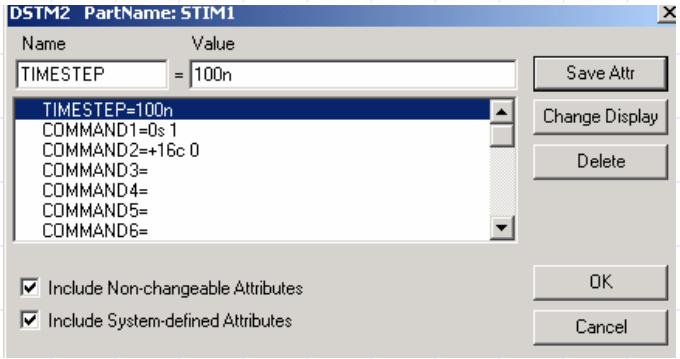
Il decoder 74154



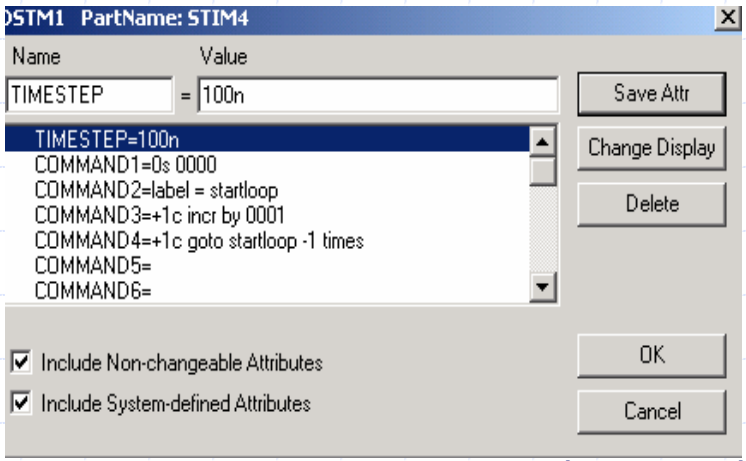


Gli stimoli al circuito e il tempo di simulazione

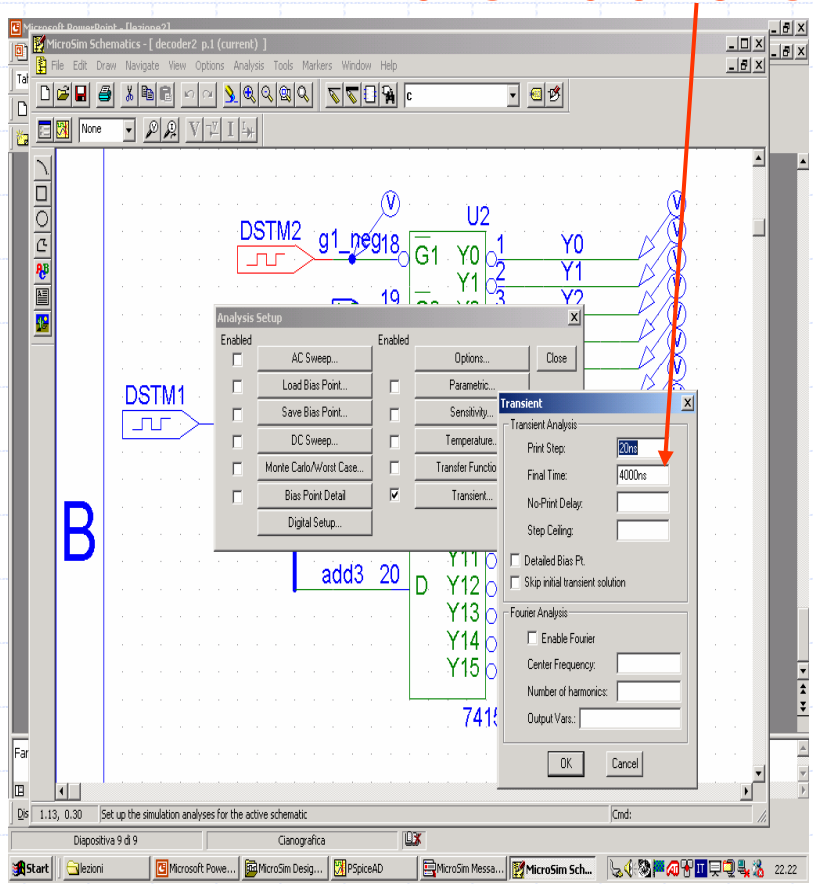
Lo stimolo di $\overline{G1}$



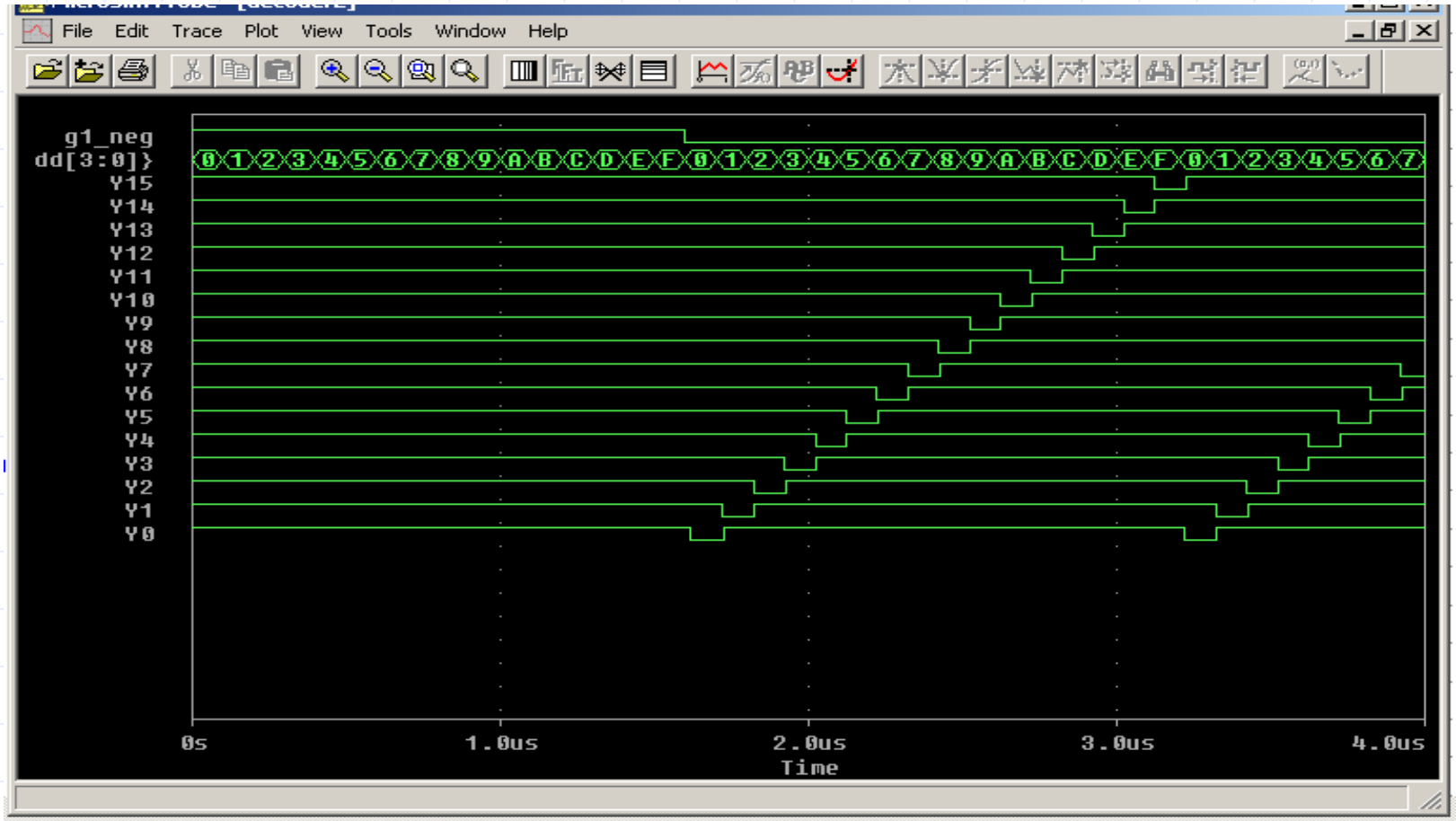
Lo stimolo del bus



Il tempo di simulazione



Risultato della simulazione

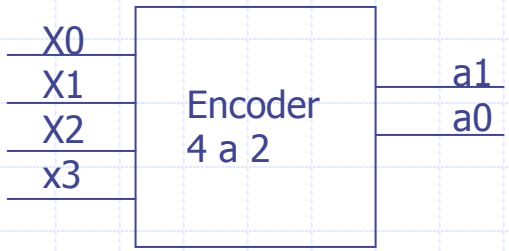


Il circuito codificatore (encoder)

- Il circuito encoder è un circuito combinatorio con M ingressi ed N uscite, all'attivazione di una singola linea di ingresso corrisponde una particolare configurazione dei bit in uscita
  - Opposto dell'operazione di decodifica
  - Il numero di linee di ingresso è superiore al numero di linee in uscita
- Il circuito codificatore più usuale è il codificatore binario, con 2^n ingressi e n uscite
  - Esempio: tasto di una tastiera $\rightarrow$ codice ascii (8 bit)

