

Digit gyűjt

RAIK

2021.09.07

1

RAIKOVICH TAMÁS

rtamas @mit.bme.hu

F1	192	256		152 : 2	0
	192	128	1	76 : 2	0
	64	64	1	38 : 2	0
	0	32	0	19 : 2	1
				9 : 2	1

F2	$192_{10} = 11000000_2 = C0_{16} = 0 \times C0$	4 : 2	0
	$152_{10} = 10011000_2 = 98_{16} = 0 \times 98$	2 : 2	0
		4	0

F3 q₂

Dec	Bin 128 64 32 16 8 4 2 1	2's C -128 64 32 16 8 4 2 1
20	00010100	11101100
14	00001110	11110010
42	00101010	11010110
111	01101111	10010001

20	16	1	42	32	1
4	8	0	10	16	0
4	4	1	10	8	1
0	2	0	2	4	0
0	1	0	2	2	1
			0	1	0

$-x = x$ kettes komplementum
 $\bar{x} + 1$

b, a+b

$$\begin{array}{r} 00010100 \\ + 00001110 \\ \hline 00100010 \end{array}$$

a+d

$$\begin{array}{r} 00010100 \\ + 01101111 \\ \hline 10000011 \end{array}$$

előjel nélküli: 131 ✓

- csak azonos előjelű
bármelyikre lehet átváltás

- kimenet előjele ellentétes a
bármelyiké

kettős komplement: -125 X

↓

átváltás

d-c

$$\begin{array}{r} 01101111 \\ + 11010110 \\ \hline 101000101 \end{array}$$

F_{1A}

$$A + \bar{A}B = A + B$$

$$A + \bar{A}B = \underbrace{(A + \bar{A})}_{1} (A + B) = 1(A + B) = A + B$$

F_{2B}

$$Y_2 = \overline{(AB + CD)(\bar{A}\bar{C} + \bar{B}\bar{D})} = \overline{AB\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D}} = \overline{AB\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D}}$$

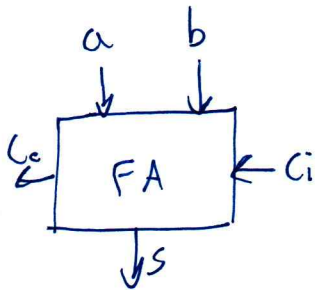
$$= \overline{AB\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D}} + \overline{\bar{A}B\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D}} = (\bar{A} + \bar{B} + C + D) \cdot (\bar{A} + \bar{B} + D) = \bar{A} + \bar{A}\bar{B} + \bar{A}C + \bar{A}D + \bar{B}\bar{C} + \bar{B}D + C + D = \bar{A} + \bar{B} + C + D$$

F₃

Teljes összeadó tervezés

$$\begin{array}{r} 00110 \leftarrow c_0 \\ 1001 \leftarrow a \\ + 0011 \leftarrow b \\ \hline 1100 \end{array}$$

1 bites teljes összeadó: elég a bináris teljes összeadás helyettesítésére tartozó műveletet



a	b	c _i	c _o	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

~~X XOR Y = X\bar{Y} + \bar{X}Y~~

X XOR Y = X\bar{Y} + \bar{X}Y

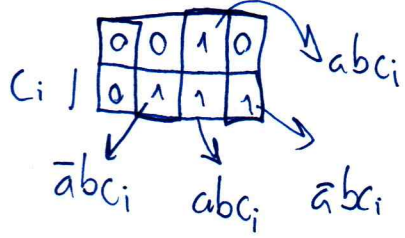
X XOR Y = \bar{X}\bar{Y} + XY

$$s(a, b, c_i) = \bar{a}\bar{b}c_i + \bar{a}b\bar{c}_i + a\bar{b}\bar{c}_i + abc_i = (\bar{a}\bar{b} + ab)c_i + (\bar{a}b + a\bar{b})\bar{c}_i =$$

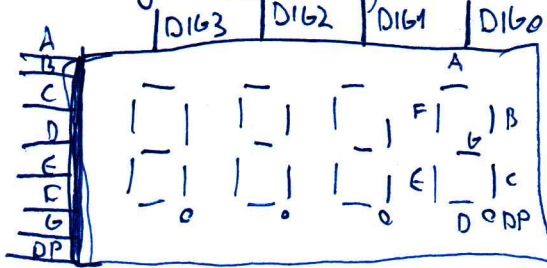
$$= (\bar{a} \text{ XOR } b) \cdot c_i + (a \text{ XOR } b) \bar{c}_i = (\overline{a \text{ XOR } b}) \cdot c_i + (a \text{ XOR } b) \bar{c}_i = a \text{ XOR } b \text{ XOR } c_i$$

$$c_o(a, b, c_i) = \bar{a}bc_i + a\bar{b}c_i + ab\bar{c}_i + abc_i = (\bar{a}bc_i + abc_i) + (a\bar{b}c_i + abc_i) + (ab\bar{c}_i + abc_i) = (\bar{a} + a)bc_i + (\bar{b} + b)ac_i + (\bar{c}_i + c_i)ab = ab + ac_i + bc_i$$

