20.....év ...hó ...nap

NÉV:.....Neptun kód:.....

A feladatokat önállóan, meg nem engedett segédeszközök használata nélkül oldottam meg:

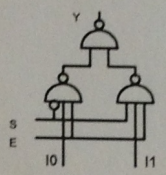
Olvasható aláírás:.....

Kedves Kolléga! A kitöltést a dátum, név és aláírás rovatokkal kezdje! Az alábbi kérdésekre a válaszokat - ahol lehet - mindig a feladatlapon oldja meg! A feladatok megoldása során a részletes kidolgozást nagyfeladatonként külön papíron végezze, (egyértelműen jelölje, hogy melyik lap melyik feladathoz tartozik) és ezeket a papírokat is adja be a dolgozatával! A kérdésekre a táblázatok vagy a pontozott vonalak értelemszerű kitöltésével válaszoljon, hacsak külön másként nem kérjük. Jó munkát!

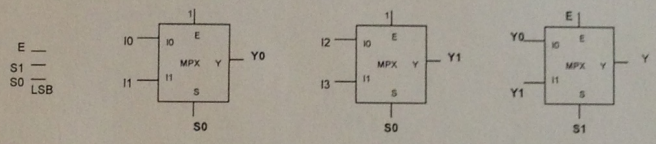
E:	
F1:	
F2:	
F3:	
Σ :	

Ellenőrző kérdések (20p)

E1. a. Rajzolja le egy 2/1-es engedélyezhető multiplexer belső felépítését, csak NAND kapukat használva! (Bemenetek: S, E, I0, I1 kimenet: Y) (2p)



b. Egészítse ki az alábbi 2/1-es engedélyezhető multiplexereket tartalmazó rajzot, hogy azok egy 4/1-es engedélyezhető multiplexertert valósítsanak meg! A bemenetek sorrendjét megadtuk, azon ne változtasson! (2p)



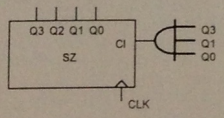
E2. Alább adott egy ismert sorrendi funkcionális elem Verilog leírása! Milyen elemről van szó? Adja meg a nevét és az UD, Cl, E bemenetek funkcióit! (3p)

```
module MiEz(input clk, input UD, input Cl, input E, output reg [3:0] q);
always @ (posedge clk)
if(Cl) q <= 0;
else
  if (E)
       if(UD)
            if(q != 9)
              q <= q + 1;
            else
              q <= 0;
       else
            if(q != 0)
              q <= q - 1;
            else
              q <= 9;
endmodule;
```

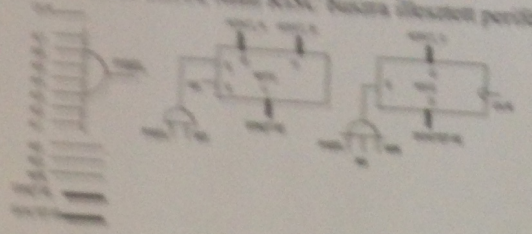
Funkc.elem neve: ...**számláló**..... Funkciók UD:...**irány**...Cl:...**törlés**.....E:...**engedélyezés**.....

Legfontosabb egyéb jellemzői:...**10-es modulusú, kétirányú**.....

E3. Készítsen az alábbi szinkron törölhető bináris felfele számlálóból 12-es modulusút! (2p)



B4. Adott az alábbi, Mini RISC buszra illesztett perifériás irányító kapcsolással. (3p)

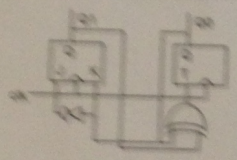


Egészítse ki, hogy a 0xfe-ről az ADAT_A-t, a 0xff-ről az ADAT_B-t lehessen beolvasni, a 0xff-re ADAT_C-t lehessen kiírni.