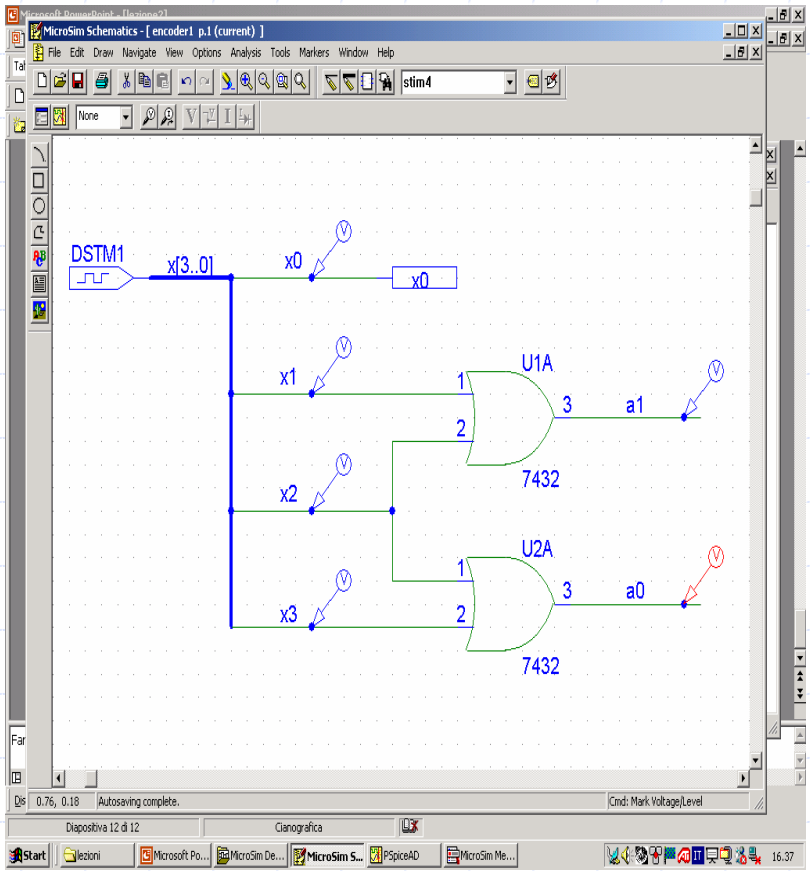
Encoder 4 ingressi, 2 uscite

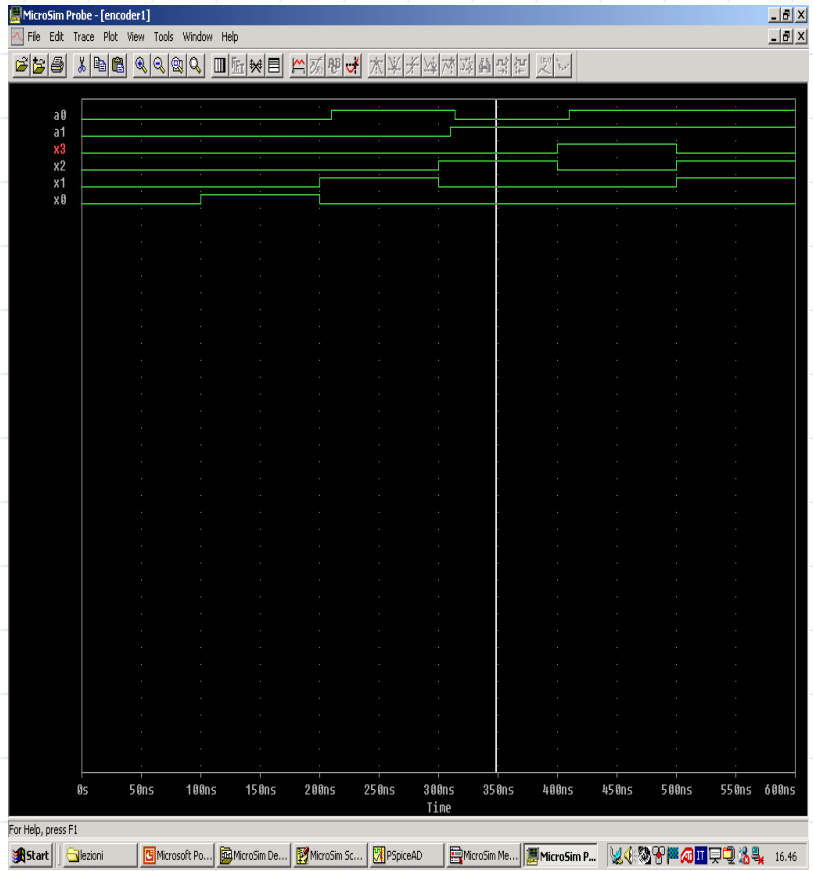
x_3	x_2	x_1	x_0		a_1	a_0
0	0	0	1		0	0
0	0	1	0		0	1
0	1	0	0		1	0
1	0	0	0		1	1



$$a_0 = x_1 + x_3$$
$$a_1 = x_2 + x_3$$

Implementazione e simulazione del circuito





Il *priority encoder*

- Il circuito funziona bene fin quando non vengono attivate contemporaneamente due linee di ingresso
- Nell'esempio precedente quando vengono attivate le linee x1 e x2, il risultato è 3 (a0 = a1 = 1) che non corrisponde né a x1 né a x2
- Per risolvere il problema si usano i *priority encoder* ossia circuiti di codifica con priorità che associano a ciascun ingresso una priorità così che se due ingressi vengono attivati contemporaneamente l'uscita corrisponde al codice della linea di ingresso con priorità maggiore
- Nella famiglia TTL sono disponibili gli integrati 74147, 74148 *priority encoder* rispettivamente 10 in 4 out e 8 in 3 out

Confronto tabella verità tra *encoder* e *priority encoder*

encoder senza priority

a0	a1	a2	a3	a4	a5	a6	a7	a8	a9		u3	u2	u1	u0
0	1	1	1	1	1	1	1	1	1		0	0	0	0
1	0	1	1	1	1	1	1	1	1		0	0	0	1
1	1	0	1	1	1	1	1	1	1		0	0	1	0
1	1	1	0	1	1	1	1	1	1		0	0	1	1
1	1	1	1	0	1	1	1	1	1		0	1	0	0
1	1	1	1	1	0	1	1	1	1		0	1	0	1
1	1	1	1	1	1	0	1	1	1		0	1	1	0
1	1	1	1	1	1	1	0	1	1		0	1	1	1
1	1	1	1	1	1	1	1	0	1		1	0	0	0
1	1	1	1	1	1	1	1	1	0		1	0	0	1