Karnaugh-table b



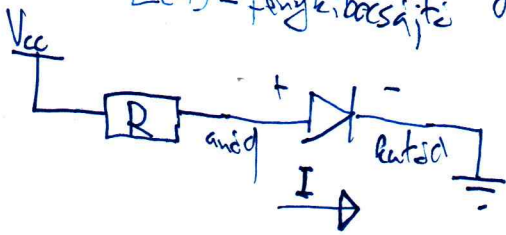
7 szegmenses kijelző vezérlés



32 led → 32 FPGA láb

SCHOR

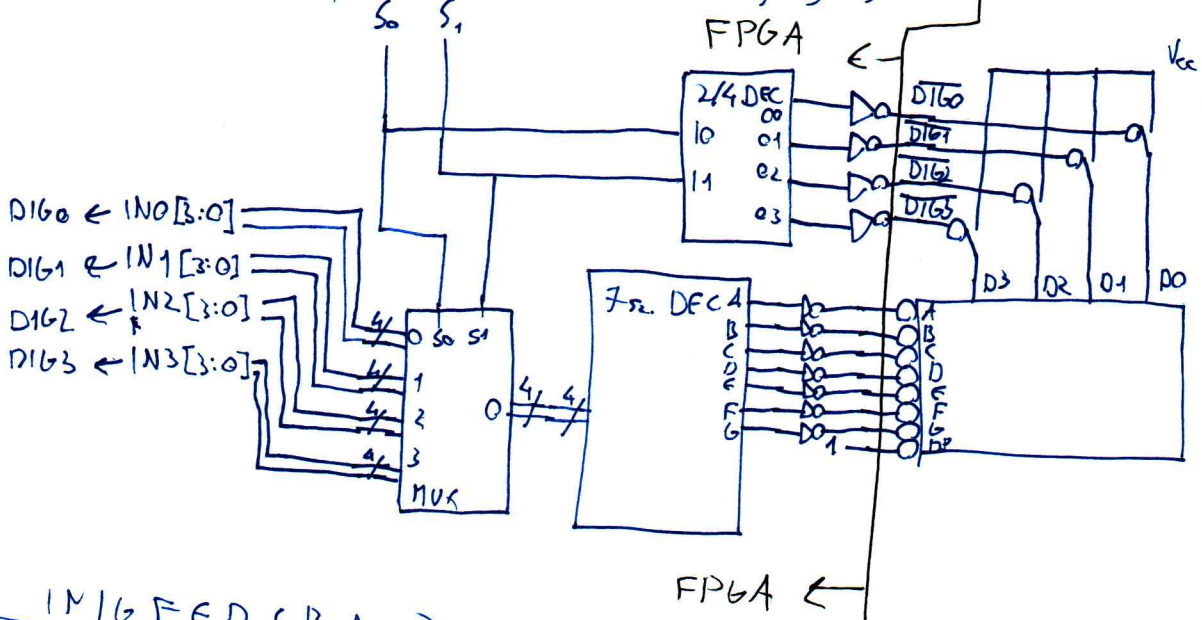
LED = fénykibocsátó dióda → félvezető



Digitkiválasztó jel: közös anód
Szegmenskiválasztó jel: közös katód

Digit kiválasztó jel sorozat

S [1:0] = 00, 01, 10, 11



DIG0 ← IN0[3:0]
DIG1 ← IN1[3:0]
DIG2 ← IN2[3:0]
DIG3 ← IN3[3:0]

IN	G	F	E	D	C	B	A
0	0	1	1	1	1	1	1
1	0	0	0	0	1	1	0
2
...
F

7 szegmenses dekodler



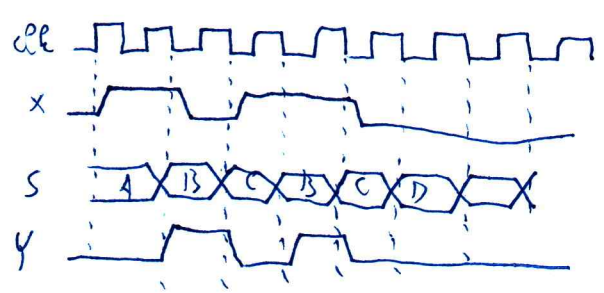
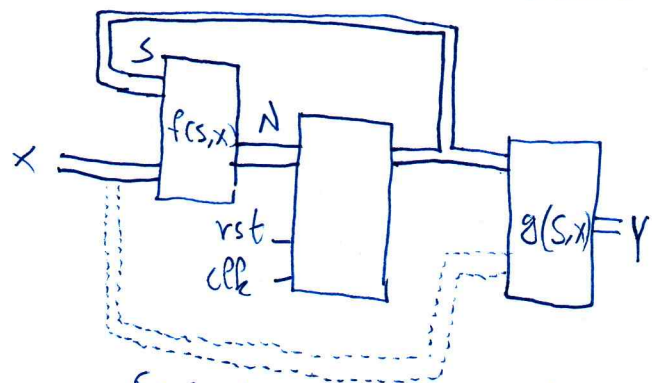
D=0-it gas

