



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem

Logi-termikus szimuláció

Dr. Timár András

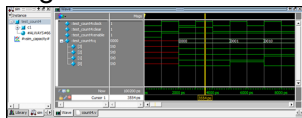
BME, Elektronikus Eszközök Tanszéke

2014. április 22.

Bevezetés

Mit jelent a logi-termikus?

Logikai szimuláció

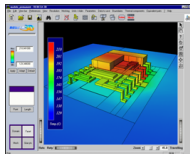


- Tisztán digitális
- Nem foglalkozik a hőmérséklettel
- Pl. QuestaSim

Termikus szimuláció



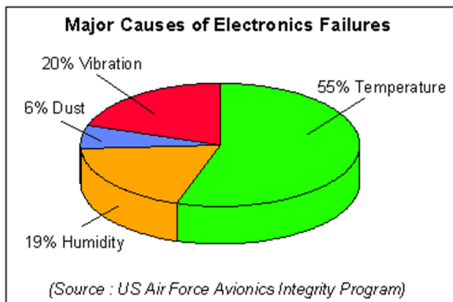
- Tisztán termikus szimuláció
- Teljesítmény, anyagjellemzők, topológia



Logi-termikus szimuláció

Miért van szükség logi-termikus szimulációra?

- A melegedés elronthatja a működést
- Az **elektro**-termikus szimulációk túl hosszúak
- Digitális cellás áramköröknél jó megoldás a *logi-termikus* szimuláció



Logikai szimulátorok

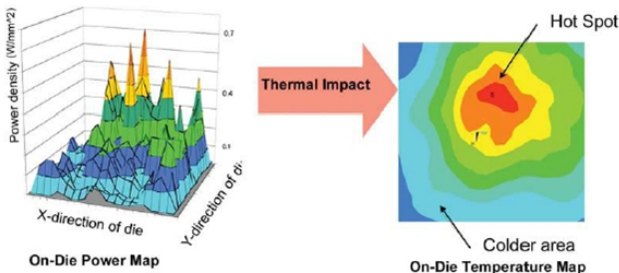
- QuestaSim, ModelSim (Mentor Graphics®)
- NCSim (Cadence®)
- VCS (Synopsys®)
- ISE Webpack (Xilinx®)
- Ingyenes szimulátorok
 - Icarus
 - ActiveHDL (csak Windows)

Termikus szimulátorok

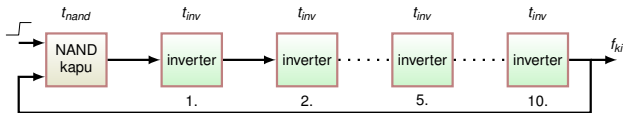
- AnSYS
- COMSOL Multiphysics
- FloTherm
- HeatWave
- HotSpot
- LayTherm, SISSI, SunRed
 - Tanszéki fejlesztés
 - Előny: tokozás és NYHL hatásainak figyelembe vétele mérési eredmények alapján

Mire használható a logi-termikus szimulátor?

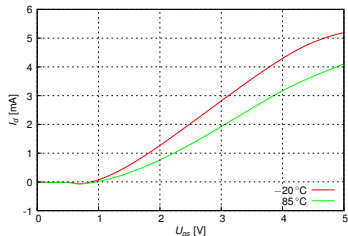
- Már a tervezés fázisban meghatározható, hogy hol lesznek a meleg pontok az áramkörön (**hot-spot**)
- Place & Route befolyásolható a hőmérsékleti térkép alapján
- Hőmérsékletfüggő időzítések szimulálhatóak, vizsgálni lehet a melegedés hatásait szimuláció közben
- **Intel P3 floating-point unit**



A saját melegedés hatása



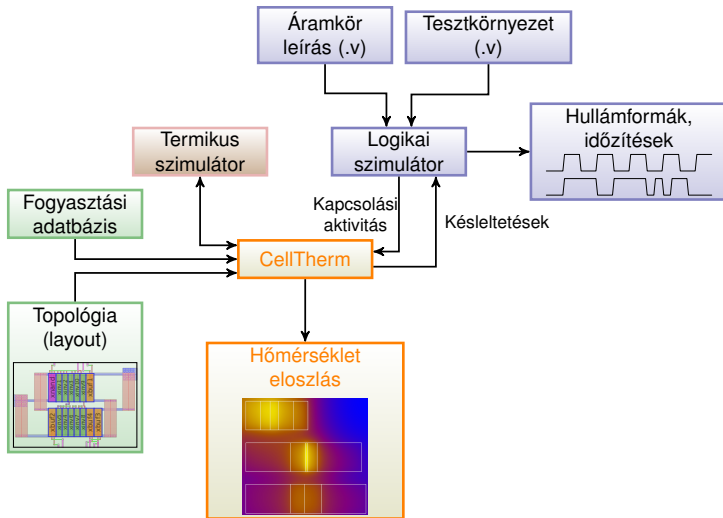
$$f_{ki} = \frac{1}{2 \cdot (t_{nand} + N \cdot t_i)}$$



Egy nMOS tranzisztor
transzfer karakterisztikája
különböző hőmérsékleten.

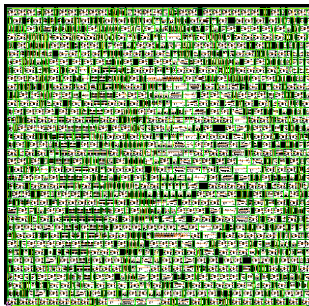
■ t_{nand} és t_i hőmérséklet-
függő

A logi-termikus szimuláció folyamata



Sztenderd cellás áramkörök

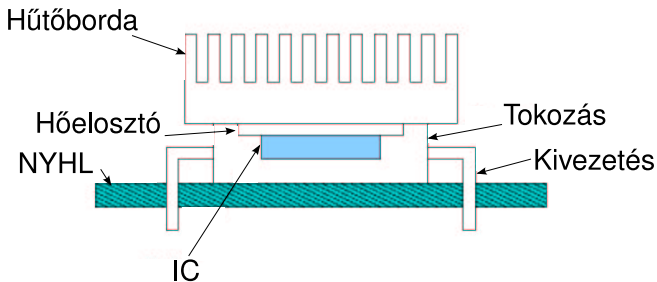
- Több ezer cella egy áramkörön
- A cellák egymáshoz a *föld* és *táp* síneken csatlakoznak
- RTL szintű építőelemek (logikai kapuk, multiplexer, tároló, stb.)
- A cella szintű felbontás elegendő → gyorsabb szimuláció