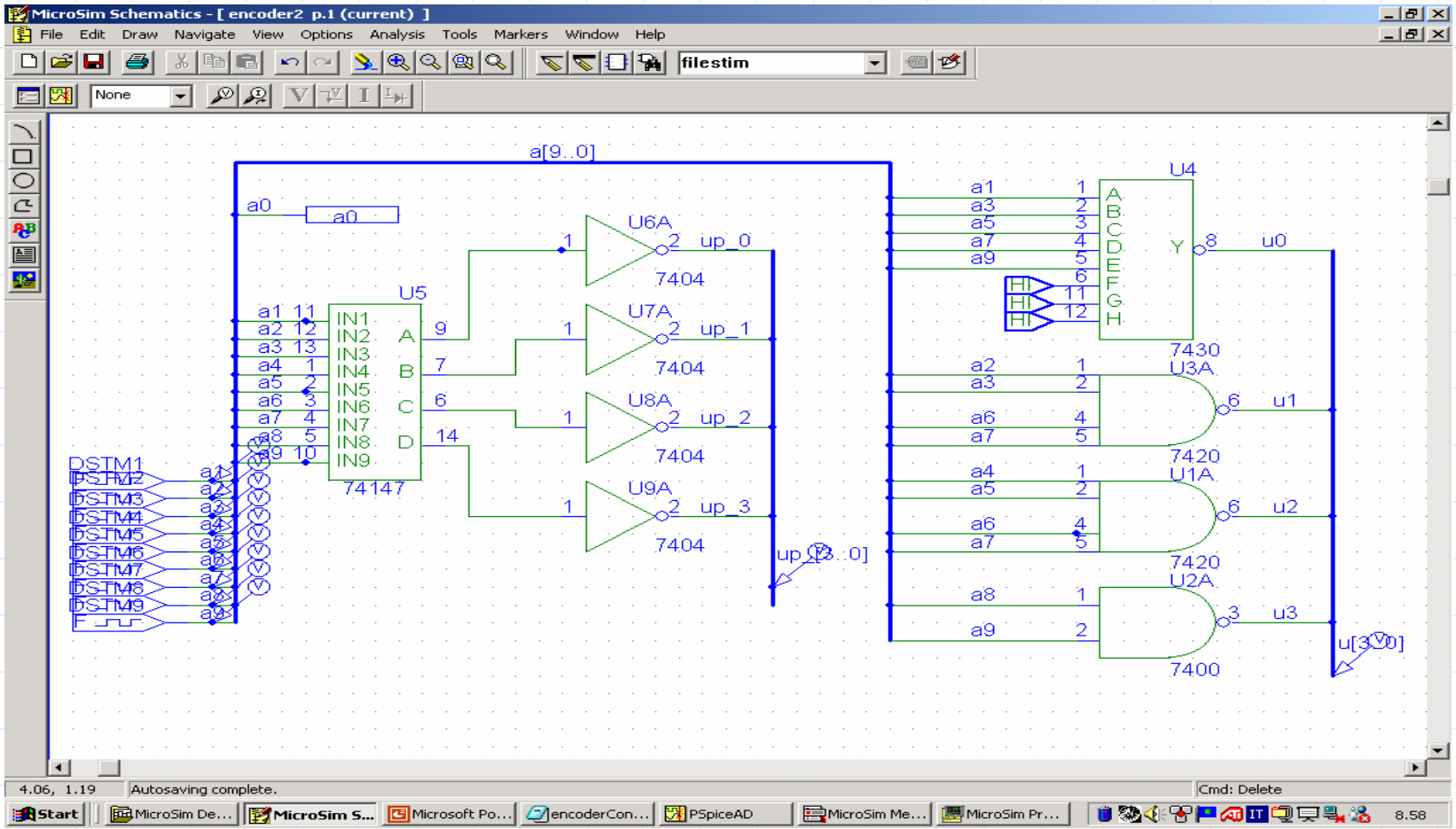
priority encoder

a0	a1	a2	a3	a4	a5	a6	a7	a8	a9		u3	u2	u1	u0
0	1	1	1	1	1	1	1	1	1		0	0	0	0
x	0	1	1	1	1	1	1	1	1		0	0	0	1
x	x	0	1	1	1	1	1	1	1		0	0	1	0
x	x	x	0	1	1	1	1	1	1		0	0	1	1
x	x	x	x	0	1	1	1	1	1		0	1	0	0
x	x	x	x	x	0	1	1	1	1		0	1	0	1
x	x	x	x	x	x	0	1	1	1		0	1	1	0
x	x	x	x	x	x	x	0	1	1		0	1	1	1
x	x	x	x	x	x	x	x	0	1		1	0	0	0
x	x	x	x	x	x	x	x	x	0		1	0	0	1

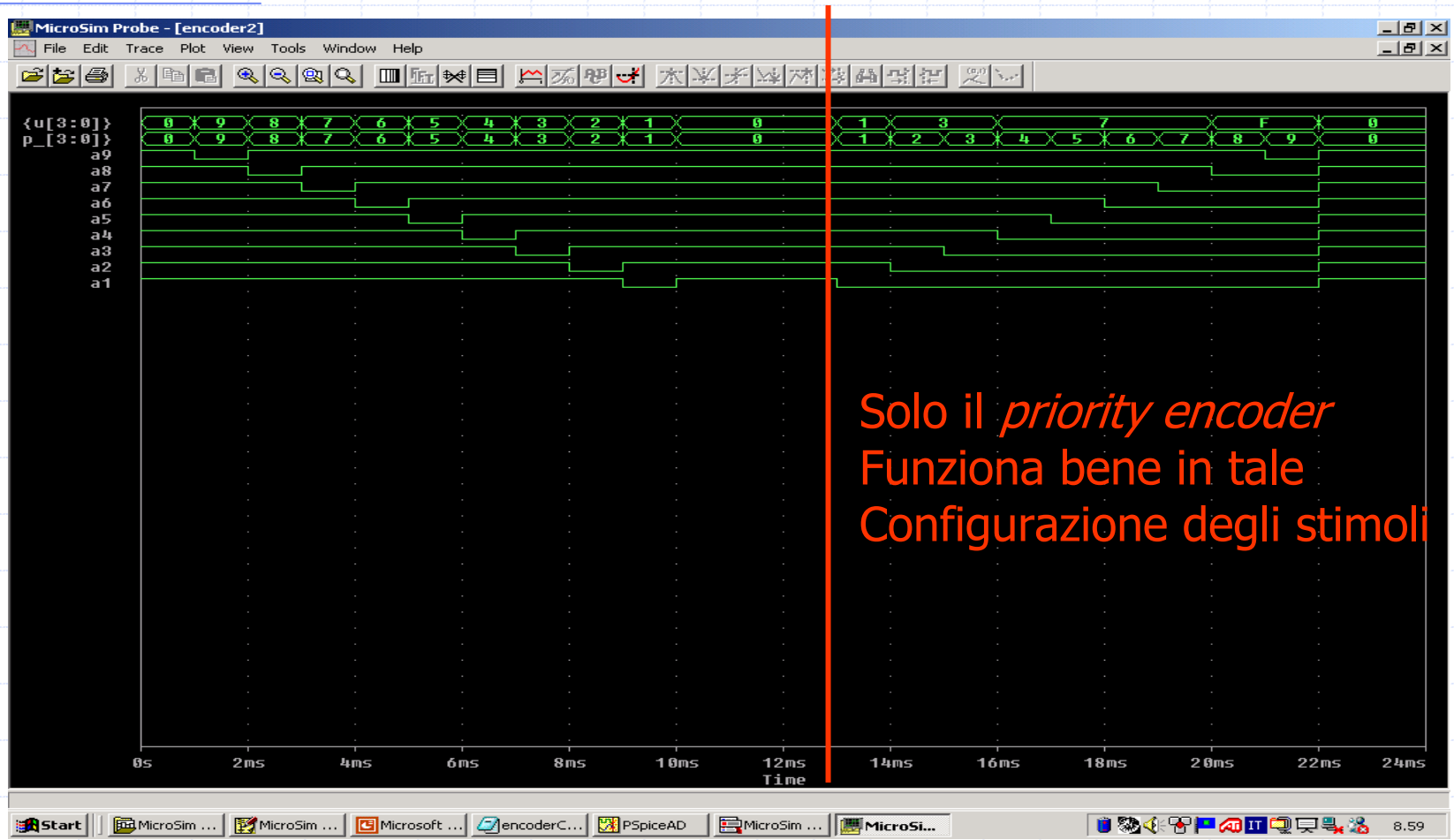
Siccome nelle tecnologie Integrate può risultare Più complicato realizzare OR a molti ingressi piuttosto che NAND a molti ingressi, risulta utile implementare Il circuito con porte NAND