2021.05.
28.
1

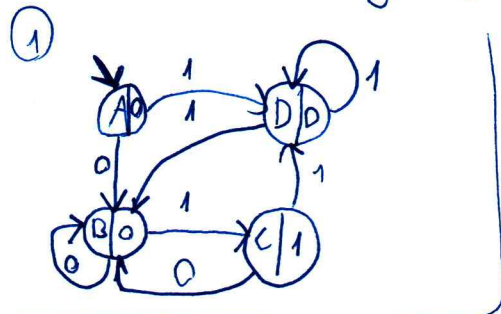
FSM tervezés

↑ felhívás DE detektálása × bemenet → 1 órajel periódus
 ↓ pulzus a kimeneten

csak Moore modell

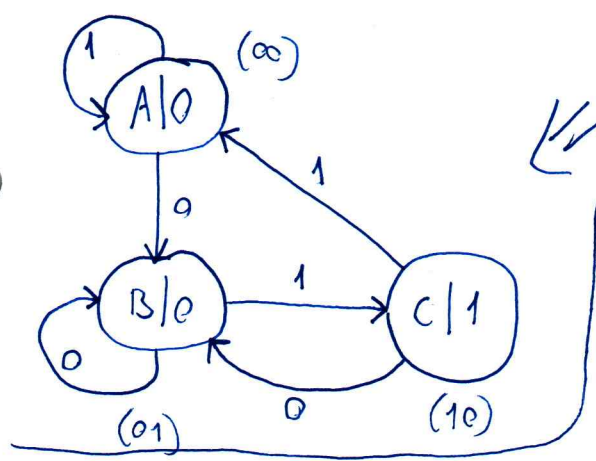


csak Mealy modell esetén



② Állaptoptimalizálás - az ekvivalens állapotos egy állapotba vonható össze
 $q_i \equiv q_j$ ha,
 1. $\forall x$ bemenete $g(q_i, x) = g(q_j, x)$
 2. $\forall x$ bemenete $f(q_i, x) = f(q_j, x)$

⇐ A=D ⇒ A



③ Állapthidolás, kódolt állapottábla
 f és g igazságtáblája

S[1:0]	Y	N[1:0]	
		x=0	x=1
A 00	0	B 01	A 00
B 01	0	B 01	C 10
C 10	1	B 01	A 00
- 11	?0	?A00	?A00

Grünfelder állapot
 egyszerűbb függvények Y és N don't care
 biztonságosabb működés
 Y: inaktív
 N: készenléti

④ f és g függvények

$$Y = g(s) = S[1] \cdot \overline{S[0]}$$

$$N[1] = \overline{S[1]} \cdot S[0] \cdot x$$

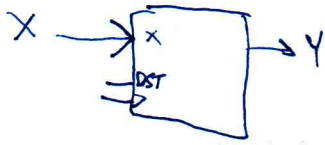
$$N[0] = \overline{S[1]} \cdot \overline{x} + S[0] \cdot \overline{x}$$

N[1]	S[0]	S[1]	
x	0	1	0
	0	1	0

N[0]	S[0]	S[1]	
x	0	1	0
	0	1	0



FSM tervezés



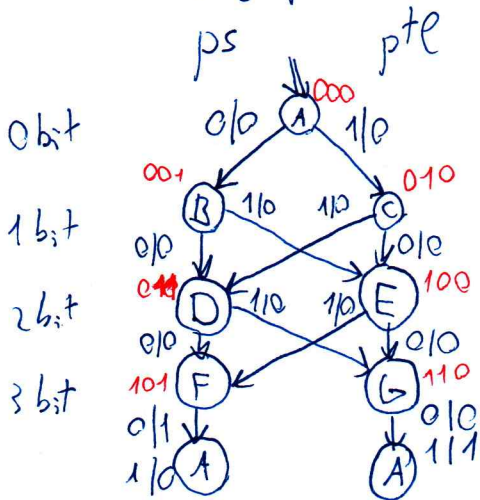
Az x bemenetre ciklusosan jöhetnek érkeznek 4 bites számok. Az Y kimenet az utolsó bit beérkezésével egy időben jelzi, ha a 4 bites szám osztható 3-mal v. 5-tel

- | | | | |
|-----|------|------|------|
| → 0 | 0000 | → 8 | 1000 |
| 1 | 0001 | → 9 | 1001 |
| 2 | 0010 | → 10 | 1010 |
| → 3 | 0011 | 11 | 1011 |
| 4 | 0100 | → 12 | 1100 |
| → 5 | 0101 | 13 | 1101 |
| → 6 | 0110 | 14 | 1110 |
| 7 | 0111 | → 15 | 1111 |

$Y = g(S, X) \rightarrow$ Mealy modell

↳ páros számú egyes

① Állapotgraf



② Állapotoptimalizálás

③

Áll. áll S[2:0]	Kör. áll / kimenet (Y)	
	x=0	x=1
A 000	B/0 001/0	C/0 010/0
B 001	D/0 011/0	E/0 100/0
C 010	F/0 100/0	D/0 011/0
D 011	F/0 101/0	G/0 110/0
E 100	G/0 110/0	F/0 101/0
F 101	A/1 000/1	A/0 000/0
G 110	A/0 000/0	A/1 000/1
- 111	A/0 000/0	A/0 000/0

④ Következő állapot és kimeneti függvények

$Y = S[2] \overline{S[1]} \overline{S[0]} \overline{x} + S[2] S[1] \overline{S[0]} x$

$$\begin{array}{c}
 \overline{s[0]} \quad \overline{s[1]} \\
 \hline
 N[2] \quad \begin{array}{c|c|c|c}
 0 & 0 & 1 & 1 \\
 \hline
 s[2] & 1 & 0 & 0 \\
 \hline
 & 1 & 0 & 0 \\
 \hline
 x & 0 & 1 & 1 \\
 \hline
 & 0 & 1 & 1 \\
 \hline
 & 0 & 1 & 1 \\
 \hline
 & 0 & 1 & 1 \\
 \hline
 \end{array}
 \end{array}$$

$$\begin{aligned}
 N[2] &= \overline{s[2]} \cdot \overline{s[1]} \cdot \overline{x} + \\
 &+ \overline{s[2]} \cdot \overline{s[1]} \cdot \overline{s[0]} + \\
 &+ \overline{s[2]} \cdot \overline{s[0]} \cdot x
 \end{aligned}$$