Termikus szimuláció

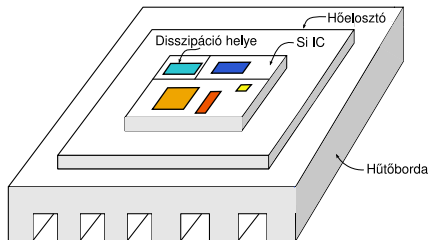
Tokozás

- Tipikus mikroelektronikai tokozás
- Kvázi-1D hőterjedés, mert oldalfalak felé nem tud menni
- Flip-chip miatt a hő a hűtőborda felé távozik (a Si fejjel lefelé van)



Disszipáló alakzatok

- Lehetnek a cellák, vagy teljes műveletvégző egységek is
- Pl. ALU, cache, stb



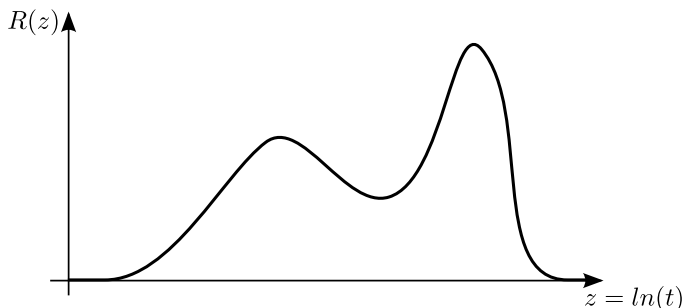
A termikus modell előállítása

- 1 Elő kell állítani a termikus válaszfüggvényeket
 - 2 Meg kell oldani a hővezetés Laplace egyenletét (1)
 - 3 Egységugrás gerjesztésre adott ugrásválasz $a(t)$
 - 4 Minden cellára kiszámoljuk a saját és a többi cella válaszát
- N cella esetén N^2 válaszfüggvény
 - Adatok: anyagparaméterek (c , λ), disszipációk helye, méretek, peremfeltételek
 - **Cél: a struktúrát modellező ekvivalens RC hálózat elkészítése**
 - SPICE analóg szimulátorral számíthatóak a hőmérsékletek

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = \frac{c}{\lambda} \frac{\partial T}{\partial t} \quad (1)$$

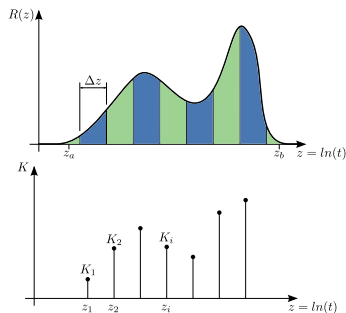
Időálló-spektrum

- Az $a(t)$ ugrás-válaszokból kiszámoljuk a termikus időálló-spektrumot $R(z)$
- Széles időtartomány (ms–óra), ezért $z = \ln(t)$ helyettesítés (logaritmikus x tengely)
- Hálózati identifikáció dekonvolúcióval (NID)

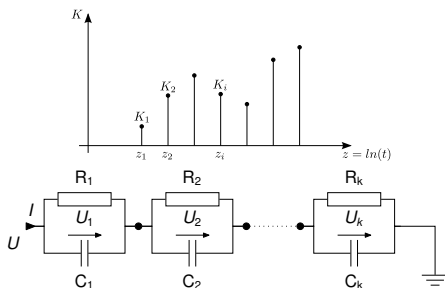


Időállandó-spektrumok diszkrétizációja

- Az ekvivalens RC modellhez véges számú értékre kell redukálni a folytonos időállandó-spektrumot



Az RC modell elemértékei



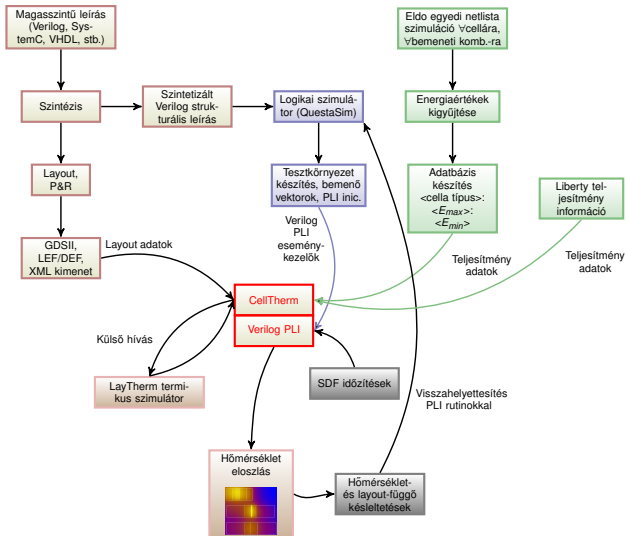
- $\tau_i = e^{z_i}$
- $R_i = K_i$
- $C_i = \frac{\tau_i}{R_i}$
- U = hőmérséklet, I = fűtőteljesítmény (disszipáció)
- A hőmérsékletek meghatározhatóak SPICE analóg szimulációval

CellTherm szimulátor

CellTherm logi-termikus szimulátor

- EET fejlesztés
- Tetszőleges szabványos logikai szimulátort használ
- Tanszéki fejlesztésű termikus szimulátor
- *Logikai és termikus* szimulátor közötti összeköttetés
- Szabványos interfészen kommunikál: **Verilog PLI**

A logi-termikus szimuláció részletes folyamata



Verilog PLI

- Minden HDL nyelv (VHDL, Verilog) támogatja a felhasználó által írt bővítményeket
- Verilog PLI: Programming Language Interface
- Saját függvények készíthetők, amik meghívhatóak a HDL kódból (pl. `$display()`, `$finish()`, `stb.`)
- Ezek C/C++ modulok, dinamikusan betölthetők (.dll/.so)
- A logikai szimulátor ezeket a felhasználói függvényeket meghívhatja
- Ki lehet egészíteni a logikai szimulátort akár egy termikus szimulátorral is

A PLI használata

```

module testpli;

  initial
    $shellovilag();
    //{usertask, 0, 0, 0,    hello,
      0, "$shellovilag"}

endmodule

