



Budapesti Műszaki és Gazdaságtudományi Egyetem  
Villamosmérnöki és Informatikai Kar

# Új módszer integrált áramkörök funkcionális és termikus viselkedésének együttes szimulációjára

## Doktori értekezés

Szerző: Jani Lázár  
okleveles villamosmérnök

Témavezető: Dr. Poppe András  
MTA doktora  
egyetemi docens

Elektronikus Eszközök Tanszéke  
Budapest, 2019.

# Nyilatkozat önálló munkáról, hivatkozások átvételéről

Alulírott Jani Lázár kijelentem, hogy ezt a doktori értekezést magam készítettem és abban csak a megadott forrásokat használtam fel. Minden olyan részt, amelyet szó szerint, vagy azonos tartalomban, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Budapest, 2019. június 9.

# Nyilatkozat nyilvánosságra hozatalról

Alulírott Jani Lázár hozzájárulok a doktori értekezésem Interneten történő nyilvánosságra hozatalához az alábbi formában:

- korlátozás nélkül
- elérhetőség csak magyarországi címről
- elérhetőség a fokozat odaítélését követően 2 év múlva, korlátozás nélkül
- elérhetőség a fokozat odaítélését követően 2 év múlva, csak magyarországi címről

Budapest, 2019. június 9.

# Tartalomjegyzék

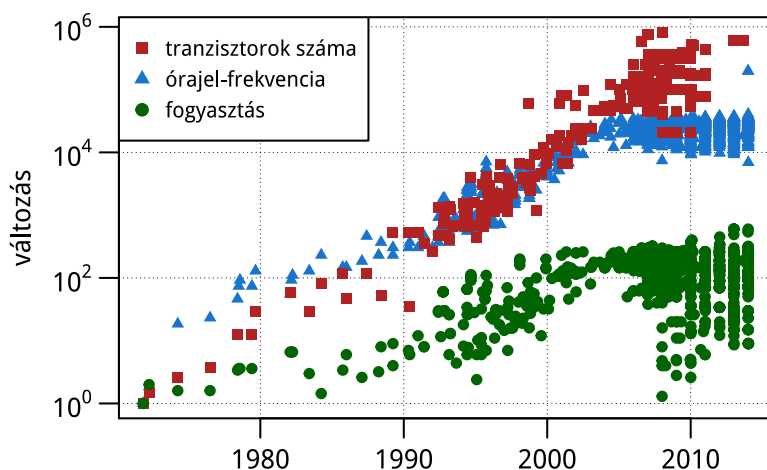
<b>1. Bevezetés</b>	<b>3</b>
<b>2. Irodalmi áttekintés</b>	<b>6</b>
2.1. IC-k modellezése a tervezés során . . . . .	6
2.1.1. Az integrált áramkörök tervezési folyamata . . . . .	6
2.1.2. Disszipáció modellezése . . . . .	11
2.2. Termikus szimulációs módszerek . . . . .	14
2.2.1. Numerikus módszerek . . . . .	15
2.2.2. Termikus kompakt modellezés . . . . .	16
2.3. Termikus mérési módszerek . . . . .	19
2.4. Funkcionális-termikus együttes szimuláció . . . . .	20
2.4.1. Az együttes szimuláció módszerei . . . . .	20
2.4.2. Együttes szimulációt megvalósító rendszerek . . . . .	22
2.4.3. Az eddigi megoldások korlátai . . . . .	23
<b>3. Példamodellek és felhasznált eszközök</b>	<b>25</b>
3.1. A dolgozat demonstrációs modelljei . . . . .	25
3.2. A felhasznált tervező szoftver eszközök . . . . .	26
<b>4. A szimulációs tartományok egyesítése</b>	<b>28</b>
4.1. Termikus viselkedés modellezése . . . . .	28
4.2. Absztrakt interfész kidolgozása . . . . .	29
I. Tézis . . . . .	31
4.3. Összehasonlítás a korábbi megoldásokkal . . . . .	31
<b>5. A tervezési folyamat bővítése</b>	<b>32</b>
5.1. A tervezés során használt formális nyelvek . . . . .	32
5.2. Szükséges többlet információ előállítás . . . . .	34
5.2.1. Layout reprezentáció . . . . .	35
5.2.2. Disszipációmodellek előállítása . . . . .	35
II. Tézis . . . . .	39
5.3. A példa processzorok disszipáció és layout modelljei . . . . .	39
5.3.1. A megvalósított fizikai elrendezések . . . . .	39
5.3.2. A felhasznált disszipáció modellek . . . . .	41
<b>6. Új termikus motor megvalósítása</b>	<b>45</b>
6.1. Tokozás szintű termikus modellezés FVM-mel . . . . .	45
6.1.1. Szimulációs tartomány diszkrétizálása . . . . .	45

6.1.2.	Peremfeltételek beépítése a differenciálegyenletbe . . . . .	48
6.2.	A megoldandó differenciálegyenlet-rendszer . . . . .	50
6.3.	Az új termikus szimulációs motor vizsgálata . . . . .	52
6.3.1.	A megoldó algoritmus kiválasztása és paraméterezése . . . . .	52
6.3.2.	Numerikus módszer kiválasztása . . . . .	53
6.3.3.	A termikus motor validációja . . . . .	56
6.3.4.	A futási idő vizsgálata . . . . .	57
6.4.	A szimuláció időlépésének megválasztása . . . . .	59
6.4.1.	Konstans időlépés . . . . .	59
6.4.2.	Adaptív időlépés . . . . .	60
III.	Tézis . . . . .	63
6.5.	Az új motor tulajdonságainak összefoglalása . . . . .	63
<b>7.</b>	<b>Funkcionális-termikus verifikációs módszer</b>	<b>64</b>
7.1.	Termikus modellel bővített verifikáció . . . . .	64
7.2.	A keretrendszer felépítése és működése . . . . .	65
7.2.1.	Logikai/funkcionális szimulátorok adaptere . . . . .	66
7.2.2.	Termikus szimulátorok adaptere . . . . .	67
7.3.	Szimulációs eredmények . . . . .	68
7.3.1.	Hőmérsékletfüggő késleltetés modellezése . . . . .	68
7.3.2.	Kevert jelű rendszer hőmérsékletfüggő viselkedése . . . . .	71
7.3.3.	Dinamikus hőmérséklet-menedzsment vizsgálata . . . . .	74
IV.	Tézis . . . . .	76
7.4.	A bővített verifikációs módszer összefoglalása . . . . .	76
<b>8.</b>	<b>Összefoglalás</b>	<b>78</b>
	<b>Köszönetnyilvánítás</b>	<b>80</b>
<b>A</b>	<b>A konduktív hővezetés differenciálegyenlete</b>	<b>81</b>
<b>B</b>	<b>Numerikus módszerek ODE-k megoldásához</b>	<b>83</b>
<b>C</b>	<b>További szimulációs eredmények</b>	<b>87</b>
C.1.	$\alpha$ processzor . . . . .	87
C.2.	$\beta$ processzor . . . . .	90
<b>D</b>	<b>Rövidítések jegyzéke</b>	<b>92</b>
	<b>Saját közlemények</b>	<b>94</b>
	<b>Tézishez szorosan nem kapcsolódó közlemények</b>	<b>95</b>
	<b>Irodalomjegyzék</b>	<b>96</b>
	<b>Ábrák jegyzéke</b>	<b>105</b>
	<b>Táblázatok jegyzéke</b>	<b>107</b>

# 1. fejezet

## Bevezetés

Az elmúlt fél évszázad félvezető technológiai fejlődésével az integrált áramkörök gyártása egyre gazdaságosabbá vált amely lehetővé tette az összetettségnek – az egységnyi területre integrált komponensek számának – folyamatos növekedését [1] (1.1. ábra).



1.1. ábra. A processzorok egyes paramétereinek változása 1970 és 2014 között [2]

A modernebb gyártósorok kisebb méretű tranzisztorok előállítását tették lehetővé, ami kedvező hatással volt a fogyasztásra és a kapcsolási időre. A Dennard-féle skálázási szabály szerint a CMOS<sup>2</sup> kapcsolástechnikában alkalmazott MOSFET-ek<sup>3</sup> fizikai méreteinek és feszültségeinek csökkentése valamint az adalékolás növelése egyazon  $\alpha$  arányossági tényezővel a fogyasztást  $1/\alpha^2$ -tel csökkenti, így a disszipációsűrűség változatlan marad [3].