B5. A felsorolt állítások közül melyek igazak és melyek hamisak? Jelölje +-al az igaz, --al a hamis állításokat! (5p)

1.	1db 3/8-as multiplexorral, 1db inverterrel és a logikai konstansokkal tetszőleges 4 változós logikai függvény megvalósítható.	+
2.	Ha egy 4 bemenetű, hazárdmentesen megvalósított kombinációs hálózatot egy 4 bites bináris számlálóra kapcsolunk, a kombinációs hálózat kimenetein nem jelenhet meg hazárd.	-
3.	Egyfázisú órajelezési technika esetén a vezérlő és az adatstruktúra sorrendi elemei ugyanarra az órajel élre működnek.	-
4.	Az interrupt megszakíthatja a DMA folyamatot.	-
5.	A MINI RISC processzor kód és adatmemóriáját külön címbusz címzi.	+

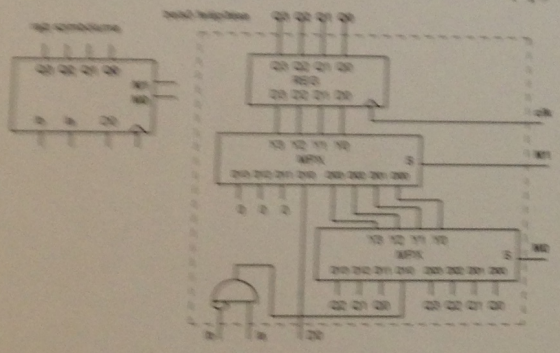
B6. a. Kapcsolási rajzával adott az alábbi szinkron sorrendi hálózat. Fejtse vissza a működését a sorok kitöltésével! Az első sort kitöltöttük. (3p)



Q1 Q0(t)	Q1Q0 (t+1)
00	10
01	**00**
10	**11**
11	**01**

F1. Adott egy a megszokottól kicsit eltérő kialakítású funkcionális elem belső kapcsolása és kapcsolási rajz szimbóluma. Oldja meg az alábbi feladatokat! (12p)

a. Az alábbi táblázatban adja meg, hogy M1M0-től függően mit csinál? A funkciót nem kell részletezni, elég a megnevezése. Pl. betölt (4p)

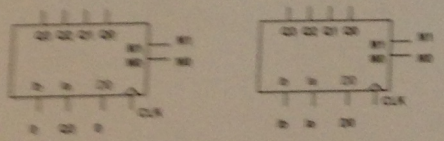


M1 M0	A funkcionális elem működése
0 0	**marad**
0 1	**shiftel**
1 0	**tölt**
1 1	**tölt**

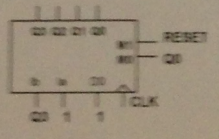
b. Milyen ismert speciális számlálók alakíthatók ki könnyen az elemből? (2p)

....gyűrűs, Johnson.....

c. Adott 2db fentiek szerinti funkcionális elem. Kaszkádosítsa őket, hogy a kaszkádosított egység úgy működjön, mint egy eredeti, de 8 bites! (Segítség: Ib, Ia közül az együket használják bemenetként (In), a másikra konstanst kötnek: Ib=0, In=Ia vagy In=Ib, Ia=1) (3p)

