Universitatea Politehnica București

Facultatea de Automatică si Calculatoare

Proiect - proiectare logică

Aspirator robot

Brebu Costin Bogdan

Grupa: 311CD

Cuprins

Tema proiectului

Aparatul ales este un aspirator robot, folosit pentru a curăța complet podeaua.

- Aspiră
- Spala
- Usucă

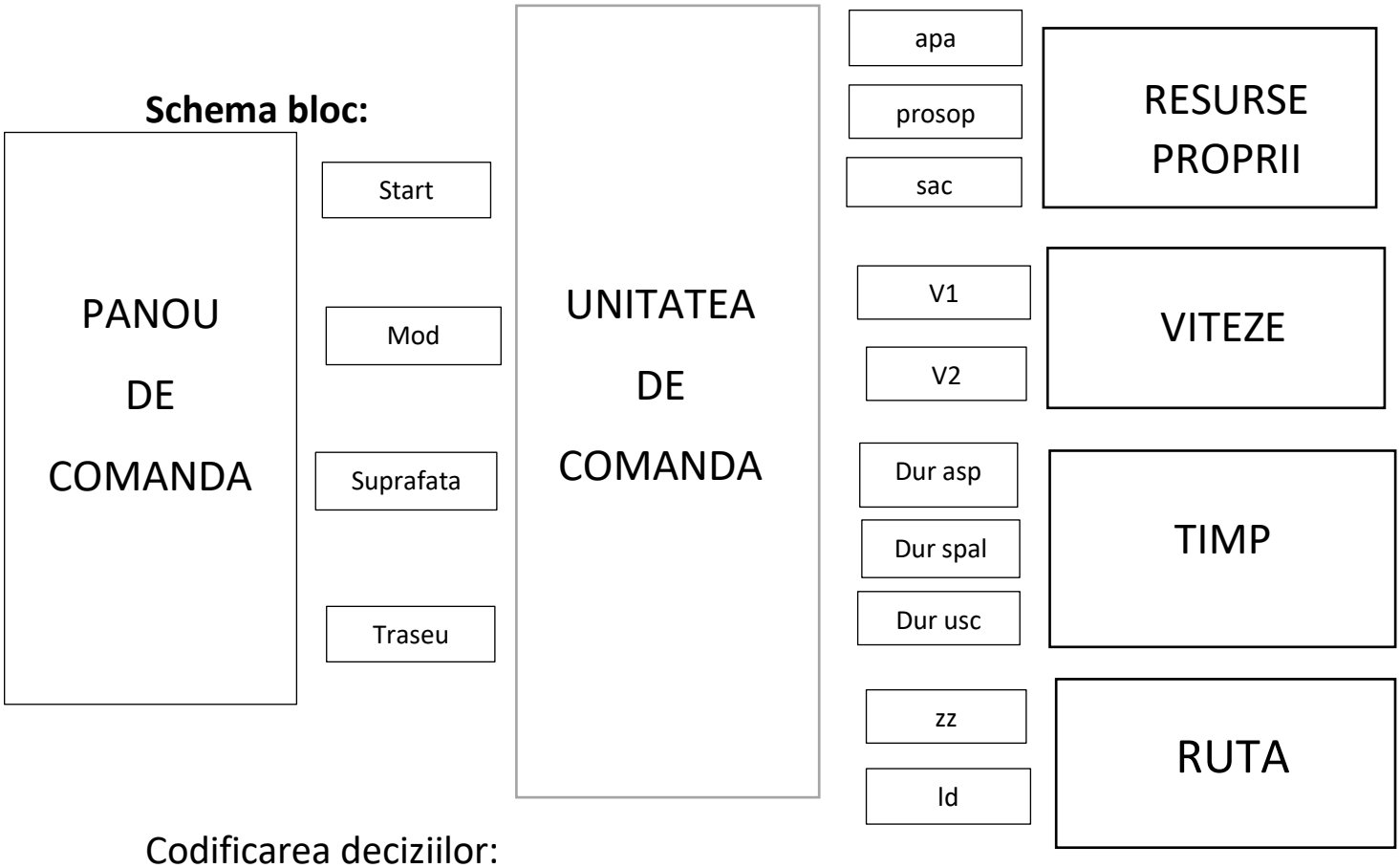
Folosește un sistem minuțios de curățare, datorită căruia robotul curăță podeaua până la capacitatea maximă a bateriei acoperind astfel întotdeauna complet toată suprafața. După finalizarea curățării, acesta este parcat automat în baza de încărcare.