Encoder funzionante in logica negativa

Confronto encoder – priority encoder

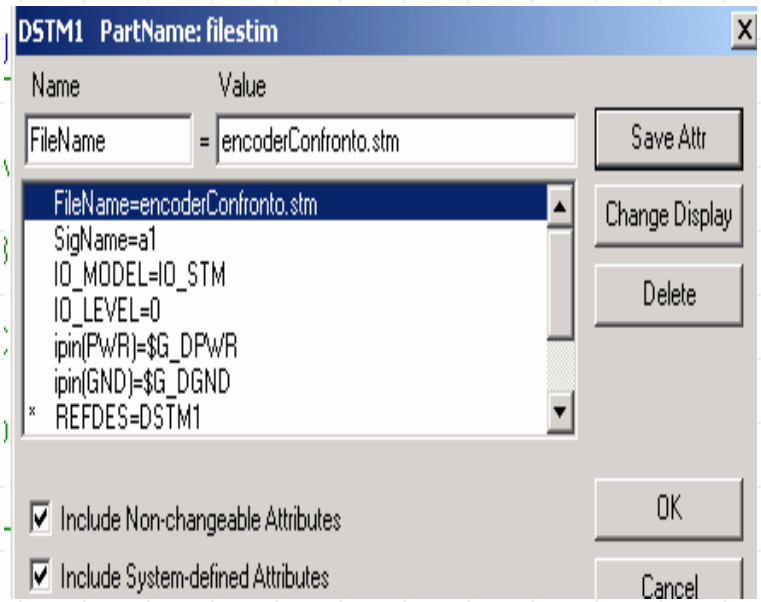


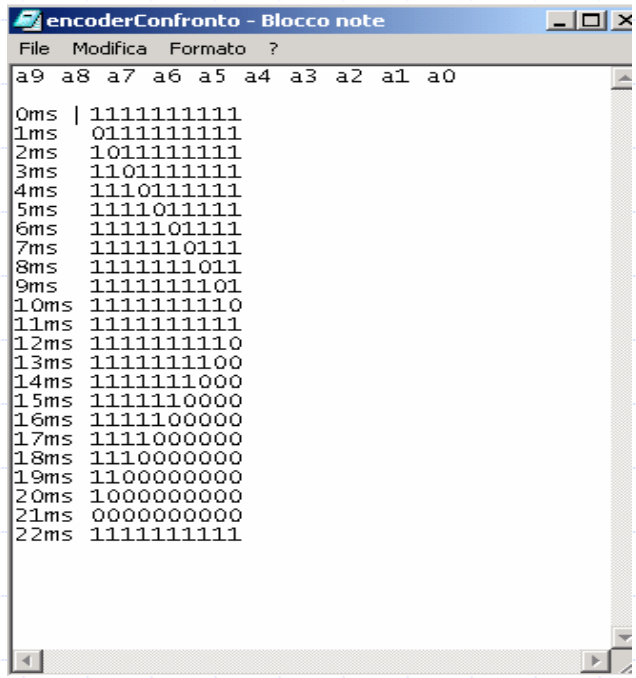
Risultato simulazione



Definizione degli stimoli

Quando è necessario variare gli stimoli troppe volte, si usano i FILESTIM

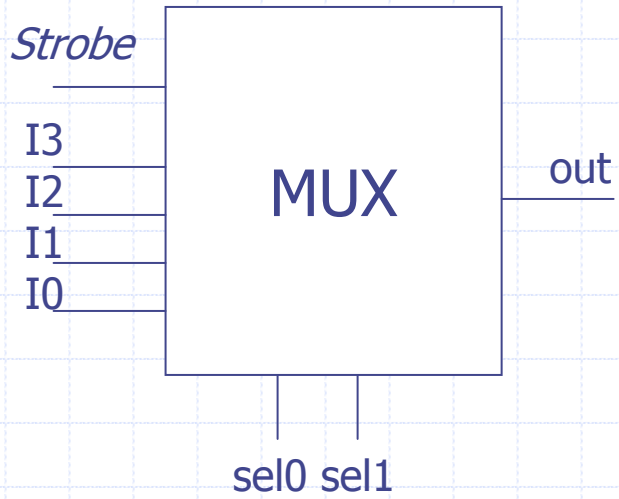




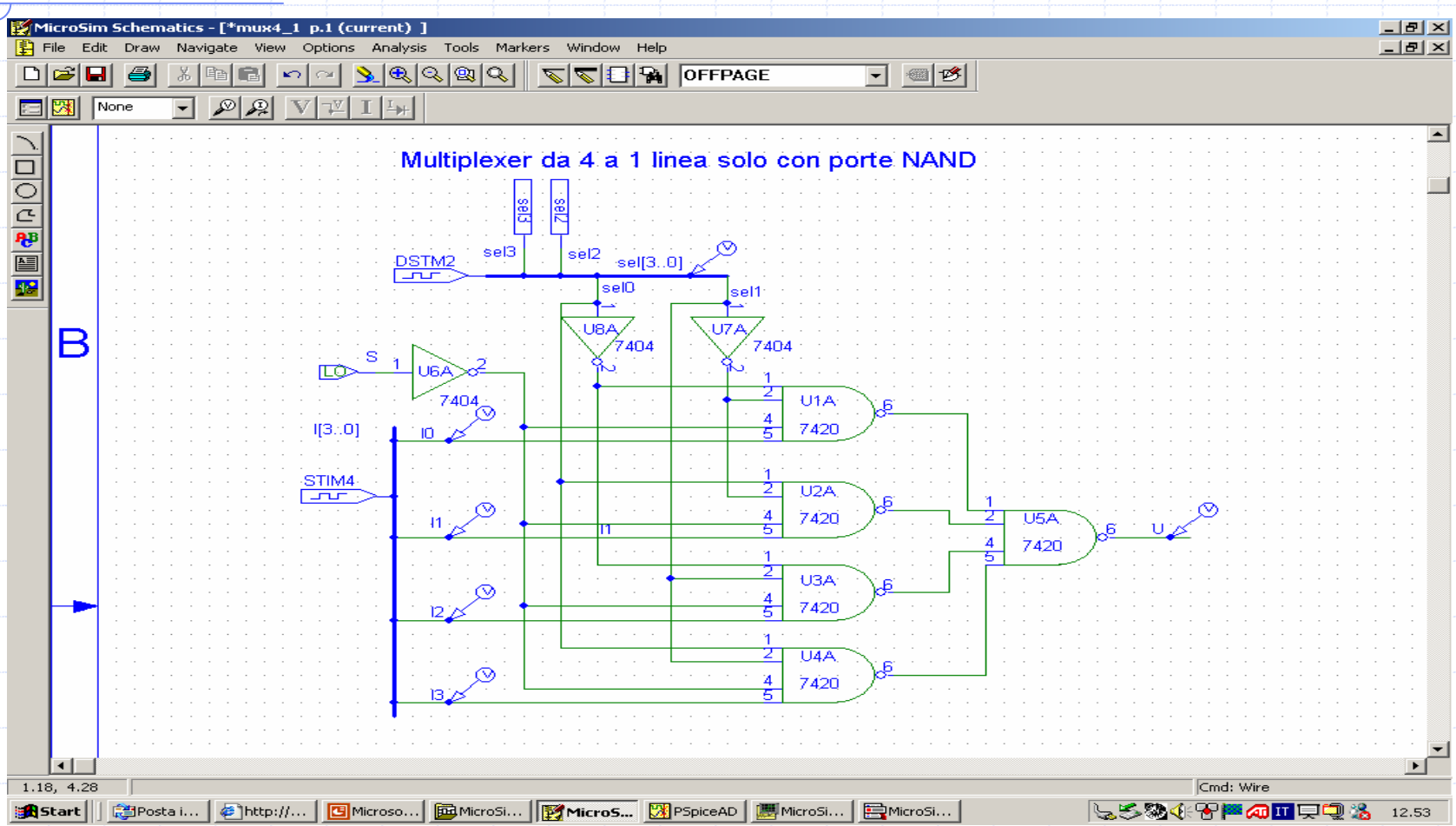
Il circuito *Multiplexer*

- Il *multiplexer* è un circuito in grado di selezionare una particolare linea di ingresso e di inviarla in uscita. La selezione della linea di ingresso è definita da opportune linee di selezione.

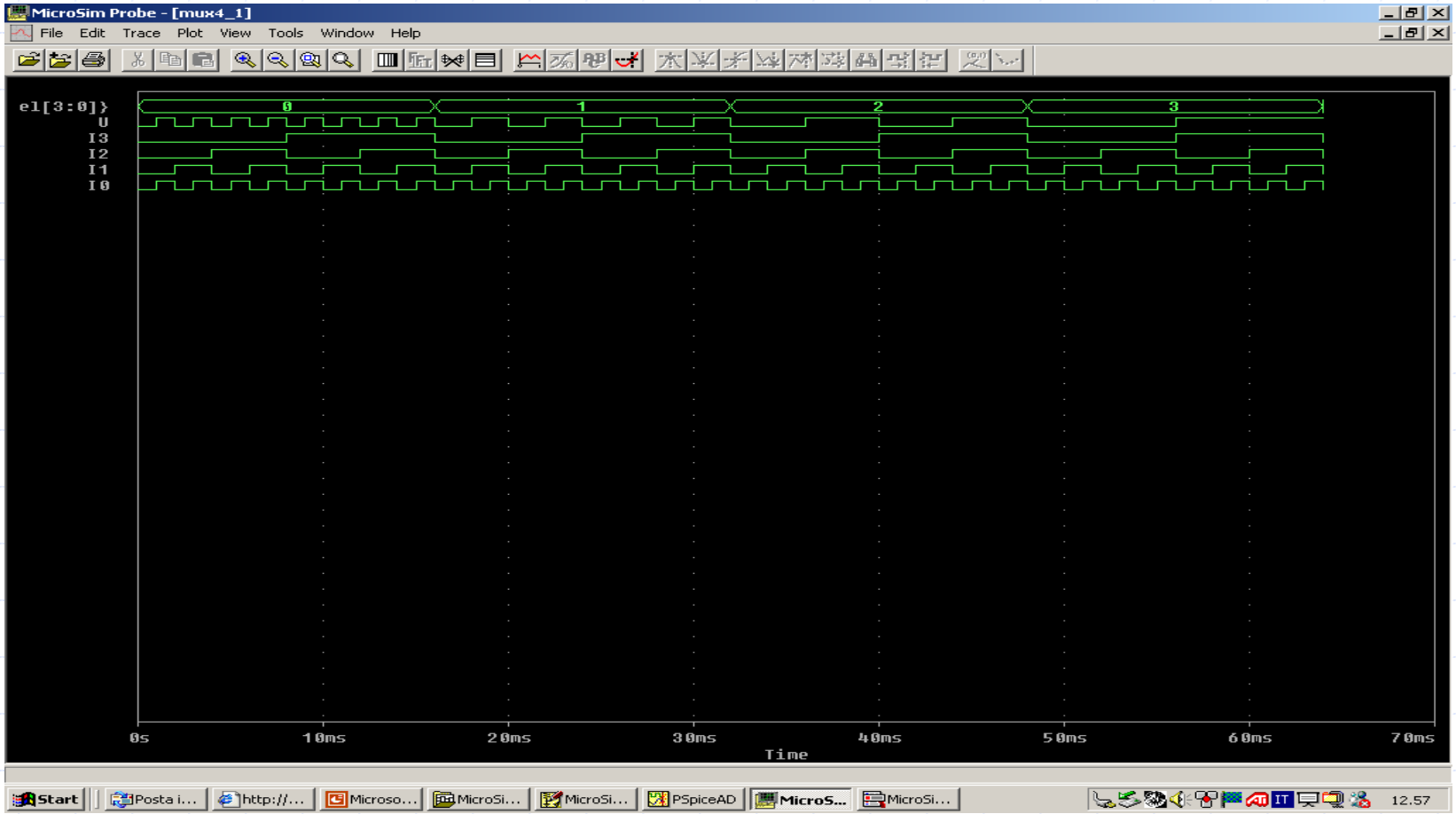
- Esempio di *multiplexer* da 4 a 1 linea: una solamente delle 4 linee di ingresso (I0.. I3) viene presentata in uscita. La selezione avviene tramite le due linee di *sel*. La linea di *strobe* permette di abilitare o meno il funzionamento del circuito.