A méretcsökkentéssel párhuzamosan felerősödő másodlagos, illetve parazita hatások egy részét új technológiai megoldások – például réz huzalozás, feszített szilícium, HKMG<sup>4</sup>, finFET<sup>5</sup> – alkalmazásával küszöbölték ki, amelyek összetettebb, így drágább gyártási eljárást eredményeztek, de a korábbi skálázódási szabály érvényét veszítette: a disszipációsűrűség mégis növekszik. Az integrált áramkörök által disszipált, a környezet felé eltávolítandó hő mennyiségét a gazdaságos hűtési megoldások<sup>6</sup> korlátozzák (1.1. ábra).

<sup>2</sup> CMOS, *Complementary Metal Oxide Semiconductor*

<sup>3</sup> MOSFET, *Metal Oxide Semiconductor Field Effect Transistor*

<sup>4</sup> HKMG, *High  $\kappa$  Metal Gate*

<sup>5</sup> finFET, *fin Field Effect Transistor*

<sup>6</sup> Tokozás és kártya szinten konduktív hőátadás, rendszerszinten konvektív hőtranszport.

A nagy integráltságú (VLSI<sup>7</sup>) áramkörök fejlődése az utóbbi másfél évtizedben egyre növekvő disszipációsűrűség, és az ennek következtében kialakuló termikus problémák miatt akadályokba ütközött [4]. A disszipációs korlát miatt az összetett integrált áramkörök nem minden részegysége üzemelhet maximális frekvencián (*dim/dark silicon*). Újabb gyártástechnológiák bevezetésével az ilyen módon kihasználatlan tranzisztorok száma növekszik, relatív költségük azonban egyre kisebb. Ezeket a szempontokat figyelembe véve egy új VLSI áramkör tervezése során a korábbi megoldásokhoz képest jellemzően egyre több funkciót dedikált hardverben valósítanak meg, így csökkentve a fogyasztást. Az egy lapkára integrált különböző funkcionalitású áramkörökkel megvalósított rendszerek gyűjtőneve SoC<sup>8</sup>, amelyek tartalmazhatnak általános és alkalmazás specifikus adatfeldolgozó processzorokat, rádiófrekvenciás és analóg áramköröket.

A dedikált funkciójú részegységek mellett a kihasználtság növeléséhez új módszereket is kidolgoztak nagy számítási teljesítményű VLSI áramkörök esetén. A modern processzorok, SoC-ok az igénybevétel és a lapka különböző pontjain mért hőmérsékletek szerint szabályozzák a feszültséget és az órajel-frekvenciát (DVFS<sup>9</sup>). A disszipációt és hőmérsékletet futási időben szabályozó módszereket a szakirodalomban dinamikus hőmérséklet-menedzsment (DTM<sup>10</sup>) gyűjtőnévvel illetik.

A gyártástechnológia fejlődésének következtében szükséges a tervezői produktivitás növelése. A rövid piacra kerülési idő, így a versenyképesség megtartásához az integrált áramkörök tervezése során számos automatizált folyamatot (EDA<sup>11</sup>) alkalmaznak. Ezek az EDA eszközök<sup>12</sup> segíthetik a leendő áramkör tervezését és vizsgálatát, szimulációját is.

Amíg az úgynevezett *mainstream* félvezető eszközök<sup>13</sup> gyártási technológiájában alapvető változás nem áll be, a leendő élvonalbeli integrált áramkörök működését egyre nagyobb mértékben fogják befolyásolni a termikus viszonyok. Ez indokolja azt, hogy a tervezési folyamat során – lehetőleg már a korai fázisban is – figyelembe lehessen venni az integrált áramkörök lapka hőmérséklet-eloszlásának az áramkör funkcionális viselkedésére gyakorolt hatását. Ehhez olyan szimulációs eszközök fejlesztésére van szükség, amelyek lehetővé teszik az áramkörök termikus-funkcionális együttes szimulációját. Ennek legelterjedtebb változata az analóg áramkörök tervezését támogató úgynevezett elektro-termikus szimuláció.

## A kutatás előzményei

A Budapesti Műszaki Egyetem Elektronikus Eszközök Tanszékén az integrált áramkörök elektro-termikus szimulációjával először Székely Vladimír professzor foglalkozott, kutatómunkája világszinten is úttörő volt. A tanszék az ő munkája nyomán sokéves múltra tekinthet vissza ezen a területen, így Poppe András témavezetőm útmutatásával, Pohl László, Timár András, Nagy Gergely és Horváth Péter eredményeiből kiindulva kezdhettem el a doktori kutatómunkámat.

## A kutatómunka célja

A kutatásom célja egy olyan szimulációs módszer és keretrendszer kidolgozása volt, amely a tervezési folyamat több lépésében is lehetővé teszi a digitális és kevert jelű integrált áramkörök működése során fellépő hőmérsékletváltozások figyelembevételét annak funkcionális szimulációja

---

<sup>7</sup> VLSI, *Very Large Scale Integration*

<sup>8</sup> SoC, *System on Chip*

<sup>9</sup> DVFS, *Dynamic Voltage Frequency Scaling*

<sup>10</sup> DTM, *Dynamic Thermal Management*

<sup>11</sup> EDA, *Electronic Design Automation*

<sup>12</sup> Tervezési keretrendszerek, hardver szintézis programok, vagy különböző szimulációs és verifikációs programok.

<sup>13</sup> Szilícium alapú eszközök, például processzorok, memóriák, kiegészítő logikai áramkörök.

---

során. A munkám eredménye egy olyan keretrendszer, amely elsősorban létező eszközöket – például szimulátor programokat, modellező nyelveket – használ, így a szimulációs módszer gyakorlati alkalmazására is lehetőség nyílik.

A dolgozat 2. fejezete áttekintést nyújt a szakirodalomban és a gyakorlatban alkalmazott tervezési eszközökről, modellezési és szimulációs módszerekről. A 3. fejezetben röviden bemutatom a felhasznált demonstrációs modelleket és eszközöket, a 4–7. fejezetekben részletezem a kutatási eredményeimet, amelyeket a értekezésem végén a 8. fejezetben foglalom össze.

---



## 2. fejezet

# Irodalmi áttekintés

A természettudományos és mérnöki gyakorlatban használt modellek a valóság egy – a vizsgálat szempontjából érdekes – részletét utánozzák, más részleteket elhanyagolnak. A szimulációk során ezeknek a modelleknek a viselkedése ellenőrizhető, ami alapján következtetések vonhatóak le a modell alapjául szolgáló jelenség működésére vonatkozóan. Integrált áramkörök tervezése során különösen nagy jelentősége van a különböző modellezési és szimulációs módszereknek, mert az IC<sup>1</sup>-k összetettsége és a kis szériás gyártás magas költsége nem teszi lehetővé prototípus áramkörök hosszas tesztelését és mérését.

### 2.1. IC-k modellezése a tervezés során

Integrált áramkörök tervezésekor különböző modellek segítségével vizsgálható a leendő áramkör egyes tulajdonságai. Az elsődleges modellnek tekinthető funkcionális leírás implementálása mellett szükség lehet a leendő áramkör más paramétereinek – például a fogyasztásnak, területigényének – modellezésére is.