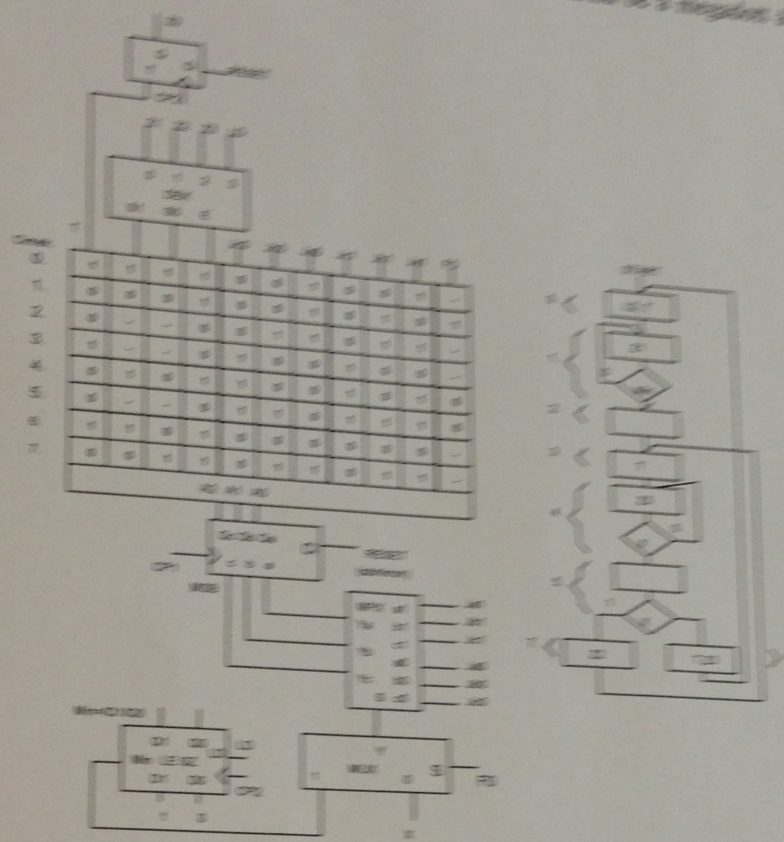


d. Alább felrajzoltunk egy fent megadott funkcionális elemet. Készítsen belőle olyan áramkört, mely a bekapcsolási RESET után egyszer az alábbi sorozatot adja ki: 0001, 0011, 0111, 1111, **1110, 1110,**.... (3p)

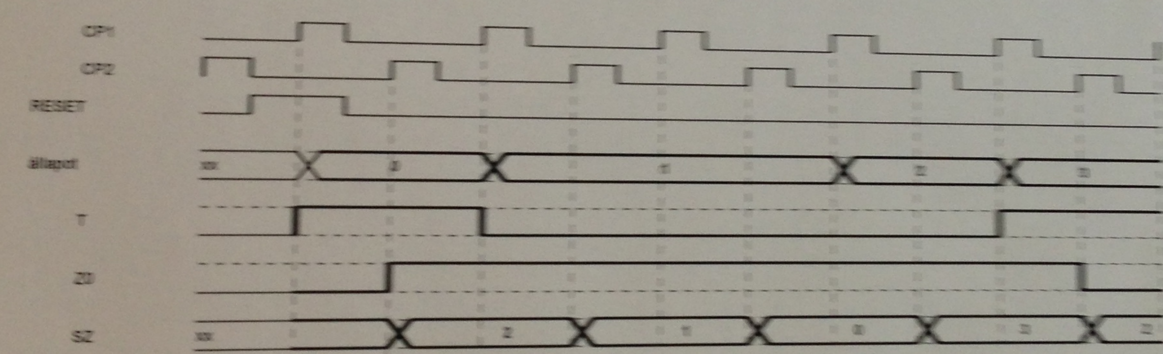


F2. Valósítsa meg az alábbi folyamatábrát a megadott mikroprogramozott vezérlővel! (12p)

a. Adja meg a mikroprogramot az alábbi táblázat kitöltésével! Tartsa be a megadott állapotkiosztást! (8p)



b. Rajzolja fel a hiányzó idődiagrammokat, ha X=1! (4p)



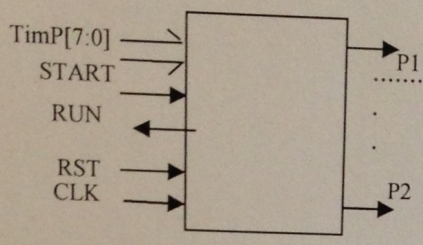
A MiniRISC CPU programozói kártyája Utasítások, periféria kiosztás (F3 feladathoz):