$$\begin{array}{c}
 \overline{s[0]} \quad \overline{s[1]} \\
 \hline
 N[1] \quad \begin{array}{c|c|c|c}
 0 & 1 & 0 & 0 \\
 \hline
 s[2] & 1 & 0 & 0 \\
 \hline
 & 0 & 0 & 0 \\
 \hline
 x & 1 & 0 & 1 \\
 \hline
 & 1 & 0 & 1 \\
 \hline
 & 1 & 0 & 1 \\
 \hline
 & 1 & 0 & 1 \\
 \hline
 \end{array}
 \end{array}$$

$$\begin{aligned}
 N[1] &= \overline{s[2]} \cdot \overline{s[0]} \cdot x + \\
 &+ \overline{s[2]} \cdot \overline{s[1]} \cdot x + \\
 &+ \overline{s[2]} \cdot \overline{s[1]} \cdot \overline{s[0]} \cdot x + \\
 &+ \overline{s[2]} \cdot \overline{s[1]} \cdot \overline{s[0]} \cdot \overline{x}
 \end{aligned}$$

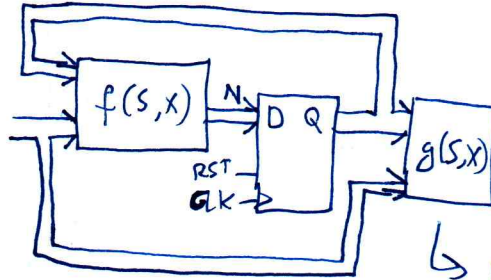
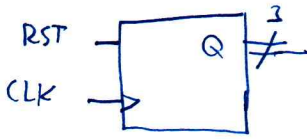
$$\begin{array}{c}
 \overline{s[0]} \quad \overline{s[1]} \\
 \hline
 N[0] \quad \begin{array}{c|c|c|c}
 1 & 1 & 1 & 0 \\
 \hline
 s[2] & 0 & 0 & 0 \\
 \hline
 & 1 & 0 & 0 \\
 \hline
 x & 0 & 0 & 1 \\
 \hline
 & 0 & 0 & 1 \\
 \hline
 & 0 & 0 & 1 \\
 \hline
 & 0 & 0 & 1 \\
 \hline
 \end{array}
 \end{array}$$

$$\begin{aligned}
 N[0] &= \overline{s[2]} \cdot \overline{s[1]} \cdot \overline{x} + \\
 &+ \overline{s[2]} \cdot \overline{s[0]} \cdot \overline{x} + \\
 &+ \overline{s[2]} \cdot \overline{s[1]} \cdot \overline{s[0]} \cdot x + \\
 &+ \overline{s[2]} \cdot \overline{s[1]} \cdot \overline{s[0]} \cdot x
 \end{aligned}$$

Simuláció tervezés

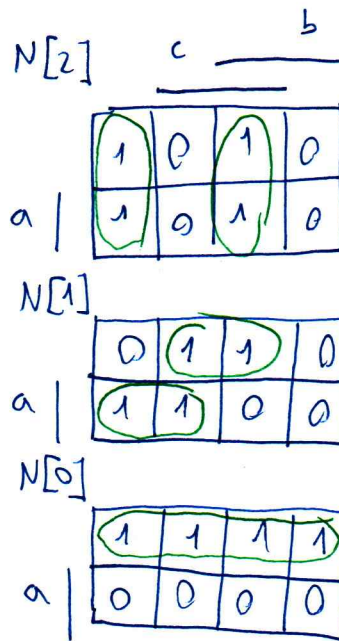
0521374609...

- 1, Kézi tervezés
- 2, Standard verilog Pérdás
- 3, Indirekt tervezés



① Kézi tervezés

Áll. jel. S[2:0] = {a,b,c}	Kén. áll. N[2:0]
0 000 START	5 101 E
1 001 A	3 011 C
2 010 B	1 001 A
3 011 C	7 111 G
4 100 D	6 110 F
5 101 E	2 010 B
6 110 F	0 000 START
7 111 G	4 100 D



$$N[2] = \overline{a}b + a\overline{b} = b \oplus c$$

$$N[1] = \overline{a}c + a\overline{b}$$

$$N[0] = \overline{a}$$

↳ ① és ② esetére $Y = g(s,x) = S$

② Standard Verilog Pérdás

megszélesítve Pérdás → case

```

localparam START = 3'd0;
localparam A = 3'd1;
localparam B = 3'd2;
:
localparam G = 3'd7;

reg [2:0] s,h;
always @ (posedge clk)
if (rst) s <= START
else s <= h
    
```

```

always @ (*)
case(s)
START: h <= E;
A : h <= C;
B : h <= A;
C : h <= G;
D : h <= F;
E : h <= B;
F : h <= START;
G : h <= D;
default: h <= START;
endcase
    
```

③ Indirekt tervezés bináris számok + szétbontás

cnt [2:0]	out [2:0]
0 000	0 000
1 001	5 101
2 010	2 010
3 011	1 001
4 100	3 011
5 101	7 111
6 110	4 100
7 111	6 110

```
always @ (posedge clk)
  if (rst) cnt <= 3'd0;
  else cnt = cnt + 1;
```