```

1. kódrészlet. Verilog kód

■ Fordítás

```

gcc -fPIC -shared
-I/questasim/include
-o pli.so pli.c

```

```

#include <veriusert.h>
#include <acc_user.h>

PLI_INT32 hello(PLI_INT32 data,
  PLI_INT32 reason)
5 {
    io_printf("Hello Vilag!");
    return 0;
}

10 s_tfcell veriusertfs[] = {
    {usertask, 0, 0, 0, hello, 0,
      "$shellovilag"},
    {0} /* last entry must be 0 */
};

15 void init_usertfs()
{
    p_tfcell usertf = veriusertfs;
    while (usertf->type)
        mti_RegisterUserTF(usertf++);
20 }

```

2. kódrészlet. PLI C kód

QuestaSim indítása saját PLI rutinnal

- Munkakönyvtár létrehozása

```
$> vlib munka
```

- Munkakönyvtár hozzárendelés

```
$> vmap work munka
```

- Verilog forrás fordítása

```
$> vlog pli.v
```

- QuestaSim indítása

```
$> vsim -pli pli.so testpli
```

A szimulátor kimenete

```
Reading
 /soft/mentor/ams/questasim/v10.1/tcl/vsim/pref.tcl
# 10.1
# vsim -c -pli pli.so testpli
# Loading ./pli.so
# // Questa Sim-64
# // Version 10.1 linux_x86_64 Dec  5 2011
...
# //
# Loading work.testpli
#
VSIM 1> run
# Hello Vilag!
```

Logi-termikus tesztkörnyezet

```
`include "ringosc.v"
`define TH_Timestep 1e6 // = 1ms
`timescale 1ns / 1ps
module testbench;
    reg clk;
    reg run;
    wire ring_out;

    top dut (clk, run, ring_out);

    initial
    begin

        clk = 0;
        run = 1;
    end

    always #1000 clk = ~clk;

endmodule
```

Logi-termikus tesztkörnyezet

```
`include "ringosc.v"
`define TH_TIMESTEP 1e6 // = 1ms
`timescale 1ns / 1ps
module testbench;
    reg clk;
    reg run;
    wire ring_out;

    top dut (clk, run, ring_out);

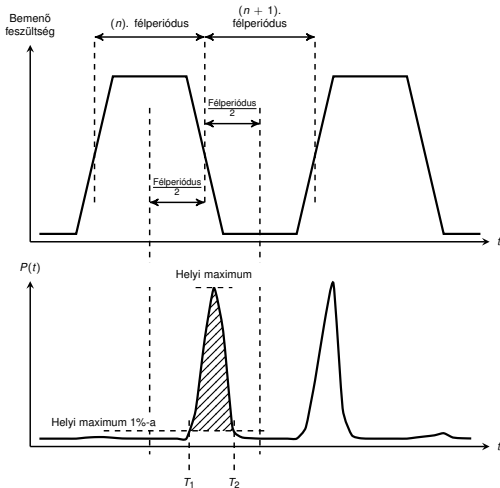
    initial
    begin
        $celltherm_initlayout(dut,10,10,"ringosc.lib","ringosc.lef","ringosc.def");
        $celltherm_init(dut,1,`TH_TIMESTEP);

        clk = 0;
        run = 1;
    end

    always #1000 clk = ~clk;

    // Termikus motor hívása, pl. dt = 1 ms
    // #1e6 = 1ms
    always
    begin
        #`TH_TIMESTEP $celltherm_calc(`TH_TIMESTEP, 25);
    end
endmodule
```

Teljesítmény karakterizáció



- Minden cellára
- Bemeneti meredekséggel és terhelő kapacitással paraméterezve
- Analóg SPICE szimulátorral
- $P(t)$ alatti terület **energia**
- CMOS-nál jó

A teljesítmény-sűrűség számítása

- Logikai szimulátorból $\rightarrow$ kapcsolási aktivitás
- Δt idő alatti jelváltások száma (n)
- Kapcsolási (dinamikus) energia $E_{\text{kapcs}} = \int_{T1}^{T2} P(t) dt$
- $P_{\Delta t} = \frac{E_{\text{kapcs}}}{\Delta t}$
- Δt alatt fogyasztott teljesítmény minden cellára
- $P_{\text{átlag}} = n \cdot P_{\Delta t}$
- $P_{\text{átlag}}$ -gal hívjuk meg a termikus szimulátort
- A termikus szimulátor minden Δt -re ad egy hőmérsékletet
- Δt milliszekundumos nagyságrendű

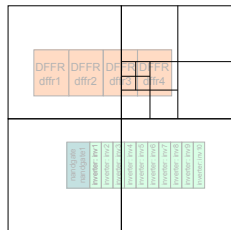
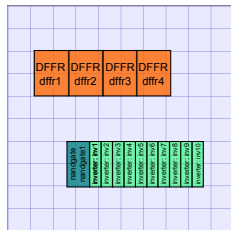
Liberty adatbázis

- A cellakönyvtárral jön (általában)
- Ha nincs ilyenünk, akkor kézzel kell karakterizálni (lásd teljesítmény karakterizáció)
- Időzítések és fogyasztások minden cellára
- Bemeneti jelmeredekséggel és terhelő kapacitással paraméterezve
- Szöveges fájl, könnyen olvasható
- Mi is tudunk ilyet csinálni SPICE szimulációk alapján

```
pin(CO) {  
    direction : output;  
    max_capacitance : 0.32;  
    internal_power() {  
        rise_power(power_outputs_0) {  
            values( " 0.078, 0.054, -0.029, -0.358, -0.798", \  
                  " 0.085, 0.061, -0.023, -0.352, -0.796", \  
                  " 0.096, 0.072, -0.013, -0.344, -0.798", \  
                  " 0.123, 0.097, 0.009, -0.323, -0.837");  
        }  
    }  
}
```

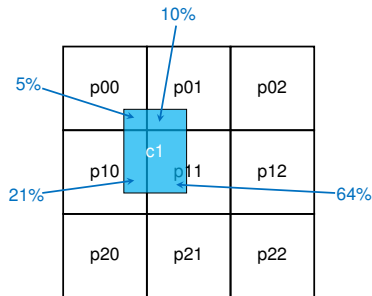
Particionálás

- N cella, modell generálási idő $\sim N^2$
- N^2 darab modell
- Cellák helyett hálószerű felbontás
 - Termikus modell számítása gyorsabb
 - Több ezer cellánál is működik, RC hálózat generálása konstans idő
- Kisebb felbontás, a felbontás növeléséhez a partíciók tovább oszthatók



Teljesítmény számítása a partíciókban

- Egy partícióban fogyasztott teljesítmény a benne található cellák teljesítményével arányos