LOGSYS MiniRISC assembler direktívák és utasítások (v1.0 2013/04/12)			
Assembler direktívák		Konstans definíció	DEF azonosító érték
Kód szegmens	CODE	Kezdőcím megadása	ORG cím
Adat szegmens	DATA	Adat sz. inicializálása	DB érték1, érték2, ...
Adatmozgató [-]	**Aritmetikai** [Z,C,N,V]	**Léptetés** [Z,C,N]	**Ugrás** [-]
mov rX, rY	add rX, rY	sl0 rX	jmp paddr/(rY)
mov rX, #imm	add rX, #imm	sl1 rX	jz paddr/(rY)
mov rX, maddr	adc rX, rY	sr0 rX	jnz paddr/(rY)
mov rX, (rY)	adc rX, #imm	sr1 rX	jc paddr/(rY)
mov maddr, rX	sub rX, rY	asr rX	jnc paddr/(rY)
mov (rY), rX	sub rX, #imm	**Csere** [Z,N]	jn paddr/(rY)
Logikai [Z,N]	sbc rX, rY	swp rX	jnn paddr/(rY)
and rX, rY	sbc rX, #imm	**Szubrutin** [-]	jv paddr/(rY)
and rX, #imm	cmp rX, rY	jsr paddr	jnv paddr/(rY)
or rX, rY	cmp rX, #imm	jsr (rY)	**Operandusok**
or rX, #imm	**Forgatás** [Z,C,N]	rts	rX : r0-r15
xor rX, rY	rol rX	**Megszakítás** [-]	rY : r0-r15
xor rX, #imm	ror rX	sti	paddr: 0x00-0xFF
tst rX, rY	rlc rX	cli	maddr: 0x00-0xFF
tst rX, #imm	rrc rX	rti	imm : 0x00-0xFF

LOGSYS MiniRISC mintarendszer periféria regiszterkészlet (v1.0 2013/04/12)											
Cím (hexa)	Reg. név	Hozzáf. (W, R)	Alapért. (RST)	Bitek hozzárendelése, funkciója							
				7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
0x80	LD	R/W	0x00	LD7	LD6	LD5	LD4	LD3	LD2	LD1	LD0
0x81	SW	R	–	SW7	SW6	SW5	SW4	SW3	SW2	SW1	SW0
0x82	BT	R	–	0	0	0	0	BT3	BT2	BT1	BT0
0x84	AOUT	R/W	0x00	AO7	AO6	AO5	AO4	AO3	AO2	AO1	AO0
0x85	AIN	R	–	AI7	AI6	AI5	AI4	AI3	AI2	AI1	AI0
0x86	ADIR	R/W	0x00	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
0x88	BOUT	R/W	0x00	BO7	BO6	BO5	BO4	BO3	BO2	BO1	BO0
0x89	BIN	R	–	BI7	BI6	BI5	BI4	BI3	BI2	BI1	BI0
0x8A	BDIR	R/W	0x00	BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0
0x8C	TR	W	0xFF	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
	TM	R	0xFF	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0
0x8D	TC	W	0x00	TIE	TPS2	TPS1	TPS0	–	–	TREP	TEN
	TS	R	0x00	TIT	TPS2	TPS1	TPS0	0	TOUT	TREP	TEN
0x8E	UCS	R/W	0x00	RXNE	TXNF	RXCLR	TXCLR	RXIE	TXIE	RXEN	TXEN
0x8F	UDAT	R/W	–	D7	D6	D5	D4	D3	D2	D1	D0
0x90-0x93	DIG0-3	R/W	0x00	DP	G	F	E	D	C	B	A
0x94-0x98	COL0-4	R/W	0x00	0	ROW7	ROW6	ROW5	ROW4	ROW3	ROW2	ROW1

