

Név.....

Neptun kód:.....

A házi feladatot egyedül, segítség nélkül oldottam meg. Aláírás:.....

## Digitális Technika 1. Házi feladat minta

**Beadás határideje: november 3.**

**Beadás időpontja:.....**

A feladatok megoldását a kinyomtatott feladatlagra, tollal írja le. Piros színt ne használjon!

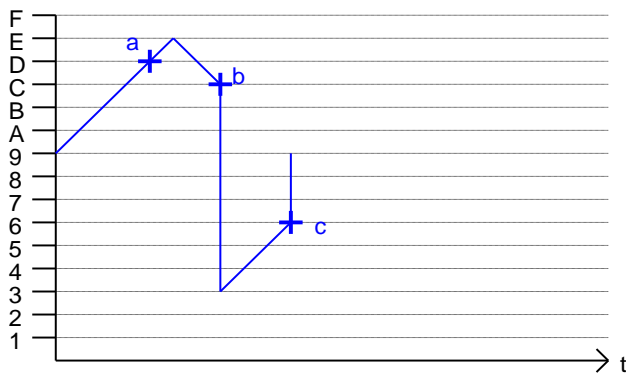
A házi feladat kódom: .....9,14,12,3,6.....

### A végeredmény összefoglalása:

A számlálónak ezek alapján így kell számolnia (hexadecimálisan megadva):

.....9,A,B,C,D,E,D,C,3,4,5,6,9,.....

1. A “varrógép minta” egy periódusa lerajzolva, megjelölve a figyelendő állapotokat:



Teendők:

- a. ...**irány vált. vezérlés**.....
- b. ..**betöltés és irány vezérlés**..
- c. ...**betöltés**.....

Alaphelyzet: JK Pr:..**RESET**.....

JK Cl ...**0**.....

2. Irányváltáshoz kikódolandó számláló állapotok és egyszerűsített kikódoltjuk:

Irány:	le	fel
Számláló állapot:	<b>D</b>	<b>C</b>
Egyszerűsített kikódoltja:	Kvez = <b>Q3.Q2.Q0</b>	Jvez = <b>Q3./Q1./Q0</b>

3. Betöltéshez kikódolandó számláló állapotok és egyszerűsített kikódoltjuk:

	első	második
Számláló állapot:	<b>C</b>	<b>6</b>
Egyszerűsített kikódoltja:	<b>Q3./Q1./Q0</b>	<b>/Q3.Q2.Q1</b>

4. Betöltő függvény: LD = ...**Q3./Q1./Q0./up\_cntr + /Q3.Q2.Q1 + RESET**.....

6. A kétféle betöltendő konstanst kiválasztó logikát vezérlő függvény:

sel = .....**RESET + /Q3.Q2.Q1**.....

5. Betöltendő 2 féle adat és a számláló betöltő bemeneteinek függvényei:

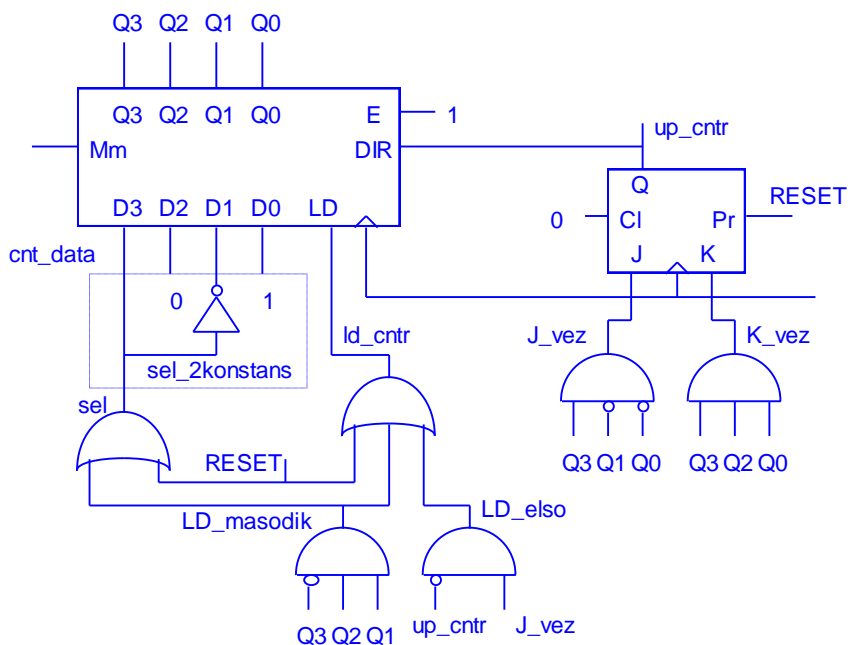
	adat	D3	D2	D1	D0
RESET alatt és .. <b>6</b> .. számláló állapot alatt	<b>9</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>
egyébként	<b>3</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>
számláló betöltő bemeneteinek függvényei		<b>sel</b>	<b>0</b>	<b>/sel</b>	<b>1</b>