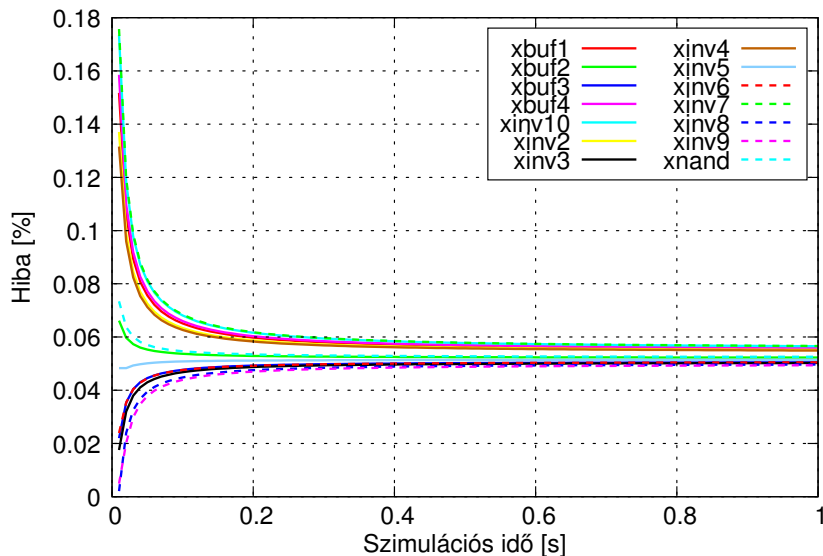
$$P(p_{00}) = 0,05 \cdot P(c_1)$$

$$P(p_{01}) = 0,10 \cdot P(c_1)$$

$$P(p_{10}) = 0,21 \cdot P(c_1)$$

$$P(p_{11}) = 0,64 \cdot P(c_1)$$

A particionálás által bevitt hiba



A szimuláció gyorsítása

- Gigahertzes működésnél a tiszta logikai szimuláció is lassú
- A nagy termikus időállandók miatt sokáig kell szimulálni (1-10 sec)
- Egy skálafaktorral (pl. $f = 10\times$ vagy $100\times$) növeljük meg minden késleltetést és kapcsolási energiát
- Késleltetés $\uparrow$, frekvencia (ν) $\downarrow$, kapcsolási energia $\uparrow$

$$E = C \cdot U^2$$

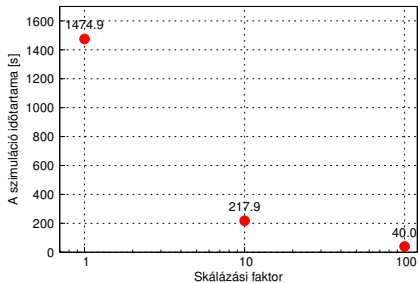
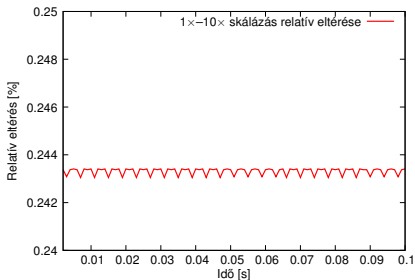
$$P = C \cdot U^2 \cdot \nu$$

$$P' = E' \cdot \nu' = 10E \cdot \frac{1}{10}\nu = E \cdot \nu$$

$$P' = P$$

A skálázással bevitt hiba

■ Nem jelentős



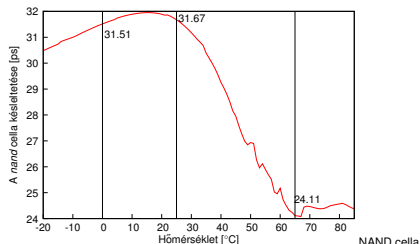
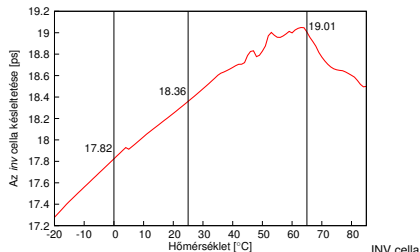
Hőmérsékletfüggő késleltetések

Időzítési problémák

- Logi-termikus szimulációval meghatározhatóak a hot-spot-ok
- Az időzítések megváltoznak a melegedés miatt
- Szélsőséges hőmérsékletnél a megfelelő logikai működés megghi-
úsulhat. (pl. **setup és hold time** nem teljesül)

Késleltetés-hőmérséklet függések

■ Minden cellára meg lehet határozni



SDF – Standard Delay Format

- A szintézer elkészíti a **pre-** és **post-layout** időzítési adatbázist
- SDF – Standard Delay Format 5
- Szöveges fájl, késleltetések minden cellára, setup/hold idők definíciója, stb. 10
- Pontos időzítési értékek az elhelyezéstől és vezetékeztől 15
- Csak 3 sarokpontban tárolja el az időzítéseket, ezzel baj lesz később! 20

```
(DELAYFILE
(SDFVERSION "2.0")
(DESIGN "ringosc")
(TIMESCALE 1 ns)
...
(CELL
(CELLTYPE "dffr")
(INSTANCE dffr1)
(DELAY
  (ABSOLUTE
    (PORT D (::0.00) (::0.00))
    (PORT CLK (::0.00) (::0.00))
    (PORT R (::0.00) (::0.00))
    (IOPATH CLK Q (::0.49) (::0.53))
    (IOPATH R Q (::0.00) (::0.58))
    (IOPATH CLK QB (::0.39) (::0.30))
    (IOPATH R QB (::0.44) (::0.00))))
(TIMINGCHECK
  (SETUP D (posedge CLK) (0.47))
  (HOLD D (posedge CLK) (0.06))))
```

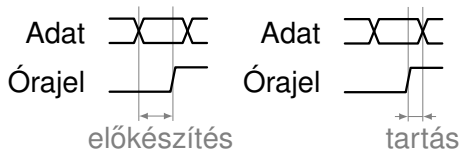
Az SDF használata logikai szimulációban

- A logikai szimulátor támogatja az SDF formátumot
- A cellák késleltetéseit az SDF-ből helyettesíti be
- Valós késleltetések

```
C:/prog/LOGITHRM/Modelsim/sources/delay2.v
Ln#
1  `timescale 1ps / 100fs
2  module delay2;
3      reg clk = 0;
4      reg z;
5      reg di;
6      wire q;
7      wire qb;
8
9      buf #(40) b1(do,di);
10     dffr dut1(.D(do), .CLK(clk), .R(z), .Q(q), .QB(qb) );
11
12     initial
13     begin
14         $sdf_annotate("sources/delay2_c.sdf",delay2,,"MAXIMUM");
15         z = 0;
16         clk = 0;
17
18         di = 1;
19         #540 di = 0;
20         #1000 di = 1;
21     end
22
23     always
24     begin
25         #1000 clk = ~clk;
26     end
27
28 endmodule
```