Linie dreapta Program scurt intens pentru o curățare direcționată a suprafeței de 2 x 2 m.

Zig - Zag Program special pentru curățarea suprafețelor mai mari.

Mod de implementare

Schema bloc:



Codificarea deciziilor:

Suprafata:

-0- mica

-1- mare

Mod:

-0- aspirare

-1- uscare

Traseu:

-0- zz = zig zag

-1- ld=linie dreapta

Explicarea functionalitatii

Robotul porneste la apasarea butonului start. La panoul de comanda se selecteaza suprafata si modul.

Daca se selecteaza aspirarea, automat la starsitul aspirarii se trece la spalare si apoi uscare. Daca se selecteaza spalarea se trece automat la uscare. Nu se poate incepe direct cu uscarea. In cazul in care programul selectat este aspirarea, robotul deschide automat saculetul de gunoi, daca e pe spalare isi extrage apa, iar daca e pe uscare coboara automat carpa care absoarbe apa.

Daca se selecteaza de la panoul de comanda o suprafata mare(mai mare de 4 pe 4) aspiratorul va functiona la viteza mai mare (v1) . Daca se selecteaza o suprafata mica aspiratorul va functiona la viteza mica (v0).

Durata aspirarii, a spalarii si a uscarii sunt presetate si se fac automat in functie de modul de functionare.