A megvalósítandó ciklus és a számláló vezérlése:

A megvalósítandó ciklus és a számláló vezérlése:

[illegible]

***A megvalósítás részletes blokkvázlata:***



**RESET hatására LD=1, DATA=1001 és DIR=1 kell.**

A DIR-t egy J-K flip-floppal állítom elő. Ezt 1-be (fel) kell billenteni RESET hatására, amit az aszinkron Pr bemenettel oldok meg.  $Pr = \text{RESET}$ ,  $Cl = 0$ .

A DIR-t 1-ből 0-ba (le) kell billenteni, ha a számláló állapota:  $Q_3Q_2Q_1Q_0 = 1101$  (13), hogy a 14-es állapot alatt jövő órajel felfutó élére már lefele számoljon (a számláló és a JK felfutó élre billen). Az 1101 ugyan 2-szer fordul elő egy ciklusban, de a második előfordulás után közömbös az értéke, így nem okoz gondot, hogy akkor is 0-ba írjuk.

A teljes ciklus alatt a  $Q_3Q_2Q_0 = 111$  csak a 13 kódjában fordul elő, így annak kikódolásakor elég ezeket figyelni:

$$\mathbf{Kvez = Q_3.Q_2.Q_0}$$

A DIR-t 0-ból 1-be (fel) kell billenteni, ha a számláló állapota:  $Q_3Q_2Q_1Q_0 = 1100$  (12), hogy a betöltés utáni 3-as állapot alatti órajel felfutó élére már felfele számoljon. Az 1100 ugyan 2-szer fordul elő egy ciklusban, de az első előfordulás után is DIR=1 kell, hogy legyen, így nem okoz gondot, hogy akkor is 1-be írjuk.

A teljes ciklus alatt a  $Q_3Q_1Q_0 = 100$  csak a 12 kódjában fordul elő, így annak kikódolásakor elég ezeket figyelni:

$$\mathbf{Jvez = Q_3./Q_1./Q_0}$$

Betöltésre RESET alatt,  $Q_3Q_2Q_1Q_0 = 1100$  és  $Q_3Q_2Q_1Q_0 = 0110$  alatt van szükség. Az 1100 2-szer fordul elő a teljes ciklus alatt, de csak a második előfordulásakor kell betöltést vezérelni. Ezért valahogyan meg kell különböztetni ezeket egymástól. A megkülönböztetésre felhasználhatjuk a számláló DIR bemenetét vezérlő up\_cntr jelet, mivel az csak a második előfordulásakor 0 értékű:

$$\mathbf{LD\_első = Jvez./up\_cntr}$$

A teljes ciklus alatt a  $Q_3Q_2Q_1 = 011$  csak a 6 kódjában fordul elő, így annak kikódolásakor elég ezeket figyelni:

$$\mathbf{LD\_masodik = /Q_3.Q_2.Q_1}$$

$$\mathbf{ld\_cntr = LD\_első + LD\_masodik + RESET}$$

A számlálóba betöltendő adat a ciklus közben változik.