F3. Periféria tervezés és illesztés mikroprocesszoros buszra. (16p)
Tervezzen perifériát, amely a P1 és P2 kimenetein időben TimP órajelciklussal késleltetve egy-egy pulzust ad ki, P1 megelőzi P2-t.
A TimP 8 bites bemeneten megadható értékek 2 és 250 közé esnek. A periféria működése a START parancsra indul, és a működési ideje alatt a RUN=1 jellel jelez. P2 kiadásakor a RUN jel megszűnik. A működés közben a státuszregiszter D0 bitjén a RUN bit jelzi az aktív állapotot, ezalatt újabb START jel hatástalan.
A RST jel alaphelyzetbe állítja a perifériát (RUN = 0, indításra kész.)

a. Az alábbi blokkvázlat mellett adja meg a fent leírt periféria ***Verilog interfész*** specifikációját! (2p)



Periféria Verilog interfész specifikációja:

.....module Pulse(input RST, input clk, input START, input [7:0] TimP, output RUN, output P1, output P2);

b. Milyen ***funkcionális egységekből*** építené fel a tervezendő egységet? Sorolja fel a nevüket és legfontosabb tulajdonságaikat! (Pl. 4 bites összeadó,… stb.) (2p)

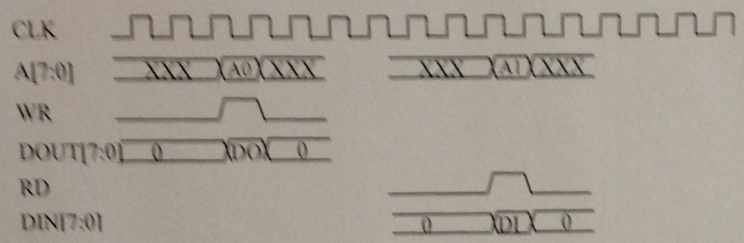
8 bites tölthető lefele számláló végérték jelzéssel vagy 8 bites törölhető felfele számláló és 8 bites egyenlőség komparátor

c. Tervezze meg a szükséges részáramköröket Verilog nyelven! ***Adja meg a belső részegységek, logikai hálózatok Verilog specifikációját!*** (6p)

```
reg P1;
reg P2;
reg [7:0] cnt;                    // Időmérő
wire RUN_END;
reg RUN;                          // működés jelző
reg RUN_OLD;

assign RUN_END = (cnt == 0x00);  // 0 végérték jelzés

//működés indítás/leállítás (RUN állítása)
always @ (posedge clk)
  if(RST | RUN_END)  RUN <= 1'b0;
  else if(START)     RUN <= 1'b1;

// tölthető engedélyezhető (RUN) lefele számláló
always @ (posedge clk)
  if((rst | (START & !RUN)) cnt <= TimP;         // futás alatt nem indítható újra
  else if(RUN) cnt <= cnt-1;

//RUN_OLD a RUN 1 órajellel késleltetve a felfutó és lefutó él figyeléshez P1,P2 miatt
always @ (posedge clk)
  if(RST)  RUN_OLD <= 1'b0;
  else RUN_OLD <= RUN;

// P1 előállítása
always @ (posedge clk)
  if(RST) P1 <= 1'b0;
  else if(!RUN_OLD & RUN) P1 <= 1'b1;    // P1 a RUN felfutó élnél
                     else P1 <= 1'b0;

// P2 előállítása
always @ (posedge clk)
  if(RST) P2 <= 1'b0;
  else if(RUN_OLD & !RUN) P2 <= 1'b1;  // P2 a RUN lefutó élnél
                     else P2 <= 1'b0;
```

P1, P2 még sokféleképpen előállítható. Pl. állapotgéppel, mely RUN-t figyeli (ez ugyanaz mint a fenti, csak másképp leírva) vagy cnt két állapotának kikódolásával stb.

d. Illessze az egységet egy mikroprocesszoros rendszerbuszhoz. A busz egy egyszerű, szinkron, áramkörön belüli busz, melynek jelei: A[7:0], DIN[7:0], DOUT[7:0], RD, WR, IRQ, CLK, RST. Az adatbusz irányának értelmezése a CPU szerinti (lásd diagram!). A működést a CLK órajel felfutó éle ütemezi, a buszciklusok a RD és WR jelek hatására egy órajel ciklus alatt végrehajtódnak.