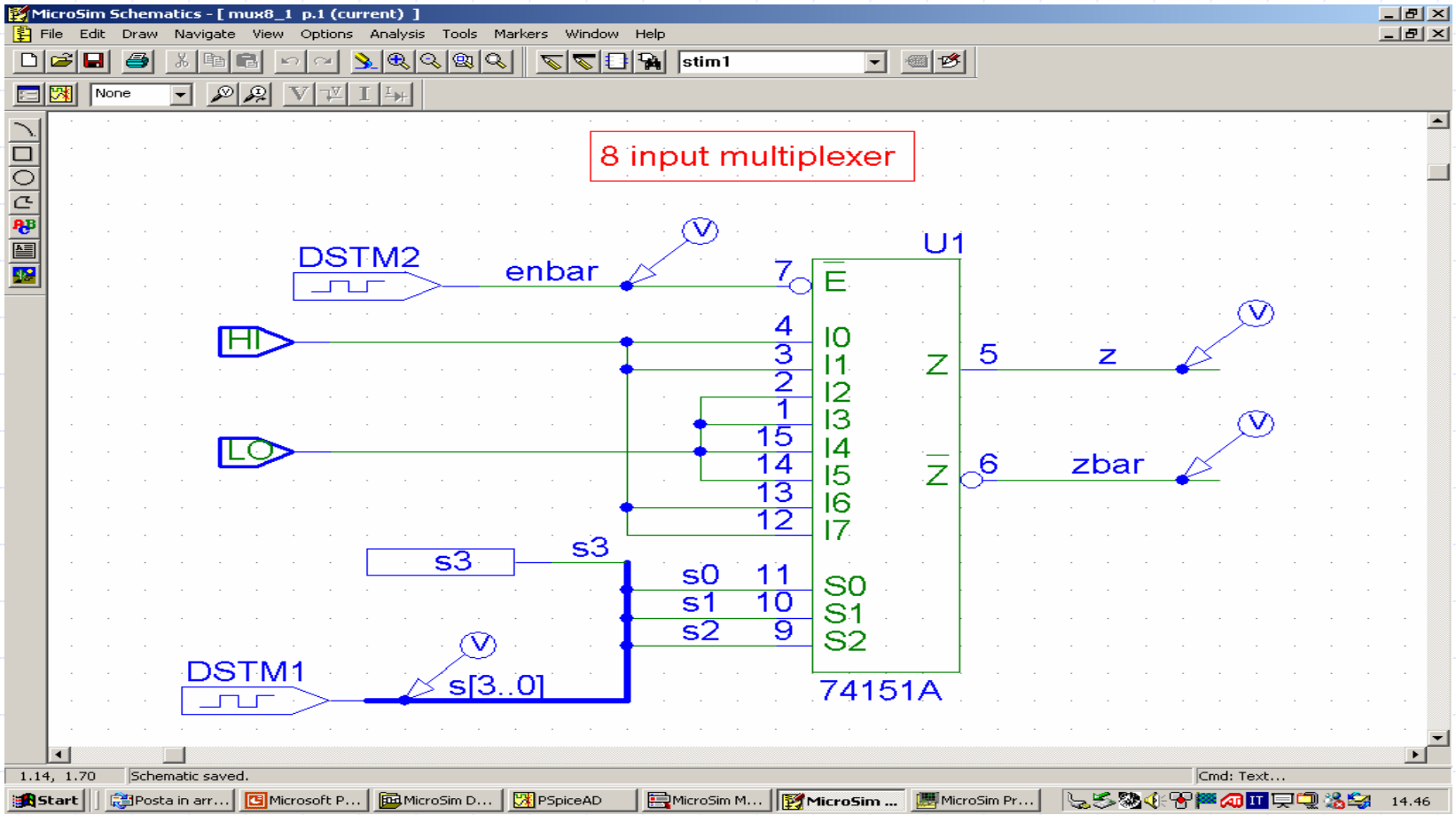
MUX con sole porte NAND



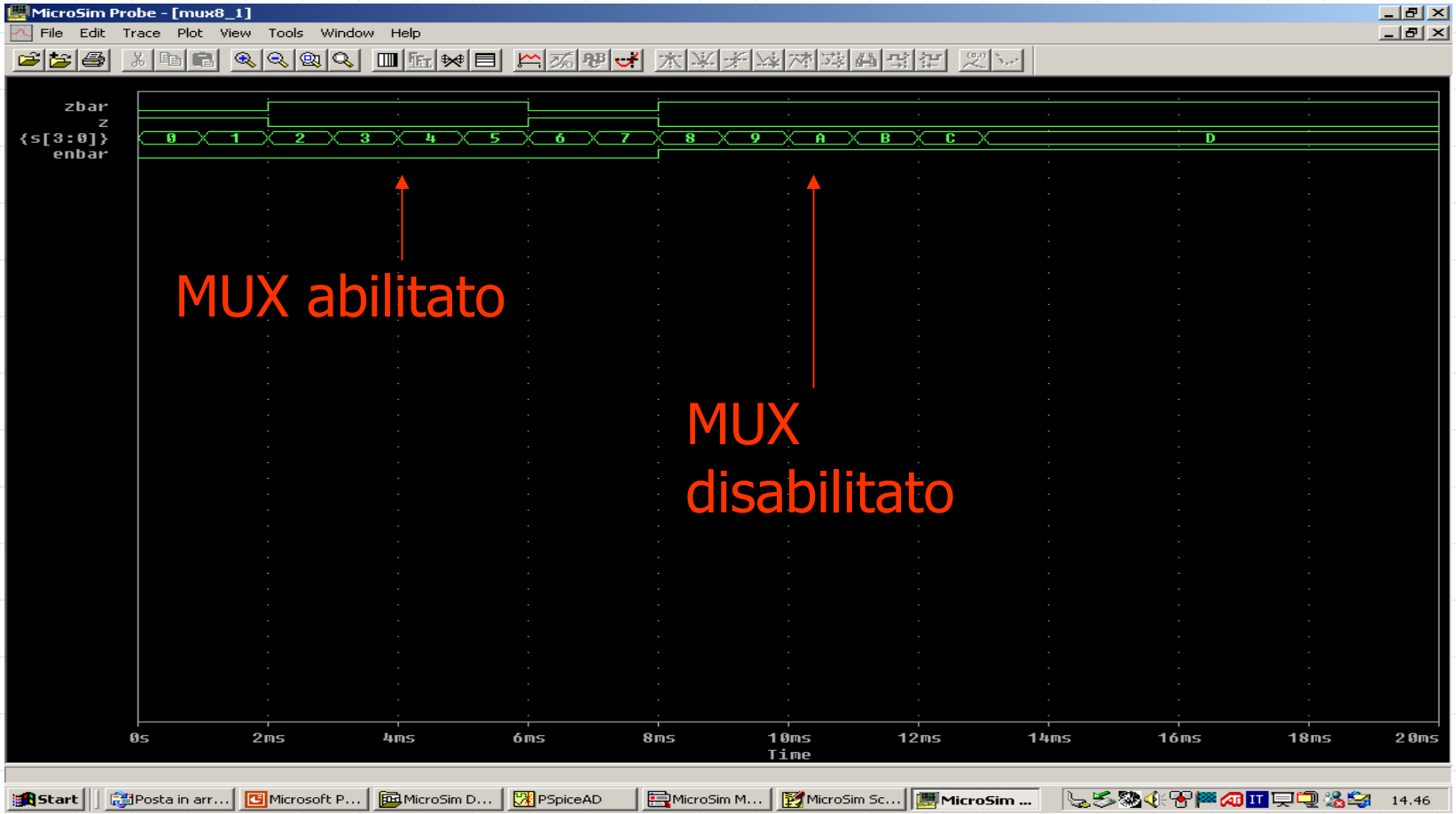
Risultato simulazione



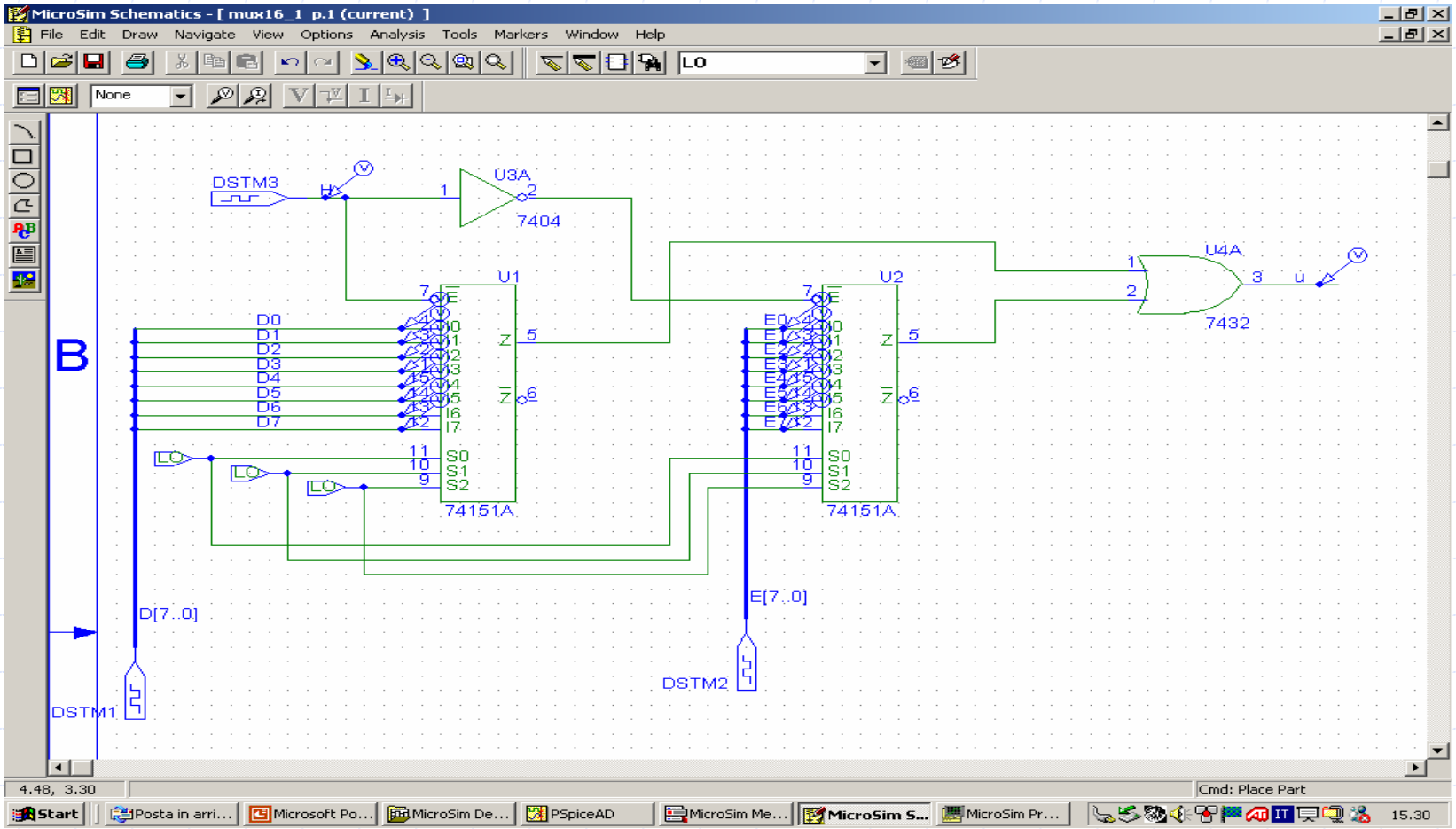
Multiplexer realizzato con l'integrato 74151A (MUX 8 a 1)



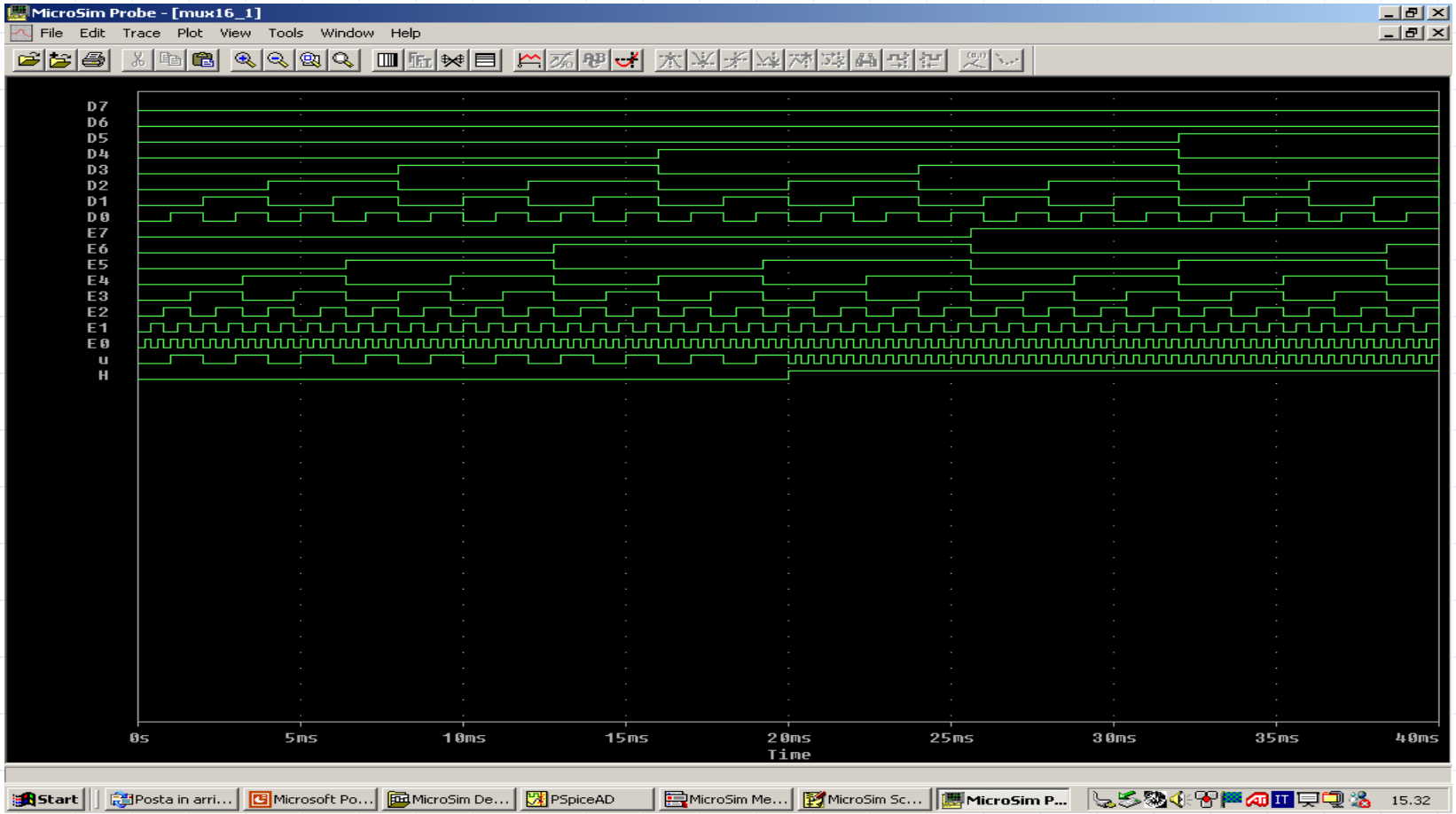
Risultato simulazione 74151A



MUX 16 a 1

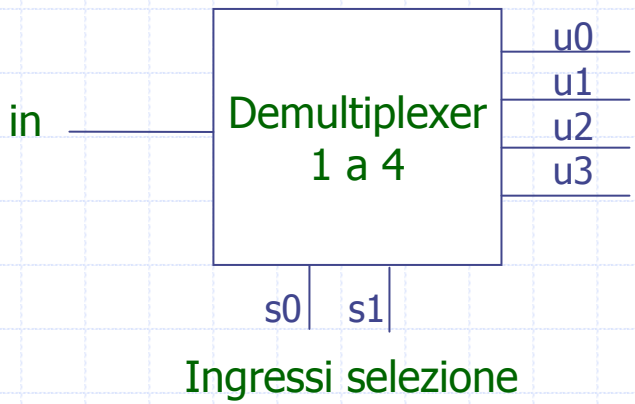

MicroSim Schematics - [mux16_1 p.1 (current)]
File Edit Draw Navigate View Options Analysis Tools Markers Window Help
LO
None
DSTM3
U3A
7404
U1
U2
U4A
7432
D0
D1
D2
D3
D4
D5
D6
D7
B
LO
S0
S1
S2
74151A
D[7..0]
E[7..0]
DSTM1
DSTM2
4.48, 3.30
Cmd: Place Part
Start
Posta in arri...
Microsoft Po...
MicroSim De...
PSpiceAD
MicroSim Me...
MicroSim S...
MicroSim Pr...
15.30