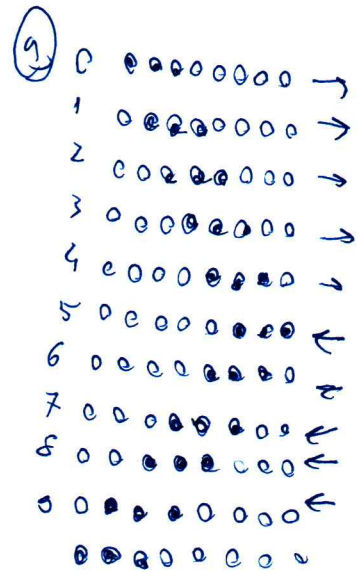
```
always @ (*)
  case (cnt)
    3'd0: out = 3'd0;
    3'd1: out = 3'd5;
    :
    3'd7: out = 3'd6;
    default: out = 3'd0;
  endcase
```

● Knight Rider futtatás

8 LCD, kezdetben bal szélső 3 LED

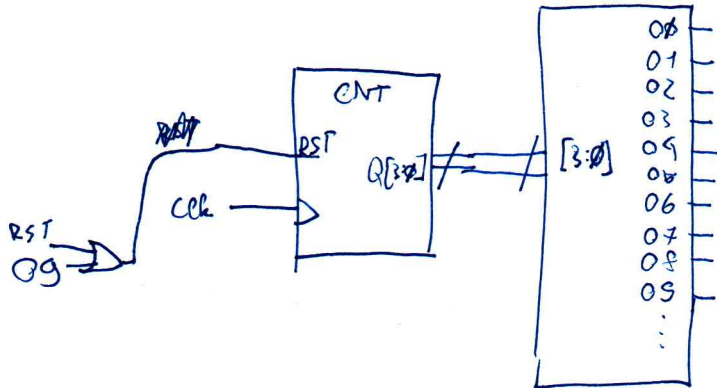
a) Egységnyi számolás

b) Shiftregiszterrel



4 bites számláló (10 állapot van)

- töredezhető: 9-es állapotban törlés
- felfelé számol



LD0 = 05

LD1 = 04 + 05 + 06

LD2 = 03 + 04 + 05 + 06 + 07

LD3 =

LD4 =

LD5 =

LD6 =

LD7 =

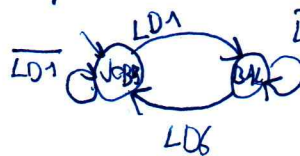
● 8 bites shift regiszter

- kétirányú
- töredezhető

környítés? - végállapotban már ellenkezős lejtetési irány kell

↳ irányváltás a végállapot elött: állapotban → jobbra lejtetés: LD1

balra lejtetés: LD6



reg dir;
always @(posedge clk)
if (rst)
ed <= 8'b00000000;

else

if (dir)

ed <= {ed[6:0], 1'b0}

else

ed <= {1'b0, ed[7:1]}

always @(posedge clk)

if (rst || ed[6])

dir <= 1'b0;

else if (ed[1])

Vérilog leírás

reg [3:0] cnt;

always @(posedge clk)

if (rst || (cnt == 4'd9)) cnt <= 4'd0;
else
cnt += 4'd1;

always @(*)

case (cnt)

4'd0: ed <= 8'b11100000;

4'd1: ed <= 8'b01110000;

...

default: ed <= 8'b00000000;

endcase

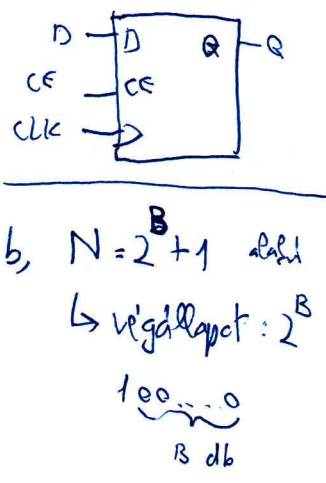


① Több órajel használata: NEM GUT!!

Ha lehet, akkor el kell kerülni

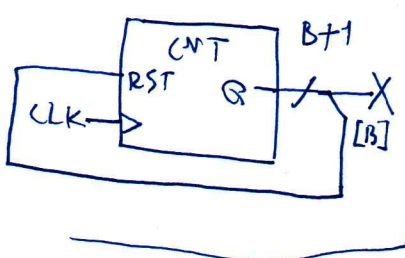
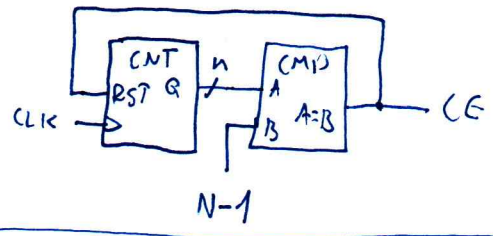
- FPGA: globális órajel vonalát szinte Pinitálat
- órajel tartományok illesztése
- Flip Flop-ok monostabil állapotok → szinkronizáció!