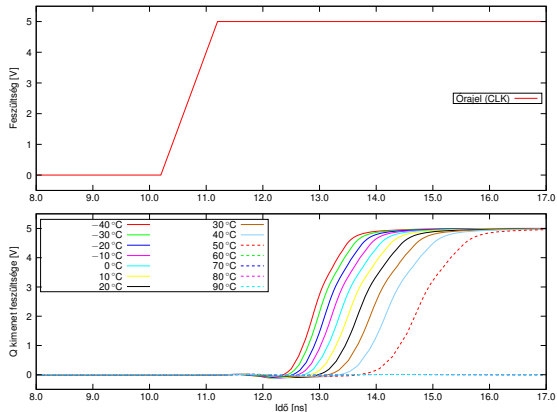
Setup- és hold időzítés

- A logikai szimulátor jelzi, ha a **setup** vagy **hold** feltételek nem teljesülnek.
- Ez be van építve minden logikai szimulátorba
- Pontos időzíti vizsgálatok végezhetők



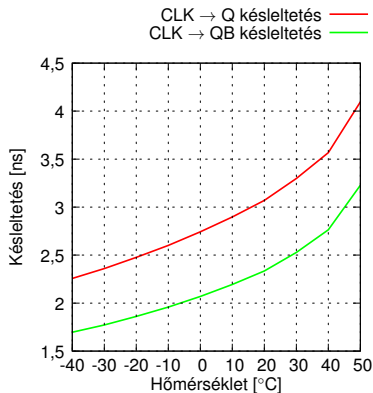
D flip-flop cella időzíti

- Egy D flip-flop cella késleltetése több hőmérsékleten
- Bizonyos hőmérséklet felett a **setup-time** nem teljesül
- A bemeneti változás nem jut ki a kimenetre



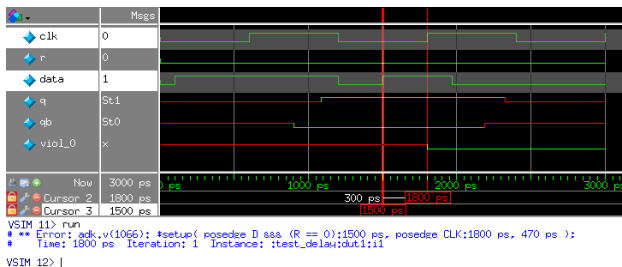
D flip-flop cella késleltetése

- Cella késleltetés a hőmérséklet függvényében
- Az analóg szimulációt -40 – $+90$ °C között futtattuk
- Csak -40 – $+50$ °C tartományban értelmezhető
- 50 °C felett nem jut ki a bemenet a kimenetre → a késleltetés nem értelmezett



■ SDF értékek a DFFR cellára (D flip-flop tároló)

- Setup time: 470 ps
- Hold time: 60 ps



■ $1800 \text{ ps} - 1500 \text{ ps} = 300 \text{ ps}$

■ $300 \text{ ps} < 470 \text{ ps}$

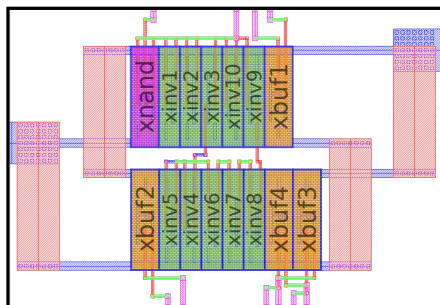
Konzekvenciák

- A hőmérsékleti eloszlást le kell képezni késleltetés értékekre
- Szükséges a hőmérséklet-késleltetés görbék kinyerése
- A megváltozott késleltetések a **szimuláció közben** visszahelyettesíthetők, a szimulátor a megváltozott értékekkel számol tovább
- Ha ebből adódóan időzítési hiba lép fel, a szimulátor ezt jelzi!

Demonstrációs áramkör

Teszt áramkör – gyűrűs oszcillátor

- Felület: 0,518 mm × 0,355 mm
- 4 puffer cella, 10 inverter, 1 NAND kapu
- Gyűrűs oszcillátor: 1 NAND + 10 inverter
- A ring oszcillátor frekvenciája eltolódik a változó hőmérséklet miatt



Ring oszcillátor Verilog leírása

```

`timescale 1ns / 1ps
module top(run, data, osc, buf_out);
    input    run;
    input    [3:0] data;
    output    osc;
    output    [3:0] buf_out;

    wire w1, w2, w3, w4, w5, w6, w7, w8, w9, w10;

    buf02 xbuf1 ( .Y(buf_out[0]), .A(data[0]) );
    buf02 xbuf2 ( .Y(buf_out[1]), .A(data[1]) );
    buf02 xbuf3 ( .Y(buf_out[2]), .A(data[2]) );
    buf02 xbuf4 ( .Y(buf_out[3]), .A(data[3]) );

    nand02 xnand ( .Y(w1), .A0(run), .A1(osc) );

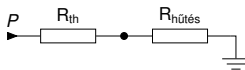
    inv01 xinv1 ( .Y(w2), .A(w1) );
    inv01 xinv2 ( .Y(w3), .A(w2) );
    inv01 xinv3 ( .Y(w4), .A(w3) );
    inv01 xinv4 ( .Y(w5), .A(w4) );
    inv01 xinv5 ( .Y(w6), .A(w5) );
    inv01 xinv6 ( .Y(w7), .A(w6) );
    inv01 xinv7 ( .Y(w8), .A(w7) );
    inv01 xinv8 ( .Y(w9), .A(w8) );
    inv01 xinv9 ( .Y(w10), .A(w9) );
    inv01 xinv10 ( .Y(osc), .A(w10) );

endmodule