Organigrama

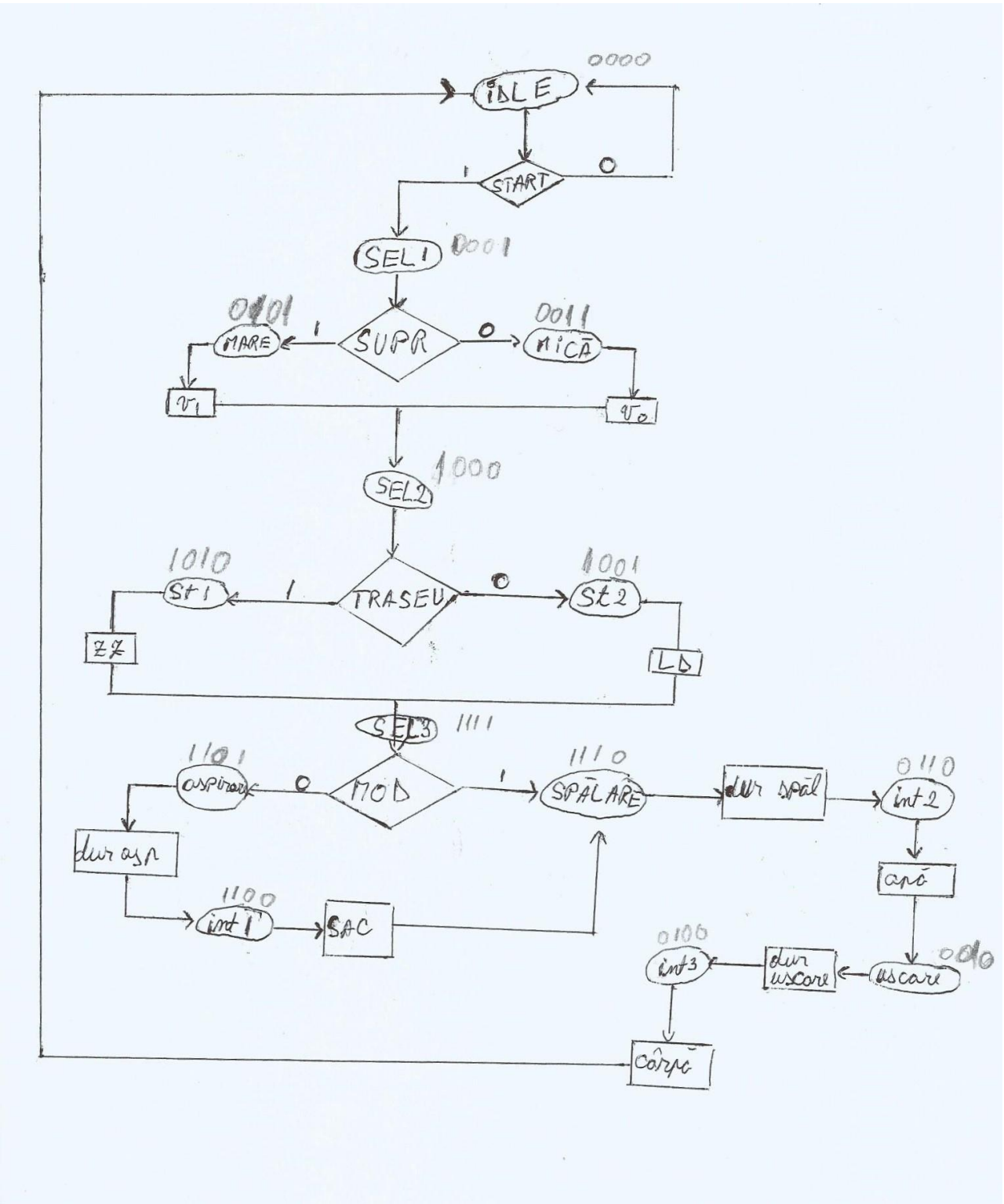


Diagrama de stari

xy // zt	00	01	11	10
00	idle	Int3	Int1	Sel2
01	Sel1	Mare	Aspirare	St2
11	Mica	*	Sel3	*
10	Uscare	Int2	Spalare	St1

Tabel de iesiri

X^n	Y^n	Z^n	T^n	apa	carpa	sac	V1	V2	dur asp	dur spal	Dur usc	Zz	Ld
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	1	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	1	0	0	0	0	0	0
0	1	1	0	1	0	0	0	0	0	0	0	0	0
0	1	1	1	*	*	*	*	*	*	*	*	*	*
1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1
1	0	1	0	0	0	0	0	0	0	0	0	1	0
1	0	1	1	*	*	*	*	*	*	*	*	*	*
1	1	0	0	0	0	1	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	1	0	0	0	0
1	1	1	0	0	0	0	0	0	0	1	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0

Apa

xy // zt	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	*	0	*
10	0	1	0	0

Apa= !x*y*z

Carpa

xy // zt	00	01	11	10
00	0	1	0	0
01	0	0	0	0
11	0	*	0	*
10	0	0	0	0

Carpa=!x*y*!z*!t

Sac

xy // zt	00	01	11	10
00	0	0	1	0
01	0	0	0	0
11	0	*	0	*
10	0	0	0	0

Sac=x*y*!z*!t

V1

xy // zt	00	01	11	10
00	0	0	0	0
01	0	1	0	0
11	0	*	0	*
10	0	0	0	0

V1=!x*y*t

V2

xy // zt	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	*	0	*
10	0	0	0	0

V2=!x*z*t

Dur asp

xy // zt	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	*	0	*
10	0	0	0	0

Dur asp=x*y*!z*t

Dur spal

xy // zt	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	*	0	*
10	0	0	1	0

Dur spal=x*y*z*!t

Dur usc

xy // zt	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	*	0	*
10	1	0	0	0

Dur usc=!x*!y*z*!t

Zz

xy // zt	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	*	0	*
10	0	0	0	1

Zz=x*!y*z

Ld

xy // zt	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	0	*	0	*
10	0	0	0	0

Ld=x*!y*t

Tabelul tranzitiilor

X^n	Y^n	Z^n	T^n	X^(n+1)	Y^(n+1)	Z^(n+1)	T^(n+1)	Dx	Jy	Ky	Dz	Jt	Kt
0	0	0	0	0	0	0	Start	0	0	1	0	start	!start
0	0	0	1	0	supr	!supr	1	0	supr	!supr	!supr	1	0
0	0	1	0	0	1	0	0	0	1	0	0	0	1
0	0	1	1	1	0	0	0	1	0	1	0	0	1
0	1	0	0	0	0	0	0	0	0	1	0	0	1
0	1	0	1	1	0	0	0	1	0	1	0	0	1
0	1	1	0	0	0	1	0	0	0	1	1	0	1
0	1	1	1	*	*	*	*	*	*	*	*	*	*
1	0	0	0	1	0	traseu	!traseu	1	0	1	traseu	!traseu	traseu
1	0	0	1	1	1	1	1	1	1	0	1	1	0
1	0	1	0	1	1	1	1	1	1	0	1	1	0
1	0	1	1	*	*	*	*	*	*	*	*	*	*
1	1	0	0	1	1	1	0	1	1	0	1	0	1
1	1	0	1	1	1	0	0	1	1	0	0	0	1
1	1	1	0	0	1	1	0	0	1	0	1	0	1
1	1	1	1	1	1	mod	!mod	1	1	0	mod	!mod	mod