Risultato simulazione MUX 16 a 1



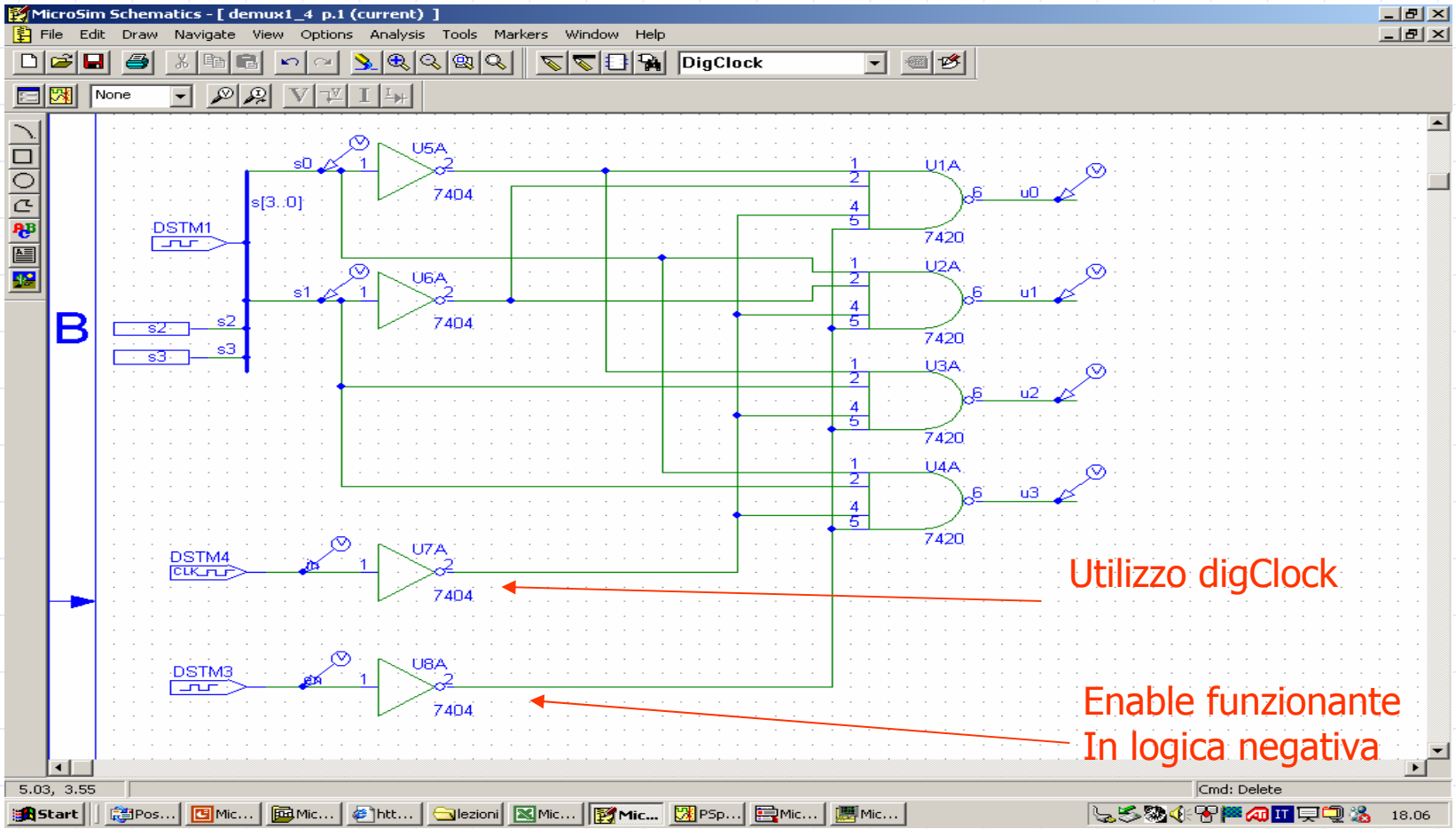
Il *DEMULTIPLEXER*

- Il *demultiplexer* è un circuito che permette di trasmettere dei dati binari seriali (ossia provenienti da una sola linea) su una particolare linea di uscita selezionata tra N linee tramite un apposito indirizzo
- Operazione opposta rispetto al *multiplexer*

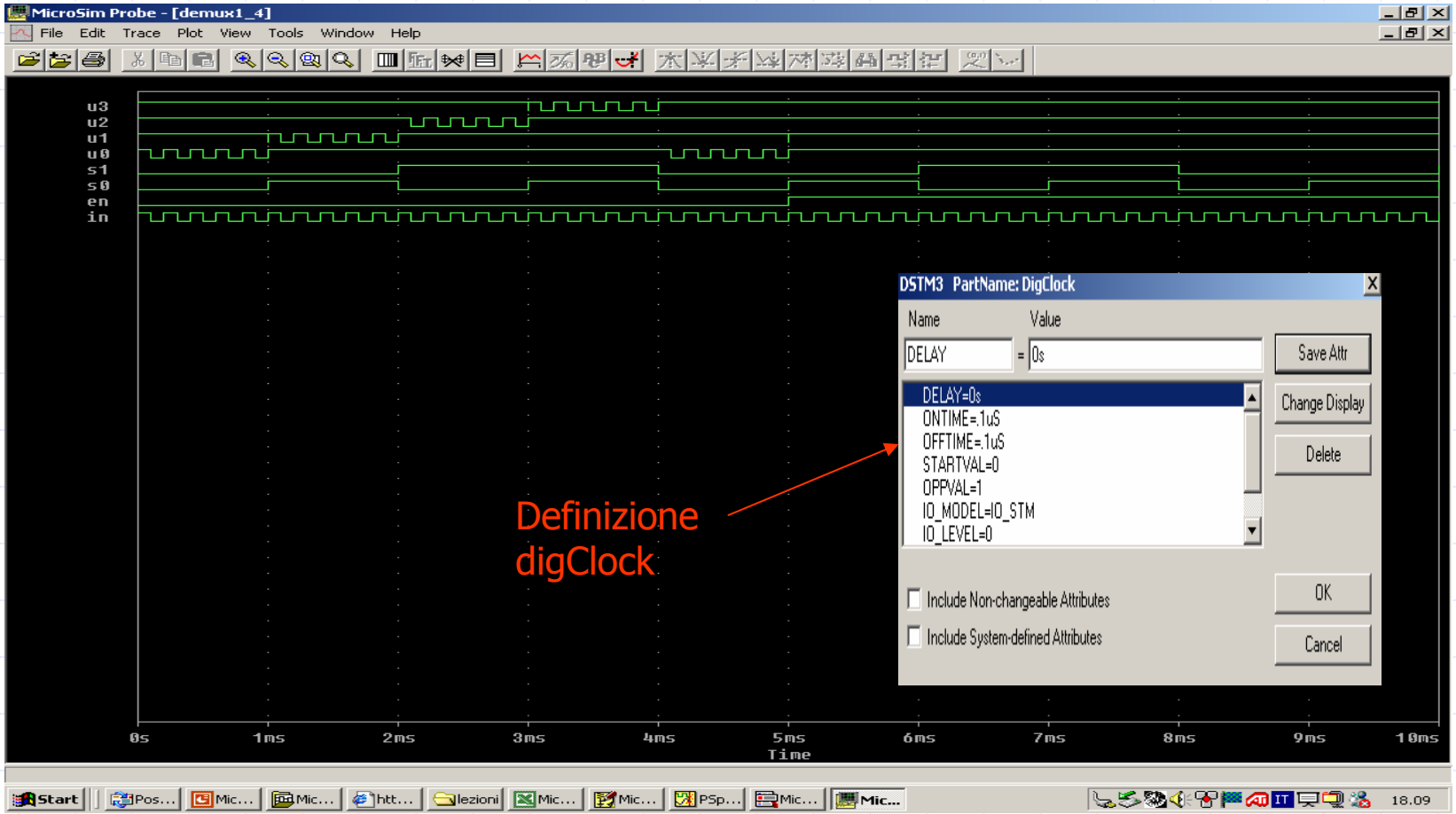


Demultiplexer 1 a 4					
s1	s0	u0	u1	u2	u3
0	0	in	0	0	0
0	1	0	in	0	0
1	0	0	0	in	0
1	1	0	0	0	in