```

Kézi számítás

	Jelölés	Érték
Felület	A	$1,841 \cdot 10^{-7} \text{ m}^2$
Si réteg vastagsága	d	$5 \cdot 10^{-5} \text{ m}$
Si termikus vezetőképessége	λ	$156,3 \frac{\text{W}}{\text{mK}}$
Fűtőteljesítmény	P	$4,022 \text{ mW}$
Alsó hűtés	HTC	$500 \frac{\text{W}}{\text{m}^2 \text{ K}}$
Környezeti hőmérséklet	T_{amb}	$0 \text{ }^\circ\text{C}$



$$R_{th} = \frac{1}{\lambda} \cdot \frac{d}{A} = \frac{1}{156,3} \cdot \frac{5 \cdot 10^{-5}}{1,841 \cdot 10^{-7}} \left[\frac{\text{mK}}{\text{W}} \cdot \frac{\text{m}}{\text{m}^2} \right] = 1,737 \frac{\text{K}}{\text{W}}$$

$$G_{hűtés} = 500 \frac{\text{W}}{\text{m}^2 \text{ K}}$$

$$R_{hűtés} = \frac{1}{500} \cdot \frac{1}{1,841 \cdot 10^{-7}} \left[\frac{\text{m}^2 \text{ K}}{\text{W}} \cdot \frac{1}{\text{m}^2} \right] = 10,863 \cdot 10^3 \frac{\text{K}}{\text{W}}$$

$$T = P \cdot (R_{th} + R_{hűtés}) = 4,022 \cdot 10^{-3} \cdot (10,863 \cdot 10^3 + 1,737) = 43,7 \text{ }^\circ\text{C}$$

Milyen adatok kellenek egy logi-termikus szimulációhoz?

- Layout leírás (LEF/DEF)
- Teljesítmények adatbázisa (Liberty)
- A sztenderd cellák Verilog viselkedési leírása
- SDF késleltetéseket tartalmazó fájl
- Verilog testbench

Layout leírás (LEF/DEF)

- Sztenderd cellás áramköröknél szabványos leírási mód
- LEF/DEF formátum, szöveges fájlok
- Minden szintézer szoftver által ismert ki- és bemeneti formátum
- LEF
 - Minden cella prototípusát tartalmazza
 - Méret, belső összeköttetések
 - Rétegek leírása, viák helyzete
 - Belső elektromos összeköttetések
- DEF
 - A cella prototípusok példányosítása
 - Pl. inverter cella: inv1, inv2, ..., inv10
 - Megadja a cellák pontos helyét a layout-on
 - Összeköttetések, vezetékek a cellák között

Teljesítmények adatbázisa (Liberty)

- Szabványos, nyílt formátum a cellakönyvtár jellemzőinek tárolására
- Szöveges formátum
- Időzítések
- Késleltetések
- Cella kapacitások
- Teljesítmények
- Egy jelváltás egy bizonyos cella egy portján **hány Joule** energia fogyasztással jár (ezt fogjuk felhasználni!)
- Minden adat táblázatos formában van jelen, pl:
 - A fogyasztott teljesítmény függ a **terhelő kapacitástól**
 - ... és a gerjesztő jelek **felfutási/lefutási idejétől**
 - Ez egy 3 dimenziós táblázat

Verilog viselkedési leírás

- A cellakönyvtár tartalmazza minden cella viselkedési leírását
- Így lehet logikai szimulációt végezni

```
`celldefine
`timescale 1ns / 1ps
module nandgate(OUT, A, B);
    input A;
    input B;
    output OUT;

    specify
        specparam delayA = 1;
        specparam delayB = 1;

        (A ==> OUT) = (delayA);
        (B ==> OUT) = (delayB);
    endspecify

    assign OUT = ~(A==1'bx ? 0 : A & B==1'bx ? 0 : B);
endmodule
`endcelldefine
```

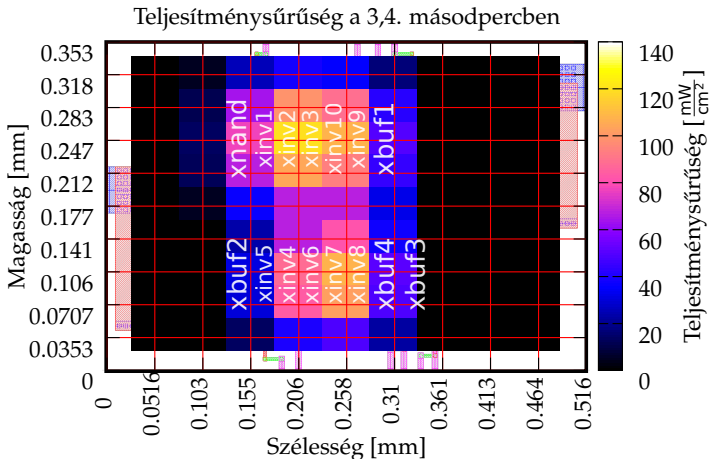
SDF késleltetések

- A szintézer program generálja
- Tartalmazza minden cella késleltetését és a **setup/hold** feltételeket

```
(DELAYFILE
(SDFVERSION "2.0")
(DESIGN "ringosc")
(TIMESCALE 1 ns)
...
(CELL
(CELLTYPE "dffr")
(INSTANCE dffr1)
(DELAY
(ABSOLUTE
(PORT D (::0.00) (::0.00))
(PORT CLK (::0.00) (::0.00))
(PORT R (::0.00) (::0.00))
(IOPATH CLK Q (::0.49) (::0.53))
(IOPATH R Q (::0.00) (::0.58))
(IOPATH CLK QB (::0.39) (::0.30))
(IOPATH R QB (::0.44) (::0.00))))
(TIMINGCHECK
(SETUP D (posedge CLK) (0.47))
(HOLD D (posedge CLK) (0.06))))
```

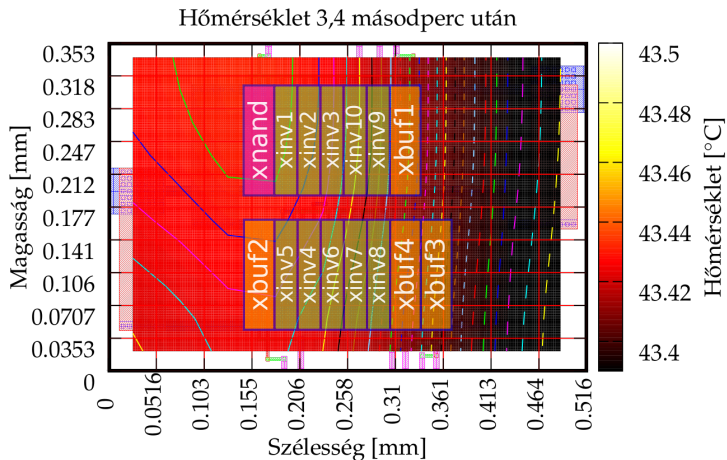
Teljesítmény térkép

- A puffer cellák fogyasztása 1 mW/billenés
- Az inverterek fogyasztása 1 μ W/billenés