② Egy órajel használata: órajel engedélyező bemenet használata

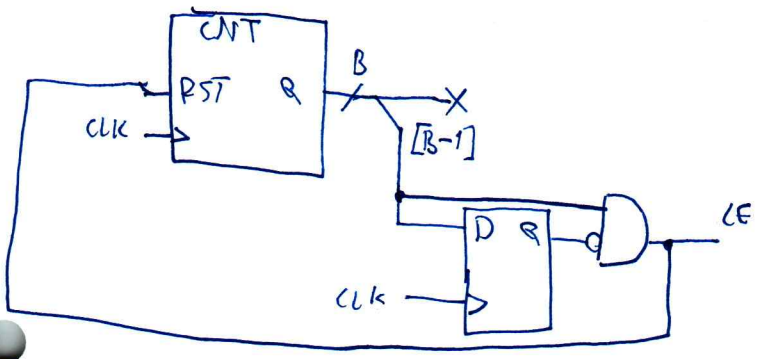
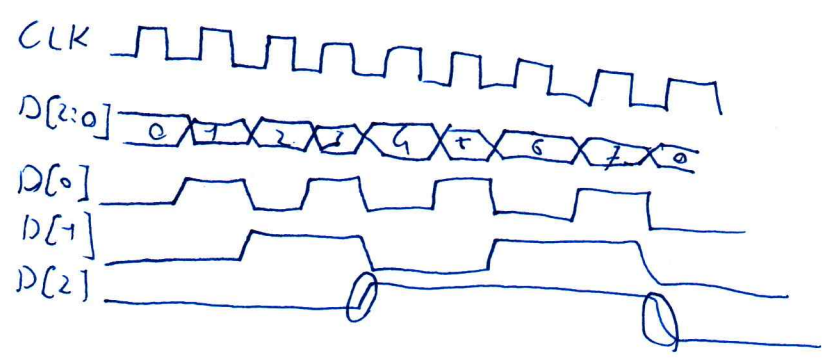


Órajel engedélyező jel előállítás

a, a rendszerórajel - frekvenciát N -ed részre osztjuk
 ↳ bitszám $\lceil \log_2 N \rceil = h$ ← felső egészrész

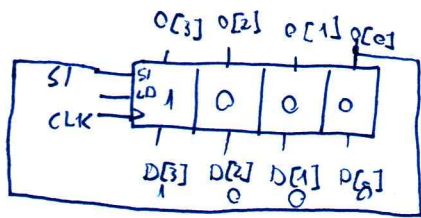


c, $N = 2^B$ darab



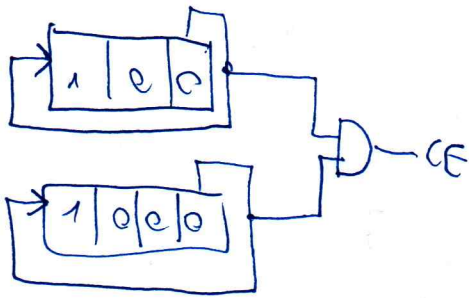
d, Shiftregister dalam

N -ed valid frekuensi osilasi $\rightarrow N$ bites SHR



CSG? Bis N eseten basn'P,uk

e, 2SHR: N is M bitesak



CSG? Bis M

$R_n(N, M) = 0 \Rightarrow$ ositis: $N \cdot M$

LNKO számológép

- Euklideszi algoritmus: maradékos osztás

$$\begin{aligned}
 a > b \quad r_1 &= a \bmod b \\
 r_2 &= b \bmod r_1 \\
 r_3 &= r_1 \bmod r_2 \\
 &\vdots \\
 r_n &= r_{n-2} \bmod r_{n-1}
 \end{aligned}$$

(a, b 8 bites regiszter nélkül)
 → Verilog % operátor nem szintetizálható
 osztó feszíthető, de nem egyszerű

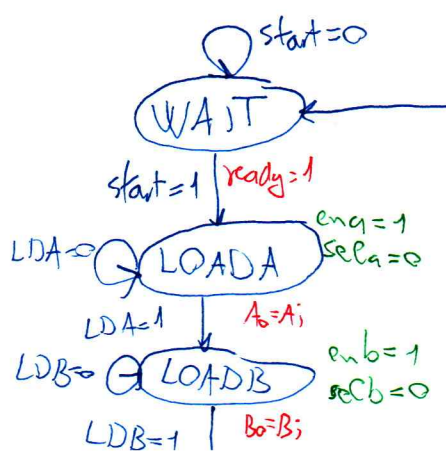
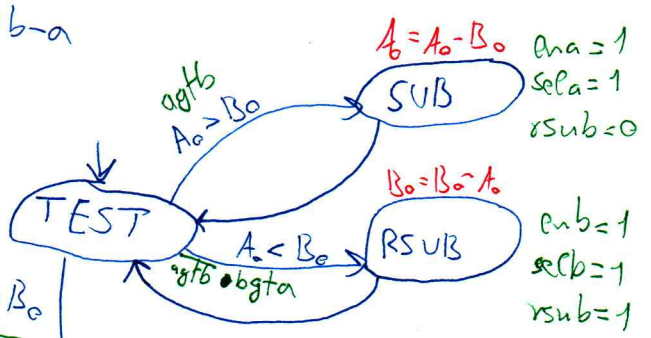
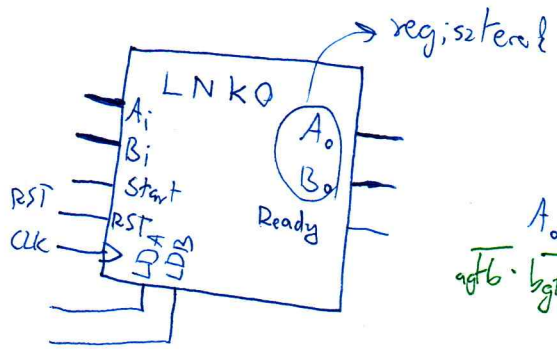
Maradékélepes kivonással:
 $a = a - b$, amíg $a > b$, utolsó kivonás: maradék