#### 2.1.1. Az integrált áramkörök tervezési folyamata

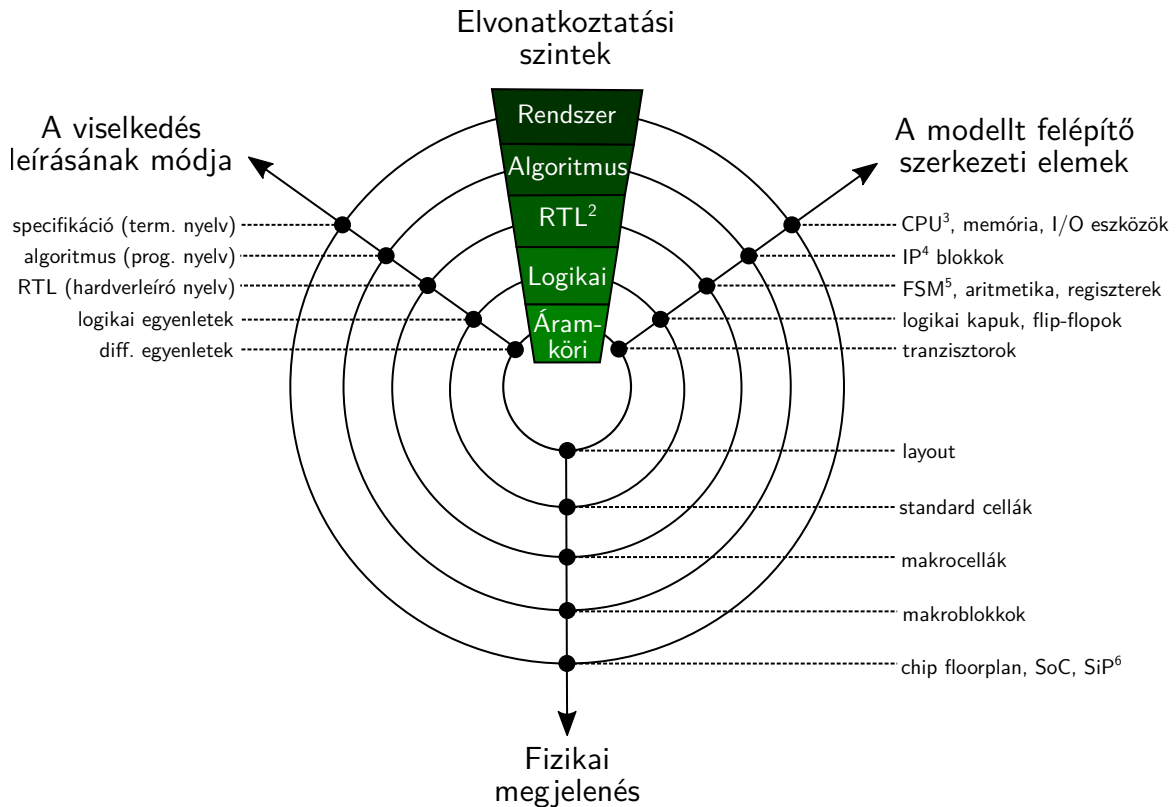
##### Elvonatkoztatási szintek a tervezési folyamatban

A digitális IC-k tervezése során a leendő áramkört különböző elvonatkoztatási szinteken modellezik. Ezekben a modellekben a viselkedés leírása, a szerkezeti elemek és azok fizikai megjelenése csak az adott absztrakciós szint által kívánt részletességgel jellemzi a leendő áramkör egyes tulajdonságait. A tervezési folyamat különböző elvonatkoztatási szintjein implementált modelleket ezen szempontok szerint ábrázolja a Gajski-Kuhn Y-diagram [5]. A 2.1. ábra ennek a diagramnak egy napjainkban is érvényes, modernizált változata.

A viselkedés leírása az áramkör funkcionalitását tükröző médium, elvonatkoztatási szint függvényében lehet természetes nyelven megfogalmazott specifikáció, programozási vagy hardverleíró nyelven megvalósított modell vagy egyéb, gépi feldolgozásra is alkalmas reprezentáció. A modellt felépítő szerkezeti elemek az adott elvonatkoztatási szintre jellemző tervezési egységek, amelyek még önállóan tekinthető funkcióval rendelkeznek. Ezen tervezési egységek fizikai megjelenési formája szintén az elvonatkoztatási szinttől függ. A különböző absztrakciós szinteken használt fontosabb modellező eszközök tulajdonságai a következők.

---

<sup>1</sup> IC, *Integrated Circuit*



2.1. ábra. Modernizált Gajski-Kuhn-féle Y-diagram [5]

**Specifikáció.** A specifikáció általában egy természetes nyelven megfogalmazott dokumentum, szigorú formai követelmény nélkül. Az értelmezést egyértelműsítő, könnyítő ábrákat – például blokkdiagramot, szekvenciadiagramot – tartalmazhat.

**Algoritmusszint.** Az algoritmusszintű modelleket futtatható specifikációnak is szokás nevezni, mivel ez az első formális leírása az áramkörnek. Hagyományos C vagy C++ programozási nyelvek mellett az IC tervezés során SystemC-t vagy SystemVerilog-ot is használnak a viselkedés absztrakt leírásához.

**Regiszterátviteli szint.** A regiszterátviteli szinten jelenik meg a modellben az időzítés fogalma. Az RTL modell implementálásakor az áramkör a fázisjelek aktív élei között elvárt viselkedését kell megfogalmazni hardverleíró nyelvek (HDL<sup>7</sup>), például SystemC, SystemVerilog, VHDL<sup>8</sup> vagy Verilog felhasználásával.

**Logikai szint**<sup>10</sup>. Ezen az elvonatkoztatási szinten az áramkört modelleket jellemzően az RTL tervezésnél is használt hardverleíró nyelveken implementálják, a fő különbséget a modell

<sup>2</sup> RTL, *Register Transfer Level*

<sup>3</sup> CPU, *Central Processing Unit*

<sup>4</sup> IP, *Intellectual Property*

<sup>5</sup> FSM, *Finite State Machine*

<sup>6</sup> SiP, *System in Package*

<sup>7</sup> HDL, *Hardware Description Language*

<sup>8</sup> VHDL, *VHSIC<sup>9</sup> Hardware Description Language*

<sup>9</sup> VHSIC, *Very High Speed Integrated Circuits*

<sup>10</sup> A logikai szintet a továbbiakban kapuszintnek is nevezem.

granularitása jelenti, a legkisebb szerkezeti elemek logikai kapuk, flip-flopok. Jellemző, hogy egy IC tervezőrendszerben a kapuk logikai modelljeit előre megtervezett, bizonyos szabályokat követő layout rajzolatukkal együtt úgynevezett standard cellakönyvtárakban bocsájtják a tervezők rendelkezésére.

**Áramköri szint.** Az áramköri vagy más néven tranzisztorszintű leírások áramköri elemek listáját (*netlist*) tartalmazzák az áramkör-szimulátor által megkövetelt – például SPICE<sup>11</sup> netlist – formátumban. A tranzisztorszintű leírásokban koncentrált paraméteres hálózati modell elemeihez rendelt fizikai mennyiségekkel jellemzik az áramkör működését. A szimulációs eredmények pontossága szempontjából meghatározó a felhasznált tranzisztormodellek – például BSIM3<sup>12</sup> – és az azokhoz rendelt eszközparaméterek pontossága.

### Az áramkörök megvalósításának technológiái

Adott funkciót ellátó magas szintű modelltől a tervezési folyamat végén gazdasági és technikai követelmények függvényében különböző technológiával megvalósított áramkörök készülhetnek el. A teljesen egyedi (*full custom*) áramkörök tervezési költsége a több ráfordított mérnökóra miatt, a gyártási költség pedig az egyedi maszkok gyártása miatt magas. A nagyobb fokú automatizálhatóság következtében a standard cellák felhasználásával készülő áramkörök tervezése egyszerűbb, a gyártáshoz azonban szintén egyedi maszkok szükségesek. A full custom és a standard cellás alkalmazás specifikus áramkörök (ASIC<sup>14</sup>) gyártása csak nagy sorozatban gazdaságos.

Újrakonfigurálható eszközökkel (FPGA<sup>15</sup>, CPLD<sup>16</sup>) lehetségessé válik az áramkör funkciójának utólagos meghatározása, így a gyártási költségek több tervezőcsapat között oszlanak el, azonban az áramkör fogyasztása és órajelfrekvenciája kedvezőtlenebb lesz az ASIC megvalósításokhoz képest.