A periféria Báziscíme 0xC0. Az illesztőegység tartalmazzon 3 regisztert: PAR parancsregisztert (D7:ITEN, D0:START), címe Bázis+0. Az STA státuszregisztert (D7:ITRQ, D0:RUN), címe Bázis+1. A TimP párhuzamos adatregisztert, címe Bázis+2. ***Adja meg a tervezendő periféria programozói interfészét, az alábbi táblázat kitöltésével!*** (1p)

Reg_cím (hex)	Név, FUN	Mód, WR/RD	Regiszter bitek szerepe							
			D7	D6	D5	D4	D3	D2	D1	D0
0xc0	PAR	WR	ITEN							START
0xc1	STA	RD	ITRQ							RUN
0xc2	TimP	ER	TIMP7	TIMP6	TIMP5	TIMP4	TIMP3	TIMP2	TIMP1	TIMP0

e. Tervezze meg a címdekódert és a parancsjeleket előállító és RUN lefutó éle után esetén interruptot kérő áramkört! A parancs regiszter START bitjét nem kell tárolni, de a törléshez az íráskor 1 értékűnek kell lennie. Az interrupt kérést a státusregiszter beolvasása törölje automatikusan! ***Adja meg a Verilog HDL nyelvű specifikációt!*** (3p)

```
parameter BASEADDR = 8'hC0;
wire psel, wr0, rd1, wr2;
// A periféria kiválasztó jele
assign psel = ((addr >>2) == (BASEADDR >>2));

// Írás és olvasás parancsjelek
assign wr0 = psel & (addr[1:0] == 2'b00) & wr;   // WR_PAR
assign rd1 = psel & (addr[1:0] == 2'b01) & rd;   // RD_STA
assign wr2 = psel & (addr[1:0] == 2'b10) & wr;   // WR_TimP

// A parancsregiszter bitjei
reg iten, IT_FLAG;
always @ (posedge clk)
   if (rst)       iten <= 1'b0;
   else if (wr0)  iten <= dout[7];

always @ (posedge clk)
   if (rst)       en <= 1'b0;
   else if (wr0)  en <= dout[0];

assign START = wr0 & dout[0];                    // Dinamikus parancsbit

assign din = {iten, 6'b0, RUN} & rd1; // STA read

// Interrupt kérés (IT_FLAG) állítása
always @ (posedge clk)
   if(rst | rd1)  IT_FLAG <= 1'b0;
   else if(P1) IT_FLAG <= 1'b1;

assign itrq = iten & IT_FLAG;                    // Ha ITEN = 1 és IT_FLAG, akkor ITRQ
```

f. Írjon egy rövid assembly programot, ami a perifériát felkészíti 128db órajel távolságú impulzus pár kiadására! A RUN megszűnésekor (P2 kiadásakor) a periféria megszakításkéréssel jelezzen. A program elkészítéséhez használható a 3. oldalon található MiniRISC programozói kártya. (2p)

```
;*********************************************************
;* ASM kód
;*********************************************************
DEF PAR  0xC0                   ; PARANCS regiszter (írható)
DEF STA  0xC1                   ; STATUSZ regiszter (olvasható)
DEF TimP 0xC2                   ; TimP   regiszter (írható)
    CODE
setup23:
        mov     r0, #128        ; Az impulzus hossz kiírása
        mov     TimP, r0        ; a TimP regiszterbe
        mov     r0, #0x81       ; Parancskód: ITEN és START
        mov     PAR, r0         ; Parancs kiadása
```

Max. pontszám: 60 pont. Rendelkezésre álló idő: 100 perc.