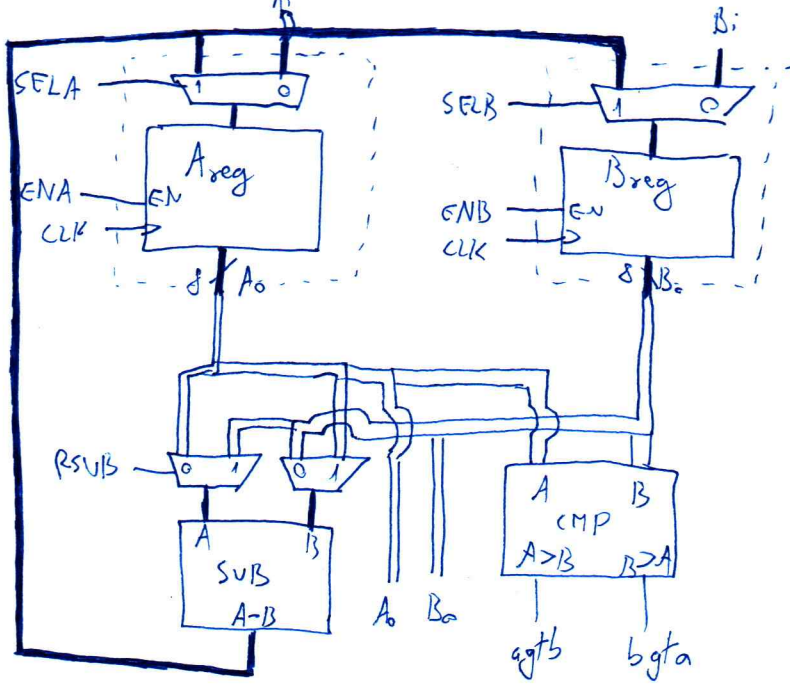
Ha $r_n = 0 \Rightarrow (a, b) = r_{n-1}$

$$\begin{aligned}
 \text{lnko}(a, b) &= \text{lnko}(a-b, b), \text{ ha } a > b \\
 \text{lnko}(a, b) &= \text{lnko}(a, a-b), \text{ ha } b > a
 \end{aligned}$$

Csere megvalósítás:

- 1, Regiszter tartalma csere
- 2, kivonás előtti operandus csere → $a = a - b$
- 3, 2 kivonás elvégzése → $b = b - a$





Digit \sum

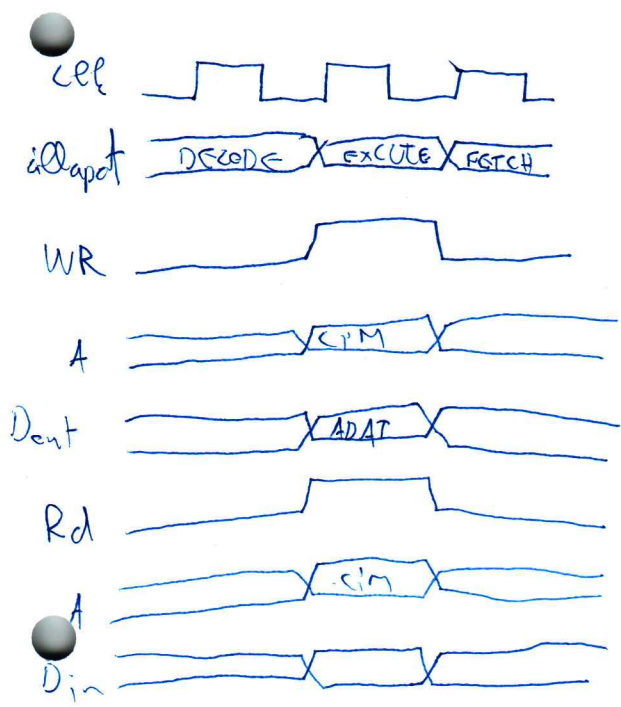
2021. 11. 09.
1

E1 $AB = 10101011$
 $10000110 = 86$

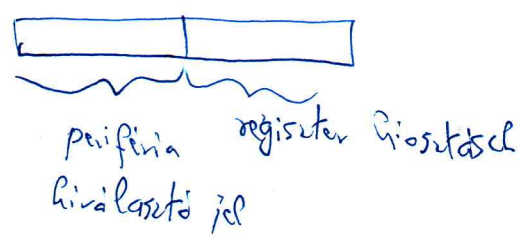


Digit 9gal

2021. 11. 23
1



2^N bites ciantanang
↳ $N=2$



Basis: 0x00

b, reg [3:0] q;

```
always @ (posedge clk)
    if (rst) q <= 4'b0000
    else if (DR_wr) q <= Dent[3:0];
```

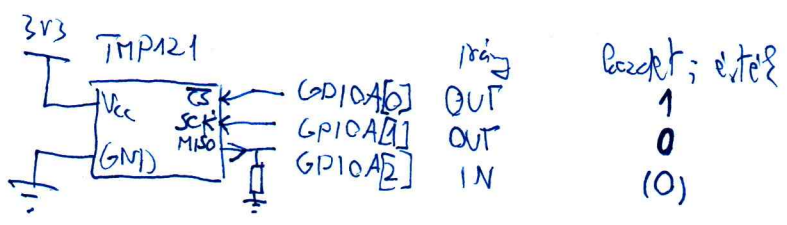
c, always(*)

```
case ({SR_rd, DR_rd})
    2'b01: Din <= {4'b0000, q};
    2'b10: Din <= {7'b0000000, BUSY};
    default: Din <= 8'h00;
```