### A tervezési folyamat iránya

A tervezési folyamat az áramkör megvalósításának technológiájától függően történhet a magas absztrakció felől lefelé haladva (*top-down*), vagy építkezhet alulról felfelé (*bottom-up*), de gyakran előfordul a két stratégia kombinációja is. A teljesen egyedi áramkörök esetén az esetleges magas szintű modell implementálása után áramköri szinten tervezik meg az egyes komponenseket, és a layout tervezésének nagy része ugyancsak kézi módszerrel történik. Standard cellás áramkörök tervezése top-down módon történik, azonban felhasználja a standard cellakönyvtárban található korábban áramköri és layout szinten megtervezett kapukat, így ez egy kevert stratégiának is tekinthető. Az újrakonfigurálható eszközöket felhasználó tervezési módszer nagyon hasonlít a standard cellás áramkörök tervezésére, a különbség csak a regiszterátviteli szintet követő kapusintű leírásra való kifejtés során számottevő, azaz ez is tekinthető a top-down és bottom-up eljárások keverékének. A 2.2. ábrán a standard cellás áramkörök tervezési folyamata látható a specifikációtól a gyártásra kész layout reprezentációig.

**Algoritmus implementálása.** A követelmények és a specifikáció meghatározását követően a tervezők megvalósítják a leendő áramkör algoritmus-szintű modelljét. Ezen az elvonatkoztatási szinten használhatnak hagyományos programozási nyelveket (C, C++) vagy kifejezetten ilyen

---

<sup>11</sup> SPICE, *Simulation Program with Integrated Circuit Emphasis*

<sup>12</sup> BSIM, *Berkeley Short-Channel IGFET*<sup>13</sup> *Model*

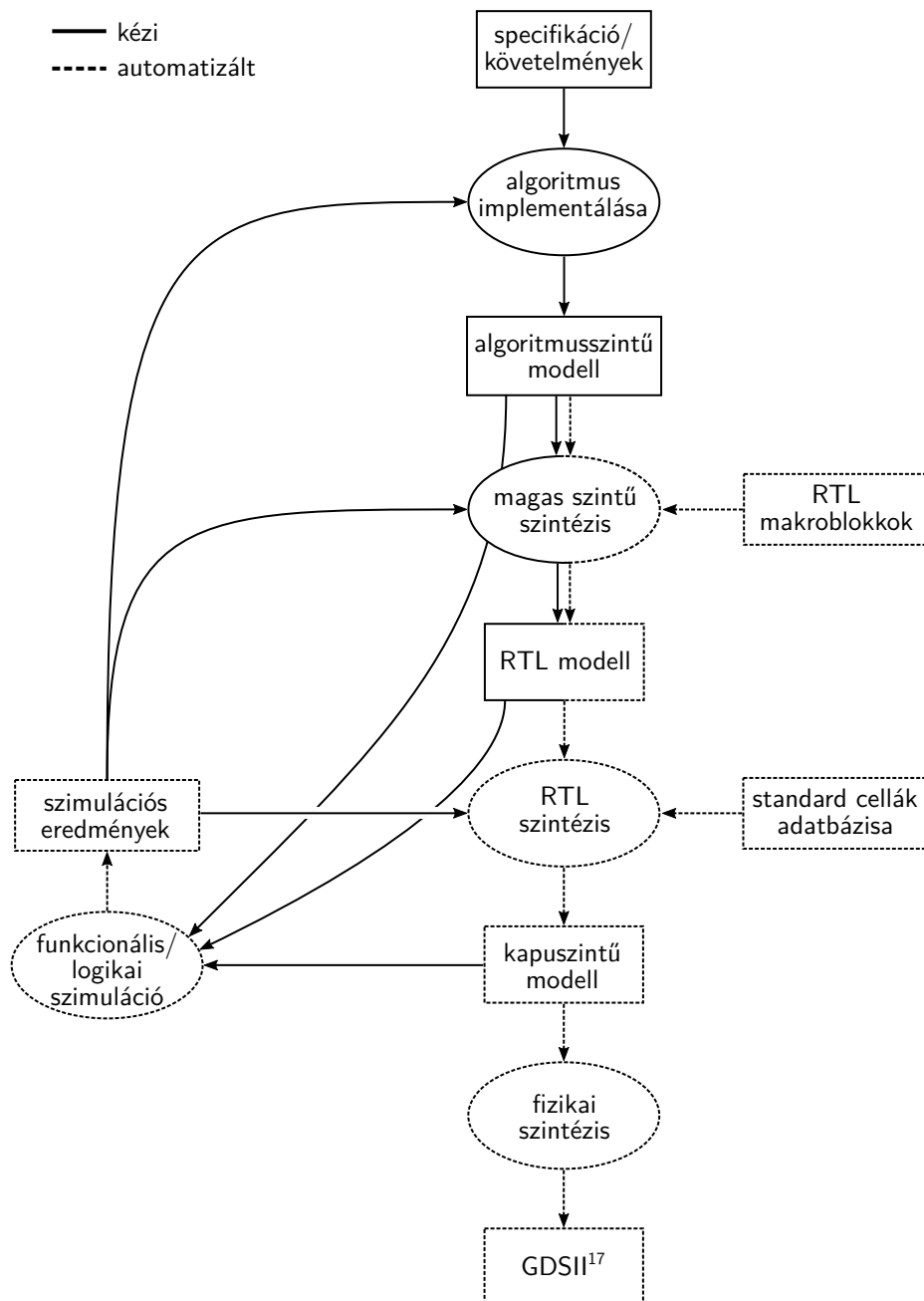
<sup>13</sup> IGFET, *Insulated Gate Field Effect Transistor*

<sup>14</sup> ASIC, *Application Specific Integrated Circuit*

<sup>15</sup> FPGA, *Field Programmable Gate Array*

<sup>16</sup> CPLD, *Complex Programmable Logic Device*

---



2.2. ábra. Standard cellás integrált áramkörök tervezési folyamata

absztrakciós szintű modellek implementálásához használatos nyelveket is (SystemC, SystemVerilog). Az algoritmusszintű leírás már használható az áramkör funkcióinak tesztelésére, processzor esetén ez a modell már szoftverek futtatására is alkalmas, lehetővé téve akár a hardver-szoftver együttes tervezést is.

**Magas szintű szintézis.** Az algoritmusszintű modellből automatizált vagy kézi úton történhet a regiszterátviteli szintű leírás előállítása. Az automatizált magas szintű szintézis (HLS<sup>18</sup>) programok – például Xilinx® Vivado® HLS, Mentor® Catapult® HLS, Cadence® Stratus™ HLS – használata ugyan gyorsíthatja a tervezési folyamatot, azonban gyakran megkötéseket

<sup>17</sup> GDS, *Graphic Database System*

<sup>18</sup> HLS, *High Level Synthesis*

ad a magas szintű modell kódolására vonatkozóan, és a tervezőknek kisebb befolyása van a leendő áramkör architektúrájára. Emiatt napjainkban továbbra is a regiszterátviteli szint a digitális integrált áramkörök implementálásának elsődleges elvonatkoztatási szintje<sup>19</sup> [6].

**RTL szintézis.** Általában az RTL elvonatkoztatású modellek a bemenetei az automatizált áramköri szintézis folyamatoknak. Az RTL vagy más néven logikai szintézis során az RTL modellből a szintézerprogram kapusintű leírást állít elő, miközben különböző optimalizációkat és ellenőrzéseket is végrehajt. Ilyen optimalizáció lehet például állapotgépek esetén az állapot-minimalizálás, vagy az egyes csomópontok terhelésének (*fan-out*) optimalizálása és a cellák méretezése.

**Fizikai szintézis.** A fizikai szintézis során a generált netlistát felhasználva készül el az áramkör részletes layoutja. A folyamat közben a netlista kis mértékben módosulhat, hogy az áramkör megfeleljen az időzítési feltételeknek a cellák elhelyezése után is, például hosszú jelvezetékbe buffereket illesztnek, valamint az időzítési szempontból kritikus órajelelosztó-hálózatokat is ebben a fázisban alakítják ki.

Az elkészült layoutot statikusan ellenőrzik, hogy megfelel-e a tervezési szabályoknak (DRC<sup>20</sup>), valamint a layoutból visszafejtett kapcsolást összehasonlítják az eredetivel (LVS<sup>21</sup>). A fizikai szintézis előtti és az azt követően előálló modellek időzítési viszonyait – az időzítési feltételek teljesülését – statikus időzítés-analízissel (STA<sup>22</sup>) ellenőrzik.