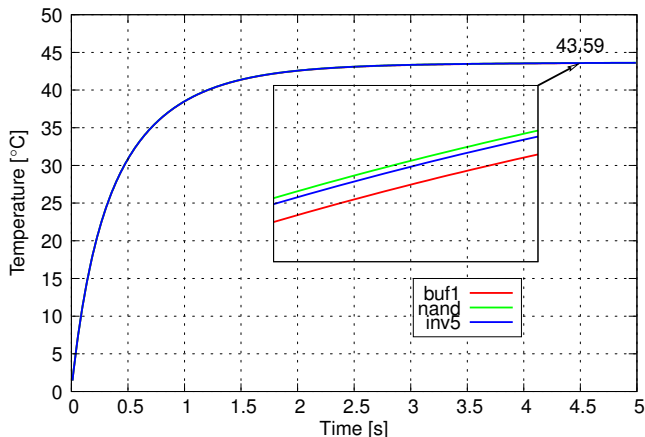
Hőmérsékleti térkép

- A fogyasztási eloszlás alapján kialakult hőmérsékleti térkép



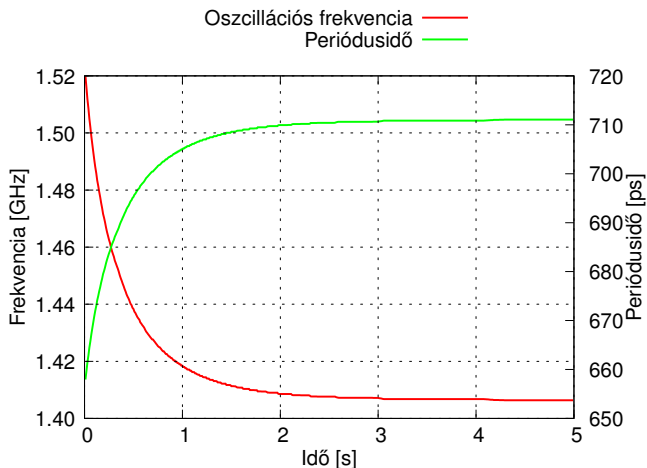
Hőmérsékleti diagram

- A DFFR cellák és az inv5 cella hőmérsékletei
- $f(t) = A \left(1 - e^{-\frac{t}{\tau}}\right)$
- A hőmérséklet megegyezik a kézzel számolttal (43,7 °C)



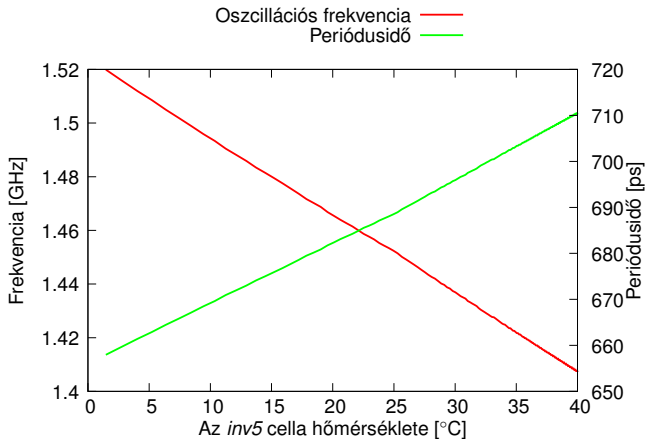
A ring oszcillátor frekvenciája

■ Frekvencia és periódusidő időfüggése



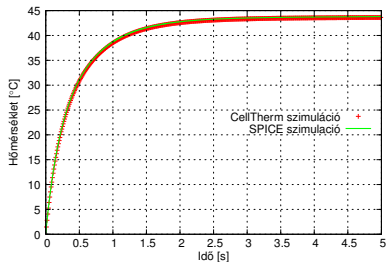
A ring oszcillátor frekvenciája

- Frekvencia és periódusidő az `inv5` cella hőmérsékletének függvényében

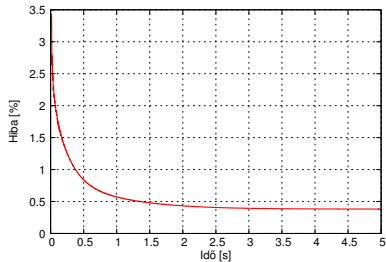


Verifikáció

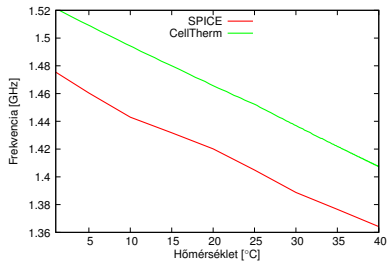
SPICE–logi-termikus verifikáció



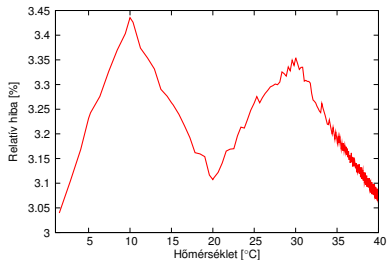
■ Az eltérés 3,5% alatti



A frekvenciamenet SPICE verifikációja

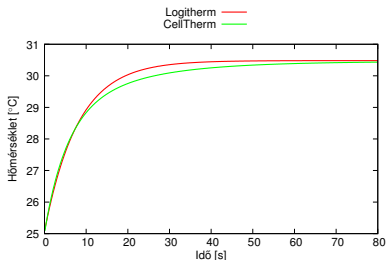


■ Az eltérés 3,5% alatti

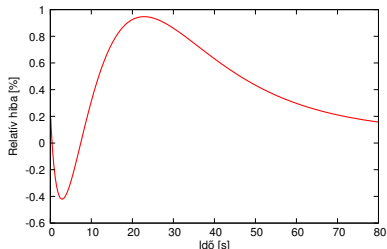


CellTherm és LogiTherm összehasonlítás

- A LogiTherm szintén az EET tanszéken fejlesztett más elvű logi-termikus szimulátor