X^(n+1)

xy // zt	00	01	11	10
00	0	0	1	1
01	0	1	1	1
11	1	*	1	*
10	0	0	0	1

X^(n+1)=x*!z+x*!y+z*t+y*t

Y^(n+1)

xy // zt	00	01	11	10
00	0	0	1	0
01	supr	0	1	1
11	0	*	1	*
10	1	0	1	1

Y^(n+1)=x*y+x*t+x*z+!y*z*!t+!y*!z*t*supr

Z^(n+1)

xy // zt	00	01	11	10
00	0	0	1	Traseu
01	!supr	0	0	1
11	0	*	Mod	*
10	0	1	1	1

Z^(n+1)=x*y*!t + !x*y*z + x*!y*t + x*z*!t + x*z*mod +

+ x*!y*traseu + !y*!z*t*!supr

T^(n+1)

xy // zt	00	01	11	10
00	Start	0	0	!traseu
01	1	0	0	1
11	0	*	!mod	*
10	0	0	0	1

T^(n+1)=!y*!z*t + x*!y*z + x*!y*!traseu + y*z*t*!mod + !x*!y*!z*start

Dx

xy // zt	00	01	11	10
00	0	0	1	1
01	0	1	1	1
11	1	*	1	*
10	0	0	0	1

Dx=x*!y + x*!z + z*t + y*t

Jy

xy // zt	00	01	11	10
00	0	0	1	0
01	Supr	0	1	1
11	0	*	1	*
10	1	0	1	1

Jy=x*t + x*y + x*z + !y*z*!t + !y*!z*t*supr

Ky

xy // zt	00	01	11	10
00	1	1	0	1
01	!supr	1	0	0
11	1	*	0	*
10	0	1	0	0

Ky=!x*y + !y*!z*!t + !y*z*t + !x*!z*t*!supr

Dz

xy // zt	00	01	11	10
00	0	0	1	Traseu
01	!supr	0	0	1
11	0	*	Mod	*
10	0	1	1	1

Dz=x*!y*t + x*z*!t + x*y*!t + !x*y*z + x*z*mod + x*!y*traseu + !y*!z*t*!supr

Jt

xy // zt	00	01	11	10
00	Start	0	0	!traseu
01	1	0	0	1
11	0	*	!mod	*
10	0	0	0	1

Jt=x*!y*z + !y*!z*t + x*!y*!traseu + !x*!y*!z*start + x*z*t*!mod

Kt

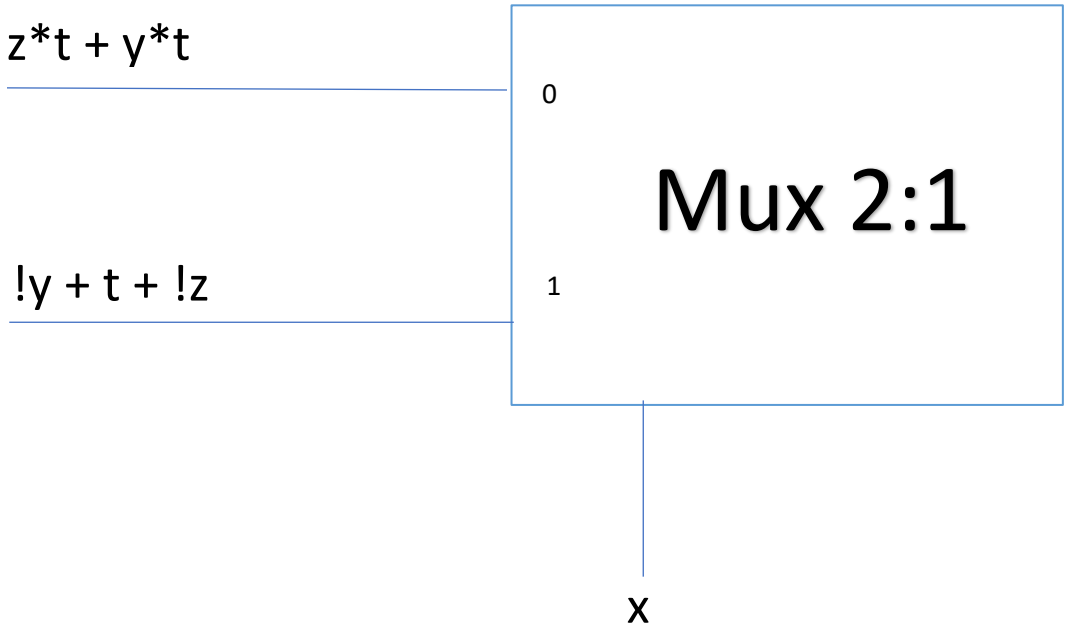
xy // zt	00	01	11	10
00	!start	1	1	Traseu
01	0	1	1	0
11	1	*	Mod	*
10	1	1	1	0