### Szimulációk a tervezés során

Az integrált áramkörök tervezése során – más tervezési eljárásokhoz hasonlóan – visszacsatolások, iterációs hurkok szükségesek az optimális megoldás megtalálásához a tervezési térben. IC-k esetén ilyen iterációk során szimulációval vizsgáljuk az áramkör adott modelljét és hasonlítjuk össze a követelményekkel, illetve a tervezés előző lépésében előállított eredményekkel. Az áramköri modell elvonatkoztatási szintjének függvényében különböző szempontok képezik a vizsgálat tárgyát, és eltérő szimulátorok használatosak.

**Algoritmusszint.** Hagyományos programnyelven implementált modell fordítása és futtatása, SystemC vagy SystemVerilog nyelvű modellek szimulációja erre alkalmas szimulátor programmal, például Mentor<sup>®</sup> Questa<sup>™</sup> Simmel. A szimuláció sebessége lehetővé teszi nagyobb szimulációs időtartamok vizsgálatát még bonyolult rendszerek esetén is, így a szoftver stack működése is vizsgálható a hardver szimulációja során.

**Regiszterátviteli szint.** Hardverleíró nyelven implementált modellek szimulációja az erre alkalmas eseményvezérelt szimulációs programmal. Ilyen szimulátor például a Mentor<sup>®</sup> Questa<sup>™</sup> Sim vagy a Cadence<sup>®</sup> IES<sup>23</sup>. Az eseményvezérelt szimuláció lehetővé teszi a hardver alapvetően párhuzamos természetének szimulációját hagyományos egy szál processzoros környezetben. A szimuláció erőforrásigénye nagyobb, mint az algoritmusszintű modellek esetén, így tervezői szempontból tolerálható futási időket feltételezve ennek a szimulációnak kisebb az átfogása a szimulált idő tekintetében.

---

<sup>19</sup> Horváth Péter kollégám a magas szintű szintézis hátrányainak feloldásával, a tervezés absztrakciós szintjének emelésével foglalkozott kutatómunkája során [7].

<sup>20</sup> DRC, *Design Rule Check*

<sup>21</sup> LVS, *Layout Vs. Schematic*

<sup>22</sup> STA, *Static Timing Analysis*

<sup>23</sup> Cadence<sup>®</sup> IES, Cadence<sup>®</sup> *Incisive Enterprise Simulator*

---

**Logikai szint.** Logikai vagy kapuszinten ugyanazok a HDL-ek használhatóak, mint RTL-en, így a szimulációs eszközök is megegyeznek. Mivel a két elvonatkoztatási szinten implementált modellek granularitásában nagy különbség van, a finomabb felbontású kapusztintú modell szimulációja jelentősen több erőforrást igényel. Különösképpen igaz ez a fizikai szintézist követő úgynevezett post-layout szimulációra, amely során a kapuk elhelyezkedése alapján számított időzítések pontosan szimulálhatóak. A nagy számításigény miatt a post-layout szimulációt bizonyos gyártók az újabb eszközeikkel nem támogatják (Intel® Quartus® Prime), helyette alternatívaként statikus időzítésanalízist javasolnak az időzítési feltételek ellenőrzéséhez.

**Áramköri szint.** Áramköri szintű szimulátor programok a Kirchoff törvények alapján felírt hálózategyenleteket oldják meg, a szimuláció erőforrás- és számításigénye sokszorosa a kapusztintúnek. A tranzisztorszintű áramkör-szimuláció eredményként a hálózat egyes ágain átfolyó áramokat és a hálózat csomópontjain fellépő potenciálokat szolgáltatja, ami digitális áramkörök tervezésekor túlságosan részletes, nehezen értelmezhető információhalmaz.

### 2.1.2. Disszipáció modellezése

A tervezési folyamat előrehaladtával a már implementált viselkedési modell módosítása egyre nagyobb költséggel jár a többlet munkaóra ráfordítása miatt. Az ilyen jellegű átdolgozások elkerülése érdekében a funkció mellett az IC sok más, a működést közvetlenül nem befolyásoló paraméterét – például területigényét vagy disszipációját – is modellezzük.

A CMOS áramkörök fogyasztását két összetevő határozza meg. A statikus fogyasztás folyamatos, az áramkör aktivitásától független, míg a dinamikus fogyasztás az áramkör aktivitásával arányos. A statikus fogyasztás CMOS áramkörökben az egyik legjelentősebb parazitahatás. Fő komponensei a küszöbalatti áramból, a drain és source diódák szivárgási áramából és a gate szigetelőn folyó alagútáramból adódnak. A dinamikus fogyasztást a töltés-pumpálás és az egymásba vezetés okozza, így ezek egyenesen arányosak a kapcsolási aktivitással.

A modern integrált áramkörök disszipációsűrűsége miatt egyre nagyobb jelentősége van a fogyasztás optimalizálásának és az energiahatékonyság növelésének, amelyhez az IC disszipációját is modellezni kell. A modern CMOS gyártástechnológiák esetében alkalmazott kapusztintú disszipációmodellek részletes magyar nyelvű áttekintése Timár András kollégám PhD értekezésének 4. fejezetében található [8]. A következőkben röviden áttekintem a magasabb absztrakciós szinteken alkalmazott disszipációmodellek irodalmát.

### Integrált keretrendszerek

Algoritmusszinten a funkcionális modell nem, vagy csak minimális információt tartalmaz a leendő IC architektúrájára vonatkozóan. Az ilyen elvonatkoztatású, disszipációt becsülő módszerek gyakran olyan architektúrális komponensek fogyasztásának becslésére alkalmasak, amelyeket már megterveztek és felhasználtak korábban megvalósított áramkörökben. Ilyen megoldás a CACTI<sup>24</sup> [9], amely gyorsítótárak és memóriák időzítésének, területigényének és fogyasztásának becslésére használható, valamint a Wattch [10] és a McPAT<sup>25</sup> [11] keretrendszerek, amelyek integráltan tartalmazzanak valamennyi, superskalár és többmagos processzorok disszipációjának becsléséhez szükséges modellkomponenst. A modernebbnek tekinthető McPAT modell extrapolációs módszerekkel a gyártástechnológia hatását is figyelembe veszi a disszipáció becslésekor.

---

<sup>24</sup> CACTI, *Cache Access and Cycle Time model*

<sup>25</sup> McPAT, *Multicore Power Area Timing model*

---

Ezeket a népszerű disszipációmodelleket és területigény-becslő eljárásokat több kutatásban is felhasználták. S. S. Kumar és társai keretrendszert valósítottak meg, amelybe integrálták a Wattch és CACTI modelljeit [12]. A Sniper x86 processzor szimulátorhoz pedig a McPAT keretrendszert illesztették, amit futásidőben használva a szimulált processzor fogyasztását lehet megbecsülni [13].

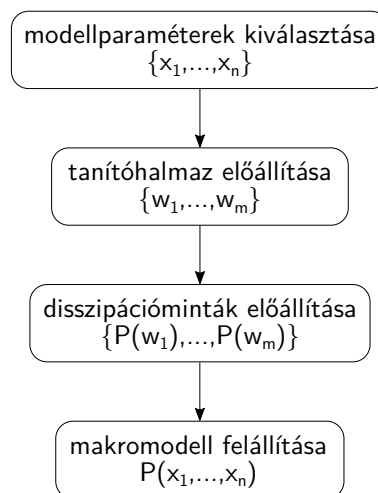
Ezen disszipációmodellek előnye, hogy az áramkört alkotó architektúrális komponenseken kívül egyéb információt nem igényelnek, így a modell együtthatóit sem kell hosszas karakterizálási eljárással meghatározni. Hátrányuk, hogy csak már létező komponensek disszipációját tudják becsülni és a becslés esetenként pontatlan lehet [14].