A számláló 6-os állapota vagy RESET alatt:  $D3D2D1D0 = 1001$ ,

Egyébként:  $D3D2D1D0 = 0011$

(A 6-os állapotot már kikódoltuk, ez az **LD\_masodik** jel.)

Mindkét esetben  $D0 = 1$ ,  $D2 = 0$ , tehát ezeket a konstansokat kell rájuk kötni.

A 2 konstansban csak  $D3$  és  $D1$  változik.

**LD\_masodik ill. RESET** alatt  $D3 = 1$  és  $D1 = 0$ , egyébként pedig  $D3 = 0$  és  $D1 = 1$

Előállítunk egy jelet (sel), amely a betöltendő konstanst előállító logikát fogja vezérelni.

**sel = LD\_masodik + RESET**

A **sel** jelet megfelelő előjellel rákötjük a változó adat bitekre ( $D3$  és  $D1$ ):

**$D3 = \text{sel}$ ,  $D1 = \text{/sel}$**

***A kapcsolási rajz Verilog nyelvű leírása az előre megadott és saját készítésű modulok felhasználásával, (vagy Digital Works-ből nyomtatott kapcsolási rajz):***

```
// top modul
// a cnt_data, LD_masodik, LD_elso, K_vez, J_vez jelek csak az idodiagrammos
// megjelenites miatt vannak kivezetve
module
varrogep_minta(RESET,CLK,Q,ld_cntr,up_cntr,cnt_data,LD_masodik,LD_elso,K_vez,Jvez);
input RESET;
input CLK;
output [3:0] Q;
output ld_cntr;
output up_cntr;
output [3:0] cnt_data;
output LD_elso;
output K_vez;
output LD_masodik;
wire RESET;
wire CLK;
wire [3:0] Q;
wire LD_masodik;
wire J_vez;
wire LD_elso;
wire K_vez;
wire up_cntr;
wire ld_cntr;
wire [3:0] cnt_data;

// ld_cntr vezerelese, or2(in1,in2,out)
or2 sel_ (LD_masodik,RESET,sel);

// 6 dekodolasa, and3(in1,in2,in3,out)
and3 LD_masodik_ (~Q[3],Q[2],Q[1], LD_masodik);

// 12 dekodolasa, and3(in1,in2,in3,out)
and3 J_vez_ (Q[3],~Q[1],~Q[0],J_vez);

// 12 dekodolasa le szamlalaskor, and2(in1,in2,out)
and2 LD_elso_ (J_vez,~up_cntr,LD_elso);

// 13 dekodolasa, and3(in1,in2,in3,out)
and3 K_vez_ (Q[3],Q[2],Q[0],K_vez);

// szaml.irany vezertes, JKpc(J, K, Pr, Cl, CLK, Q)
JKpc up_cntr_ (J_vez,K_vez,RESET,1'b0,CLK,up_cntr);

// betoltas vezertes, or3(in1,in2,in3,out)
or3 ld_cntr_ (LD_elso,LD_masodik,RESET,ld_cntr);

// betoltendo adat
sel_konstans sel_konstans_(sel,cnt_data);

// fel le szamlalo, counter_ud16(DATA, LD, E, DIR, CLK, Mm, Q)
counter_ud16 counter (cnt_data,ld_cntr,1'b1,up_cntr,CLK,,Q);
endmodule
```

Saját készítésű modulok:

A sel-től függően 3-at vagy 9-et tölt be.