Kt=y*!z + !x*z + y*z*!t + x*y*z*mod + x*!z*!t*traseu +

+ !x*!z*!t*!start

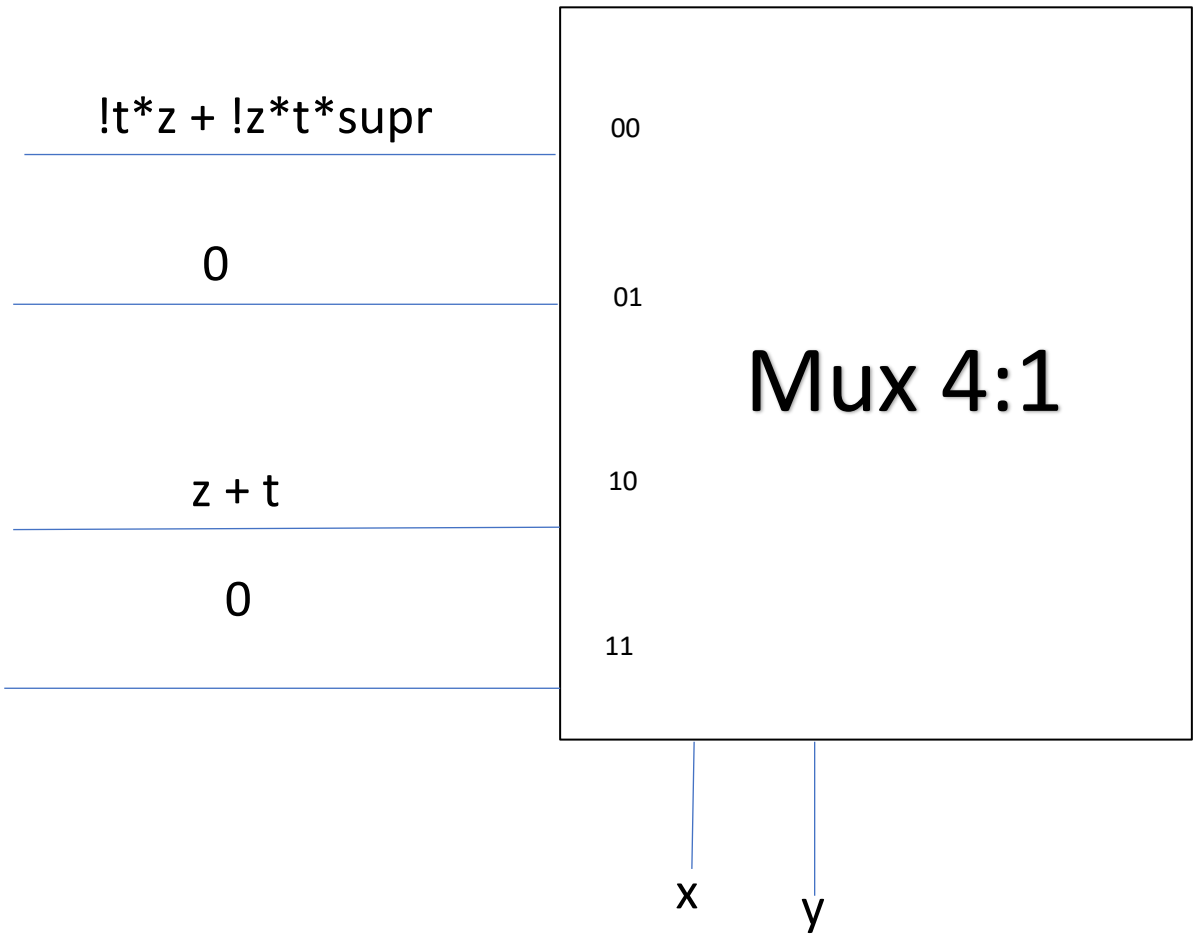
Dx implementarea cu MUX 2:1

xy // zt	00	01	11	10
00	0	0	1	1
01	0	1	1	1
11	1	*	1	*
10	0	0	0	1



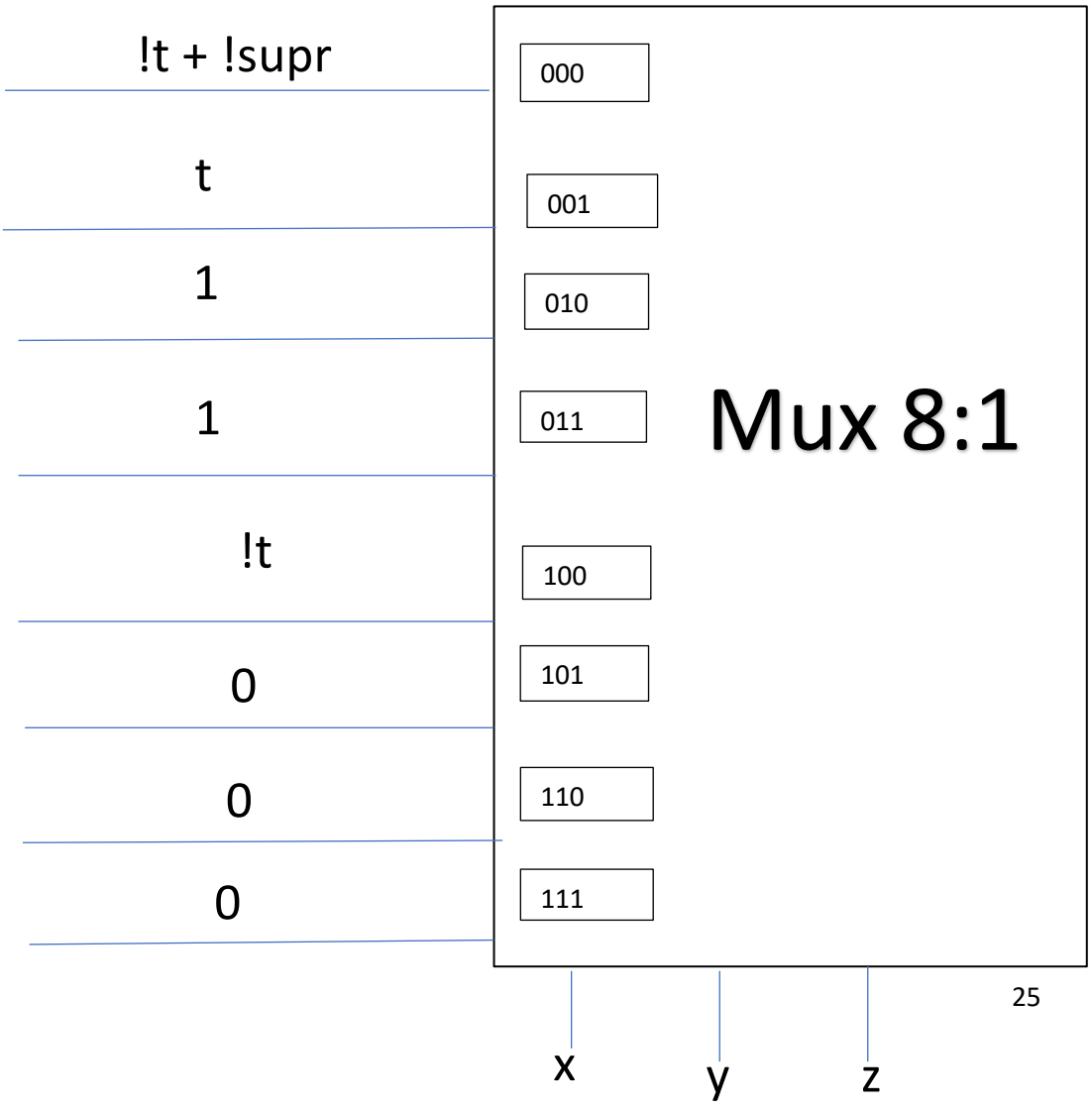
Jy Implementarea cu MUX 4:1

xy // zt	00	01	11	10
00	0	0	1	0
01	Supr	0	1	1
11	0	*	1	*
10	1	0	1	1



Ky Implementarea cu MUX 8:1

xy // zt	00	01	11	10
00	1	1	0	1
01	!supr	1	0	0
11	1	*	0	*
10	0	1	0	0



Implementarea circuitului