### A disszipáció makromodellezése

A disszipáció makromodellezése oly módon küszöböli ki a fenti problémát, hogy a funkcionális leírásban megfigyelt paraméterek alapján becsüli a fogyasztást [15]:

$$\hat{P} = P(x_1, \dots, x_n), \quad (2.1)$$

ahol  $\hat{P}$  a becsült fogyasztás,  $x_i$  a megfigyelt paraméter,  $n$  a paraméterek száma. Elvonatkoztatási szint függvényében ilyen paraméter lehet az áramkörü modul bizonyos jelek értéke vagy aktivitása, vagy akár annak hőmérséklete is.



2.3. ábra. Disszipáció makromodellezés lépései [15]

A fogyasztás makromodellezésének lépései a 2.3. ábrán láthatóak. Első lépésben meg kell határozni a makromodell paramétereit, vagyis ki kell választani a megfigyelni kívánt változókat. Kevés választott paraméter egyszerűbb makromodellt eredményez, de ha a „paramétertér” kicsi, a modell pontatlan lehet. A második lépésben a makromodell karakterizálásához szükséges tanítóhalmazt kell előállítani. E halmaz számosságának sokkal nagyobbának kell lennie, mint a paraméterek száma ( $m \gg n$ ). A harmadik lépés a karakterizálás, azaz a referencia modell kiértékelése a tanítóhalmaz összes elemére ( $P(w_i)$ ). Utolsó lépésben a referencia értékek és a tanítóhalmaz elemei alapján fel lehet állítani a makromodellt.

Disszipáció makromodellek használhatóak algoritmus szinten [16, 17, 18] és regiszterátviteli szinten is [19, 20, 21]. Hátrányuk a korábban említett keretrendszerekhez képest, hogy karakterizálás szükséges (harmadik lépés), amelyhez szükség van referencia fogyasztási értékekre is.

S. Ahuja és munkatársai magas szintű szintézissel állítják elő az algoritmus szintű modelltől az RTL modellt, amelynek disszipációmodelljét referenciaként használják az algoritmus szintű disszipációmodell karakterizálásához [17, 18]. G. B. Vece és kollégái ezzel szemben kézi úton valósították

meg az RTL modellt, amiből azután kapusintű modellt szintetizáltak és ez utóbbit használták az algoritmusszintű disszipáció makromodell karakterizálásához [16]. Magas szintű szintézissel előállított regiszterátviteli szintű leírás gyakran nem olyan hatékony az erőforrásigényt tekintve, mint a kézi úton tervezett RTL modell (2.1.1. fejezet), így ezzel a módszerrel karakterizált magas szintű disszipáció modell nem biztos, hogy a tényleges áramkör fogyasztását fogja tükrözni. Az utóbbi megközelítés hátránya a jelentős többletmunka, amely három hónappal is meghosszabbíthatja a tervezési folyamatot [16]. Ez megnehezíti az algoritmusszintű modell tesztelését, fogyasztásának elemzését.

Regiszterátviteli szintű makromodellek megvalósításakor a referencia fogyasztási adatokat rendszerint kapusintű leírás szimulációjával határozzák meg. A kapusintű netlista rendelkezésre állhat, ha a makromodell egy létező áramköri egységhez készül, de előzetes RTL szintézissel is elő lehet állítani. Az RTL szintézis – a magas szintű szintézissel ellentétben – évek óta része a hagyományos tervezési folyamatnak. Az RTL modelltől szintetizált áramkör tulajdonságait a modell implementálásakor és megfelelő tervezési kényszerek meghatározásakor kézben tudják tartani az regiszterátviteli szinten dolgozó tervezők.

A regiszterátviteli szintű disszipáció makromodellek közös tulajdonsága, hogy a makromodell paraméterei ( $x_i$ ) az RTL leírások jelei, de a megvalósításuk nagyon eltérő lehet. A modellezés nehézségét az jelenti, hogy hogyan lehet minél kevesebb paraméterrel, minél rövidebb karakterizálással jellemezni a funkcionális (RTL) modell jelei és annak fogyasztása közötti nemlineáris kapcsolatot. A makromodellekben alkalmaztak már többszörös lineáris regressziót és regresszió fát [19, 20], táblázatos (LUT<sup>26</sup>) módszert [22], és neurális hálózatot is [21].

## Cellakönyvtárak teljesítménymodelljei

A félvezetőgyártók a gyártástechnológiájuk tulajdonságait, az azon megvalósítható standard cellákat az úgynevezett PDK<sup>27</sup>-ban bocsájtják a tervezők rendelkezésére. Ennek a csomagnak a része a standard cellák HDL-ben – rendszerint Verilog nyelven – megvalósított viselkedési modelljei, valamint a cellák geometriai méreteire, késleltetésére és fogyasztására vonatkozó információk. Ez utóbbi modellezésére több eljárás is ismert, amelyek összetettségükben, az adatbázisok méretében és pontosságukban különböznek [8].

Amennyiben nem áll rendelkezésre ilyen adatbázis, a standard cellák áramköri szintű modelljeinek SPICE szimulációjával is elő lehet állítani a szükséges adatokat [23].

## Tranzisztormodellek

Az áramkörök absztrakciós szintjén a tranzisztorok modelljei fizikai mennyiségekkel – feszültséggel, árammal – jellemzik azok viselkedését, így önálló disszipációmokmodellek ebben az esetben nem szükségesek.

A bipoláris és térvezérlésű tranzisztorok működésének jellemzésére számtalan modellt kidolgoztak, amelyek között a fő különbség az, hogy milyen effektusokat vesznek figyelembe. Az Ebers-Moll modell [24] a pn átmenetek fizikai modelljével jellemzi a bipoláris tranzisztor működését. Ezzel a tranzisztor viselkedése minden működési tartományban leírható, azonban másodlagos effektusokat nem vesz figyelembe. A Gummel-Poon [25] modell ezeket a hatásokat is modellezi, idealizált paraméterek esetén pedig visszavezethető az Ebers-Moll modellre. A térvezérlésű tranzisztorok ma az

---

<sup>26</sup> LUT, *Look Up Table*

<sup>27</sup> PDK, *Process Design Kit*

---



egyik legszélesebb körben használt modellje a BSIM [26], amelyet különböző – például SOI<sup>28</sup> vagy bulk – technológián megvalósított tranzistorokhoz folyamatosan továbbfejlesztnek.

## Szabványosított disszipációmodellek

A tervezési folyamat különböző elvonatkoztatási szintjein a (várható) fogyasztás modellezése már egy jól körbejárt, kutatott terület, azonban a szabványosítása még nem teljeskörű. A többek között elektronikai szabványok kiadásával foglalkozó IEEE<sup>29</sup> először az alacsony fogyasztású eszközök verifikációjához használható IEEE Std 1801-2015 szabványt adta ki [27], amely a tervezők számára lehetővé teszi a tápellátás viselkedésére vonatkozó követelmények megfogalmazását (*power intent*). Ennek a szabványnak a célja tehát nem a fogyasztás kvantitatív modellezése, hanem a funkcionális verifikáció megkönnyítése kis fogyasztású áramkörök esetén.

Az IEEE két projektet támogat a disszipációmodellek szabványosításának érdekében [28, 29]. A leendő szabványok a disszipációmodellekkel szemben elsősorban formai követelményeket fogalmazhatnak meg és nem a modellezés módját szabványosítják, mivel – a feljebb leírtaknak megfelelően – az elvonatkoztatási szint, a funkcionális modell granularitása és egyéb paraméterek függvényében eltérő disszipációmodellt érdemes alkalmazni.

## 2.2. Termikus szimulációs módszerek

Az 1. fejezetben tárgyalt okok következtében az integrált áramkörök termikus vizsgálata elengedhetetlen a tervezés során. A termikus szimulációk elsődleges célja a működés során kialakuló hőmérsékleti viszonyok meghatározása a chipen és annak környezetében, a chip hűtéséhez alkalmazott megoldások (például hőnyelő, *heat sink*) minősítése. Ezekben a rendszerekben a hőterjedés három módozata – kondukción, konvekción, sugárzás – közül a vizsgált tartomány függvényében a kondukción vagy konvekción történő hőtranszport a jellemző. Chip, tokozás és NYHL<sup>30</sup> szintjén kondukción, rendszer szintjén konvekción megvalósuló transzport domináns [30].