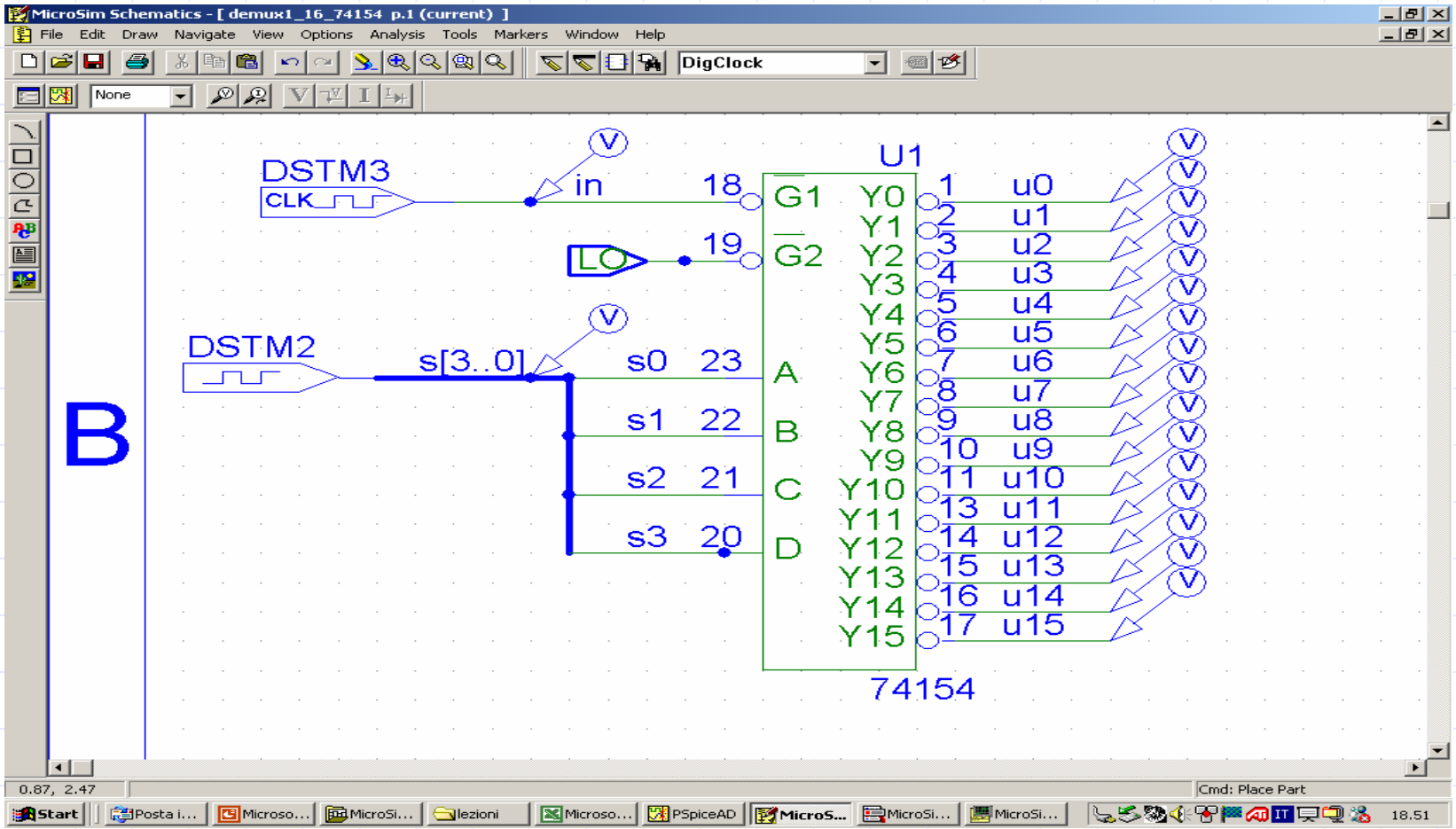
DEMUX implementato con porte NAND



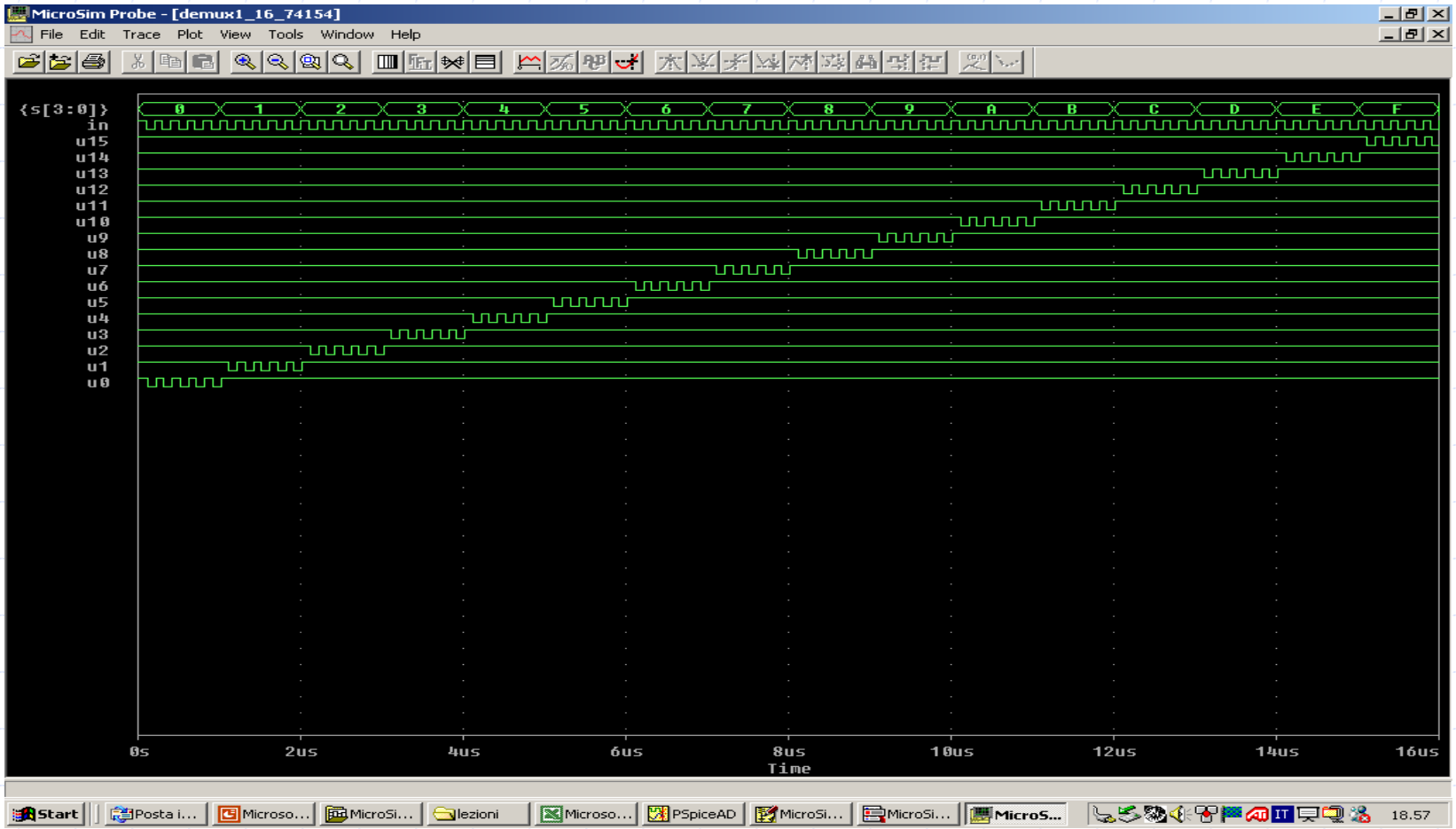
Risultato della simulazione



DEMUX 1 a 16 con l'integrato 74154



Simulazione del DEMUX 74154

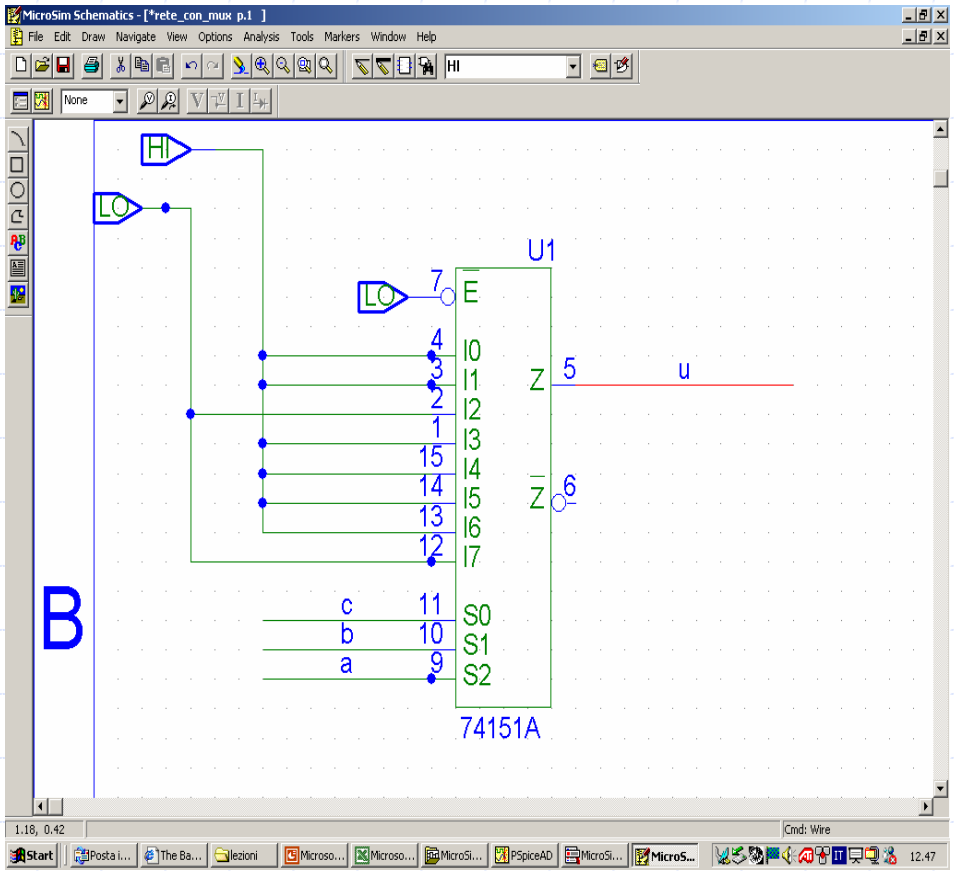


Utilizzo di MUX e DEMUX per realizzare una rete combinatoria

- Il MUX e DEMUX possono essere usati per implementare una rete combinatoria di cui si conosca la tabella di verità
- Combinando opportunamente gli ingressi e le uscite di un MUX o di un DEMUX si realizzano circuiti anche complessi senza utilizzare troppi componenti

Esempio di realizzazione di una rete combinatoria tramite MUX

tabella di verità di circuito arbitrario			
a	b	c	y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



Esempio di realizzazione di una rete combinatoria tramite DEMUX

tabella di verità di un circuito arbitrario				
a	b	c	d	y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1