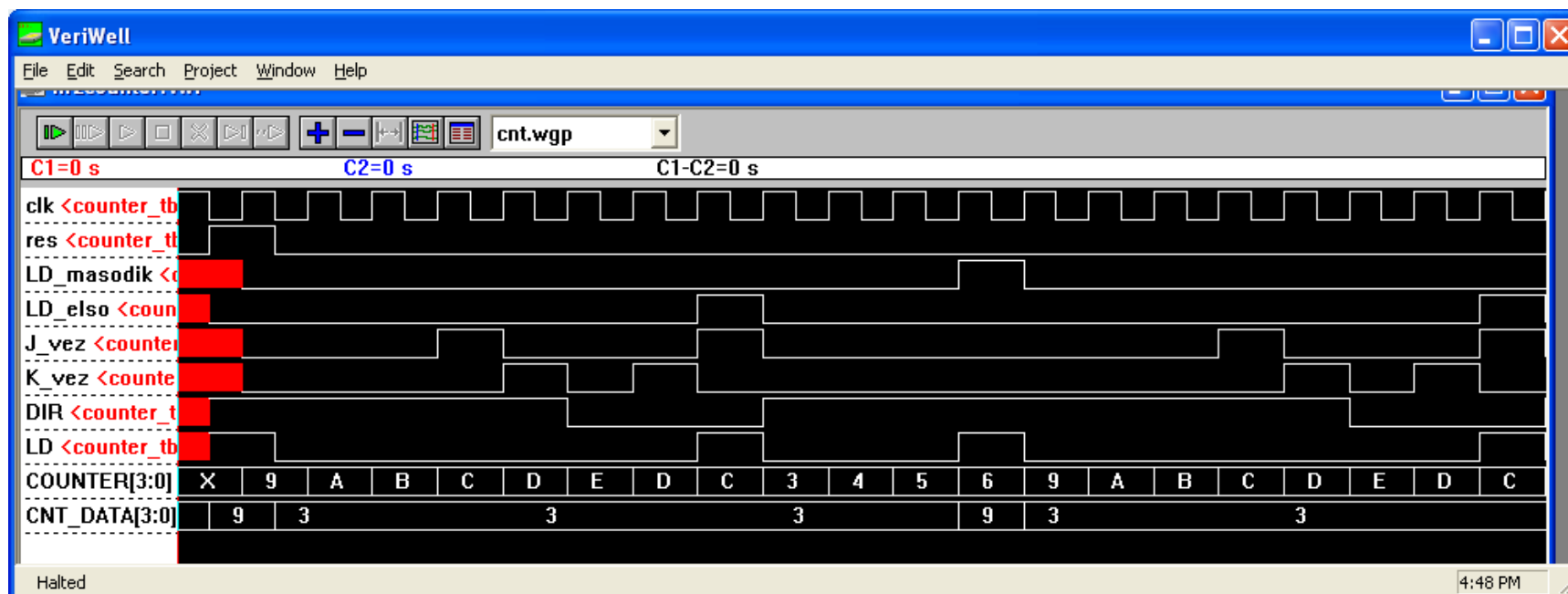
```
module sel_konstans (sel,out);  
output [3:0] out;  
input sel;  
  
// sel = 1, out = 3  
// sel = 0, out = 9  
  
assign out[0] = 1;  
assign out[1] = ~sel;  
assign out[2] = 0;  
assign out[3] = sel;  
  
endmodule
```

*Megjegyzés:*

*Az egyes jel nevek módosítása esetén a tb\_counter file-ban is módosítani kell a jelneveket és az idődiagramm megjelenítéséhez a group file-t is módosítani kell. Ez elkerülhető, ha a minta file-ban nem módosítja a jelneveke.*

*A szimuláció eredménye (Veriwell vagy Digital Works idődiagram) és az eredmény értékelése:*

*A szimulációnak egy ciklusnál hosszabb részletet kell mutatnia. (Digital Works esetén a számláló állapotait kézzel kell kipreparálni, vagyis a számláló állapotait az egyes kimeneti jelei alapján dekódolni és azt hexadecimális számként beírni.)*



Ezt összehasolítva a kívánttal: 9,A,B,C,D,E,D,C,3,4,5,6,9,... láthatóan megegyeznek.

Budapest, 2011. 10. 3.