$V$  térfogatú elemre felírható az energiamegmaradás törvénye:

$$\int_V c_v T dV = \int_t \oint_A \lambda \cdot \text{grad}(T) dA dt + \int_t \oint_A c_v \bar{v} T dA dt + \int_t \oint_A \sigma (T - T_a)^4 dA dt + \int_t \int_V p dV dt, \quad (2.2)$$

amelyben  $c_v$  az anyag térfogategységre eső hőkapacitása,  $\lambda$  annak hővezetése,  $T$  a hőmérséklete,  $T_a$  a környezet hőmérséklete,  $p$  a disszipációsűrűség és  $t$  az idő. Ebből az energiamérlegből levezethető a hőtranszport folyamatokat leíró differenciálegyenlet (A függelék):

$$c_v \frac{\partial T}{\partial t} = \text{div}(\lambda \cdot \text{grad}(T)) + c_v T \cdot \text{div}(\bar{v}) + c_v \bar{v} \cdot \text{grad}(T) + \text{div}(\sigma (T - T_0)^4) + p. \quad (2.3)$$

Termikus szimuláció során ezt a differenciálegyenletet kell megoldani. A hőmérsékleti sugárzás a szilícium alapú integrált áramkörök üzemi hőmérsékleti tartományában elhanyagolható, lapka méretű szimulációs tartomány esetén jellemzően a konduktív hőtranszportot szükséges modellezni, ekkor a fenti egyenlet a

$$c_v \frac{\partial T}{\partial t} = \text{div}(\lambda \cdot \text{grad}(T)) + p \quad (2.4)$$

differenciálegyenletre egyszerűsödik, amelynek a megoldására számos numerikus megoldó algoritmus létezik.

<sup>28</sup> SOI, *Silicon On Insulator*

<sup>29</sup> IEEE, *Institute of Electrical and Electronics Engineers*

<sup>30</sup> NYHL, *Nyomtatott Huzalozású Lemez*

### 2.2.1. Numerikus módszerek

A probléma modellezéséhez és a (2.4) egyenlet megoldásához a termikus szimulátorok egy része diszkretizálja a szimulációs tartományt, majd az így kapott közelítést oldja meg. A három legismertebb eljárás a véges elem (FEM<sup>31</sup>), a véges differenciák (FDM<sup>32</sup>), valamint a véges térfogatok (FVM<sup>33</sup>) módszere.

**FEM.** A véges elemek módszere a tartományt elemekre bontja, a megoldást pedig ezeken a résztartományokon felvett bázisfüggvények lineáris kombinációjaként keresi. Előnye ennek az eljárásnak, hogy a tartomány rugalmasan bontható részekre, azonban a határokon nem garantált a megmaradási törvények teljesülése [31].

**FDM.** A véges differenciák módszere a szimulációs tartományon felvett rácspontok között diszkretizálja a megoldandó differenciálegyenletet, majd az így kapott differenciaegyenletet kell megoldani. A véges elemek módszeréhez képest ennek a módszernek az használata esetén csak erősen kötött diszkretizálási séma alkalmazható [31].

**FVM.** A véges térfogatok módszere olyan differenciálegyenletek esetén használható, amelyekre alkalmazható a Gauss-tétel. Az eljárás a megoldandó differenciálegyenlet térfogati integrálját felületi integrállá alakítja. A módszer előnye, hogy a megmaradási törvények teljesülnek, és a tartomány térfogatelemekre bontása nem kötött, az egyszerűbb algoritmizálás érdekében azonban gyakran az FDM-hez hasonló sémát használnak [31]. A 6.1. fejezetben részletesebben is bemutatom ezt a módszert.

A fenti numerikus módszereket alkalmazó termikus szimulátorokra példák hozhatóak mind egyetemi kutatócsoportok fejlesztései, mind kereskedelmi megoldások közül is.

#### HotSpot

A HotSpot szimulátort a Virginia Egyetemen fejlesztették [32, 33, 34] 2004-ben. Eleinte csak kompakt modellt (2.2.2. fejezet) tartalmazott, 2006-tól véges differenciák módszerét alkalmazó diszkretizált modell is elérhető lett [32]. Más – például 3D-ICE<sup>34</sup>, SUNRED<sup>35</sup> – szimulátorokhoz képest kötöttebb a termikus modell felépítése, kizárólag a komplett IC lapka, TIM<sup>36</sup>, hőelosztó réteg (*heat spreader*), hűtőborda struktúra szimulálható vele, ettől eltérő modell vizsgálatához módosítani kell a szimulátort. A HotSpot szimulátor forráskódja nyilvános [35], így több kutató is ezt a termikus szimulátort használta munkája során – például [12, 36]. Érdekességképpen megemlítendő, hogy a HotSpot programot az Elektronikus Eszközök Tanszéke, az egykori MicReD Kft.<sup>37</sup> és a Grenoble-i TIMA<sup>38</sup> Laboratórium együttműködésében tervezett termikus teszt chip [37] és mérő kit felhasználásával validálták [38].

---

<sup>31</sup> FEM, *Finite Element Method*

<sup>32</sup> FDM, *Finite Difference Method*

<sup>33</sup> FVM, *Finite Volume Method*

<sup>34</sup> 3D-ICE, *3D Interlayer Cooling Emulator*

<sup>35</sup> SUNRED, *Successive Network Reduction*

<sup>36</sup> TIM, *Thermal Interface Material*

<sup>37</sup> A MicReD Kft. ma a Mentor<sup>®</sup> egy részlege.

<sup>38</sup> TIMA, *Techniques de l'Informatique et de la Microélectronique pour l'Architecture des systèmes intégrés*

---

### 3D-ICE

A 3D-ICE nevű termikus szimulátort a lausanne-i egyetemen fejlesztették [39, 40, 41]. Ezen termikus szimulációs motor újdonsága az, hogy támogatja a mikrocsatornás hűtési megoldások szimulációját is. A csatorna kompakt modelljében a konvektív hőtranszportot vezérelt áramforrások reprezentálják, a modell paramétereit az áramlási tulajdonságok alapján határozzák meg. A HotSpot-hoz hasonlóan a 3D-ICE is a véges differenciák módszerét használja a diszkretizáláshoz. A szimulátor forráskódja szabadon hozzáférhető [42], 2016 júniusáig mintegy ezer letöltést regisztráltak és számos publikációhoz használták. A 3D-ICE szerzői együtt validálták a termikus szimulációs motort és a mikrocsatornás kompakt modellt [41].

### SUNRED

A szukcesszív csomópont-redukción alapuló termikus szimulációs program első változatát az Elektronikus Eszközök Tanszékén fejlesztette ki Székely Vladimír professzor [43, 44]. Az eljárással a véges differenciák módszerével kapott lineáris egyenletrendszer egyenleteinek száma csökken, a redukciót követően direkt módszer is alkalmazható az egyenletrendszer megoldásához. A 3D elrendezésű IC-k termikus szimulációjára is alkalmas szimulátort [45] Pohl László fejlesztette tovább, aminek köszönhetően a SUNRED többmagos processzorokon hatékonyabban fut [46] és a korábban alkalmazott véges differenciák módszere helyett a véges térfogatok módszerét használ a szimulációs tartomány diszkretizálásához [31]. Később OLED<sup>39</sup> modellek [47, 48], és a konvektív hőtranszfer kompakt modellezésével mikrocsatornák [S1, S2] szimulációjára is alkalmas lett a SUNRED. Ezt a termikus motort az Elektronikus Eszközök Tanszék munkatársai validálták egy ilyen célra fejlesztett teszt chipen és OLED fényforrásokon végzett mérésekkel [48, 49].

### Ipari megoldások

A korábban említett egyetemi fejlesztések céleszközök lapka szintű termikus szimulációhoz. Ilyen feladatokra kereskedelmi megoldások is léteznek, azonban az utóbbiak jellemző alkalmazási területe szélesebb spektrumú. Ilyen kereskedelmi szimulátor program például a Mentor<sup>®</sup> FloTHERM<sup>®</sup>, az ANSYS<sup>®</sup> Fluent<sup>®</sup> és Mechanical<sup>®</sup> vagy a COMSOL Multiphysics<sup>®</sup> terméke is.