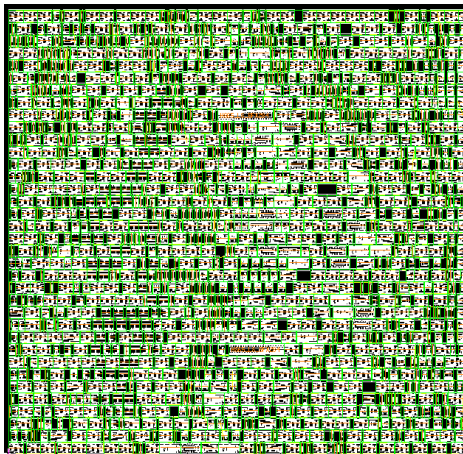
■ Az eltérés 1 % alatti



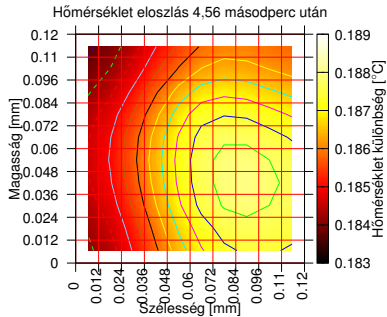
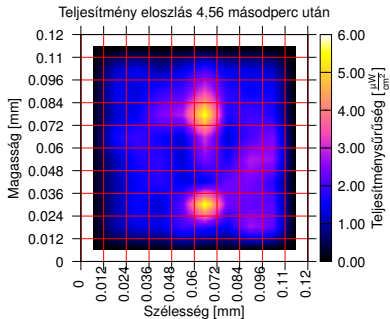
Ipari áramkör

Ipari méretű és bonyolultságú áramkör

- POLITO torinói műszaki egyetem fejlesztése
- Jelfeldolgozó áramkör, 1490 cella



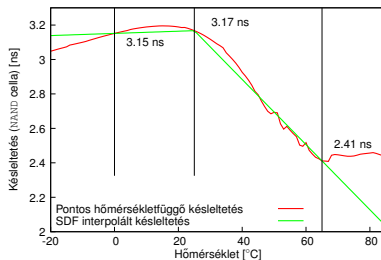
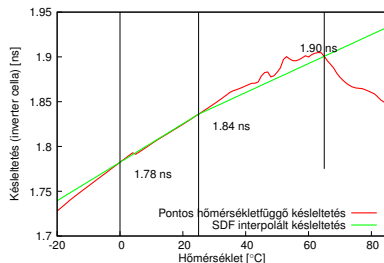
POLITO áramkör szimulációja



Új hőmérsékletfüggő időzítési modell

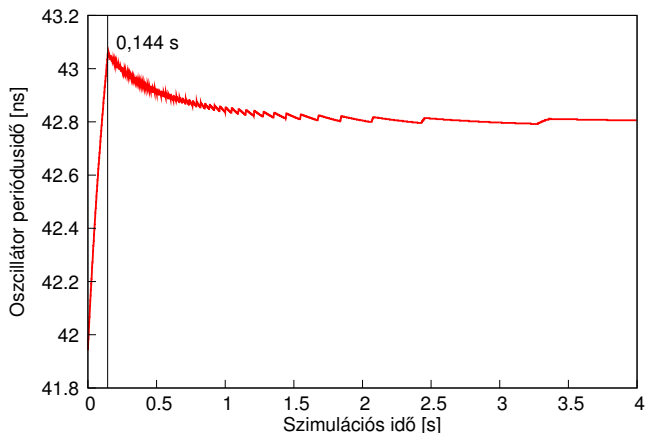
Probléma az SDF modellel

- Csak a sarokpontoknál (design corners) pontos
- A sarokpontok között lineáris interpoláció kell
- Nem a pontos késleltetés-hőmérséklet görbével számolunk!



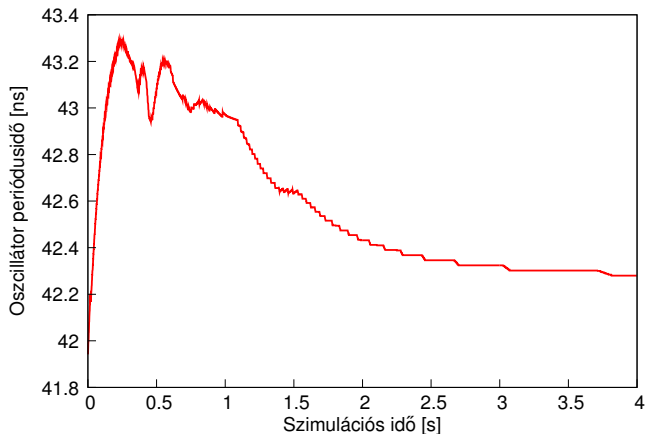
Szimuláció SDF késleltetések alapján

- Törés 0,144 s-nál
- Itt lesz 25 °C fokos a hőmérséklet
- **Töréspont** a linearizált SDF karakterisztikában!
- Pontatlan!

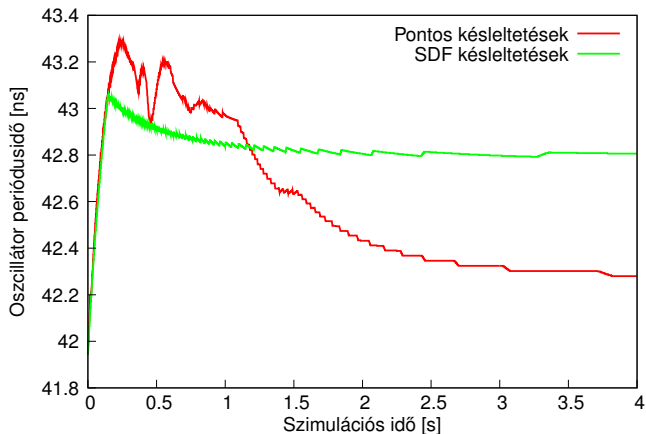


Szimuláció a tényleges hőfüggő késleltetésekkel

- Sokkal pontosabb eredmények!



A pontos és a linearizált SDF eredmények eltérése



Összegzés

- A CellTherm a BME EET tanszéken fejlesztett logi-termikus szimuláció futtatására alkalmas eszköz
- Digitális standard cellás áramkörök szimulációjára alkalmas
- Ipari logikai és termikus szimulátorokat kapcsol össze szabványos interfészekkel
- Az IC tervezőknek nem kell új eszközöket megismerniük
- A hőmérsékletfüggő késleltetések figyelembe vételével képes jelezni, ha a megváltozott hőmérsékletek miatt az áramkör funkcionálisan működésképtelenné válik