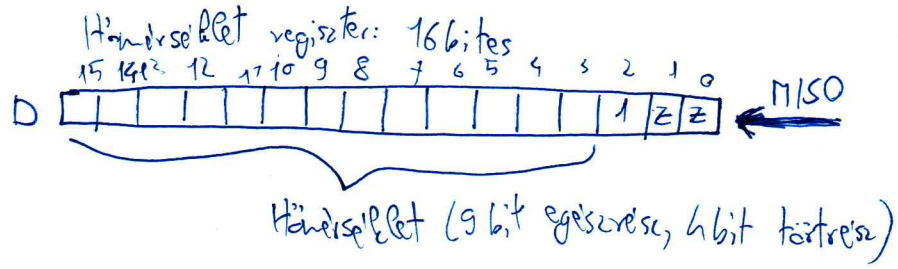
```
d, DEF PR 0x00
DEF SR 0x01
DEF DR 0x02
DEF BUSY 0x01
```

```
e, DatWR: mov r2, SR
tst r2, BUSY
jnz DatWR
mov DR, r0
mov PR, r0
swp r0
mov DR, r0
mov PR, r0
rts
```



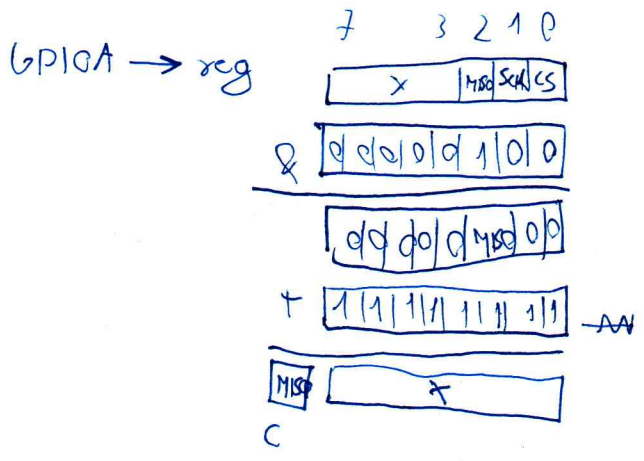
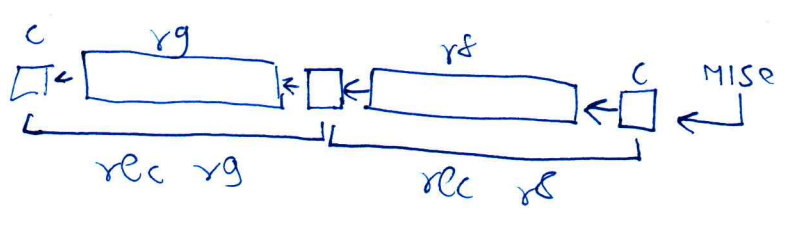
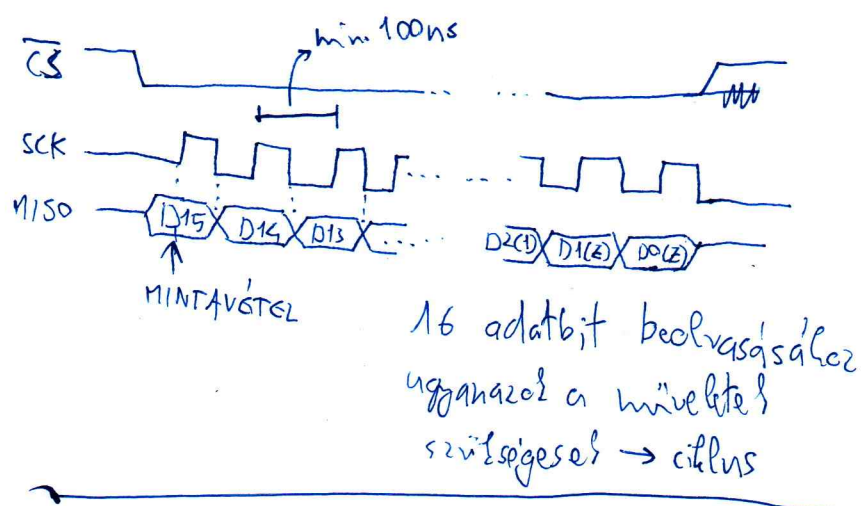


Hőmérséklet beolvasása
szubrutin
16 bit beolvasása az RS-R9-be



```

    graph TD
      A[TMP_READ] --> B["CS=0, SCK=0"]
      B --> C["cíklusváltó=16"]
      C --> D["CS=0, SCK=1"]
      D --> E["MISO beolvas"]
      E --> F["D8:D9 ← C"]
      F --> G["CS=0, SCK=0"]
      G --> H["cíklusváltó"]
      H --> I["ciklus"]
      I --> J{"cíklus == 0"}
      J -- NEM --> D
      J -- IGEN --> K["CS=1, SCK=0"]
      K --> L["Visszatérés"]
  
```





3 vezeték: CLK, TXD, RXD

- full duplex

- kezelt adatátvitel



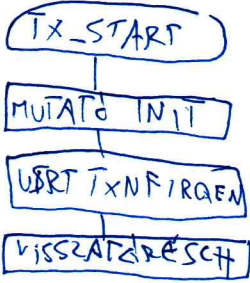
- regiszterkezelés

- vezérlő reg: adás/vétel reg
FIFO töltés

- FIFO status

- Megszállítás engedélyezés

- Adatregiszter, FIFO elérés



1) USART adás megszakításos perifériakezelés

-> régi string ellenőrzése

- "Adási FIFO nincs tele" (TXNF) megkezdés