### 2.2.2. Termikus kompakt modellezés

A termikus kompakt modellezés során a a 2.2.1. fejezetben leírtakkal ellentétben a modellt nem a differenciálegyenlet és a szimulációs tartomány diszkretizálásával állíthatjuk elő, hanem a szimulálandó struktúra és fizikai megfontolások alapján hozzuk létre. A makroszkopikus méretű térfogatok, felületek közötti termikus kölcsönhatás koncentrált paraméterű modellel is jellemezhető, amit a konduktív hőtranszport és az elektromos hálózatok analógiája alapján állítanak fel.

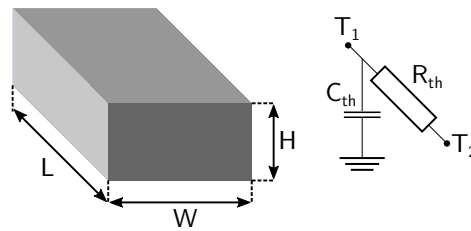
### Elektromos-termikus analógia

A 2.4. ábrán látható homogén tömb két vége között a hőellenállás és hőkapacitás a fizikai méretek,  $\lambda$  fajlagos hőellenállás, valamint a  $c_v$  térfogategységre eső hőkapacitása alapján a 2.5. összefüggés szerint számítható.

---

<sup>39</sup> OLED, *Organic Light Emitting Diode*

---



2.4. ábra. Makroméretű anyagdarab hőellenállása és hőkapacitása

$$R_{th} = \frac{1}{\lambda} \cdot \frac{L}{HW} \quad (2.5)$$

$$C_{th} = c_v \cdot LHW$$

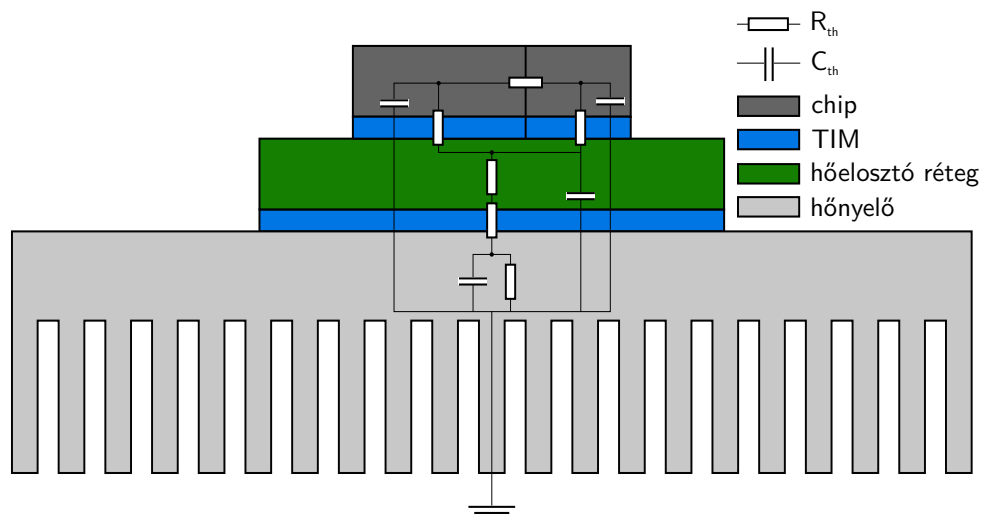
A koncentrált paraméteres kompakt modellekben és elektromos hálózatokban alkalmazott fizikai mennyiségek közötti kapcsolatot a 2.1. táblázat foglalja össze.

2.1. táblázat. Elektromos és termikus mennyiségek analógiája

Elektromos mennyiség	Termikus mennyiség
töltés [C]	hőenergia [J]
áram [A]	hőáram [W]
feszültség [V]	hőmérséklet-különbség [K]
ellenállás [ $\Omega$ ]	hőellenállás [K/W]
kapacitás [F]	hőkapacitás [J/K]

Az elektromos mennyiségek közötti relációnak megfelelő összefüggés felírható a termikus tartományra is, azaz az elektromos Ohm törvény alkalmazható egy „termikus hálózatra”, ahogy létezik a Kirchoff törvények termikus megfelelője is.

### Termikus kompakt modellezési módszerek és az azokat megvalósító programok



2.5. ábra. Egy egyszerű struktúra termikus kompakt modellje

Termikus kompakt modellezés során a cél egy olyan, hőellenállásokból és hőkapacitásokból álló hálózat előállítása, amely a fizikai struktúra kívánt pontjaiban, a struktúrában található

hőforrásokból származó disszipáció esetében szolgáltatja ezen pontok hőmérsékletét. Egy ilyen kompakt modellt mutat be a 2.5. ábra. Az egyes csomópontok potenciálja az adott térfogat hőmérsékletét modellezi a gerjesztés – a chipben felszabaduló Joule-hő, disszipáció – hatására.

A termikus kompakt modellek használatának az egyik előnye a kis számítási igény, amely lehetővé teszi hosszabb időtartamok szimulációját is. Nagy összetettségű integrált áramkörök esetében a termikus hálózati modell bonyolultsága elérheti például a véges differenciák módszerével kapott diszkrétizált összetettséget, azaz lényegében ekvivalenssé válhat. E probléma kezelésére születtek az úgynevezett redukált rendű modellezési eljárások (ROM<sup>40</sup>).

**HotSpot.** A 2.5. ábrán látható struktúrához hasonló kompakt modellt is tartalmaz a HotSpot termikus szimulátor [33, 34]. Ennek a modellnek a számításigénye jelentősen kisebb, mint a szimulátor FDM módszerrel diszkrétizált „grid” modellje, így a futásidő kedvezőbb lehet ugyanazon struktúra szimulációja esetén. Hátránya, hogy a lapkán elhelyezkedő alkatrészek erősen befolyásolják a szimuláció pontosságát, nagy oldalarányú téglalapok rontják azt [50, 51].

**FANTASTIC.** L. Codecasa és munkatársai *multipoint moment method* nevű eljárást használnak termikus modellek rendjének csökkentésére [52, 53]. Az általuk fejlesztett FANTASTIC programmal kis számításigényű, több hőforrásos termikus rendszerek modelljét lehet előállítani akár SPICE netlista formátumban [54].

**SUNRED.** A SUNRED termikus motor a véges térfogatok módszerét alkalmazva oldja meg a termikus hatásokat leíró differenciálegyenletet. Németh Márton kollégám ezt a termikus motort egészítette ki 3D IC-k mikrosatornás hűtésének leírására szolgáló kompakt modellel [55], amely az én munkámhoz is csatlakozik [S1, S2, S3].

**TERMODEL.** Kompakt modellek előállításához használható a THERMODEL nevű program [56, 57]. A NID<sup>41</sup> módszer a frekvencia- vagy időtartománybeli szimulációból vagy mérésből származó ugrásválaszból dekonvolúciós eljárást alkalmazva határozza meg a struktúra időállandó-spektrumát. A diszkrétizált időállandó spektrumnak közvetlenül megfeleltethető egy Foster-típusú termikus RC hálózat, amely egy ekvivalens átalakítással úgynevezett Cauer-típusú RC létrahálózattá konvertálható [58]. Ezen létrahálózat elemei megfeleltethetők a hővezetési út egyes fizikai szakaszainak. Termikus szimulációs feladatokra 3-6 fokozatú hálózatok már kielégítő pontosságot szolgáltatnak. Az időállandó spektrum 100-200 spektrumvonalat eredményező diszkrétizációja alapján meghatározott Cauer hálózat egy speciális reprezentációja az úgynevezett struktúra függvény, amely létező tokstruktúrák roncsolásmentes vizsgálatát teszi lehetővé [59].

Kompakt termikus modellezés legnagyobb problémája az áramkör részletes geometriai leírással adott fizikai környezetére vonatkozó termikus kompakt modell szisztematikus előállítása [60]. Redukált rendű modellek alkalmazása esetén a pontosság kritikus lehet, mert nem létezik becslés az eredmény hibájának felső korlátjára [61].

---

<sup>40</sup> ROM, *Reduced Order Modeling*

<sup>41</sup> NID, *Network Identification by Deconvolution*

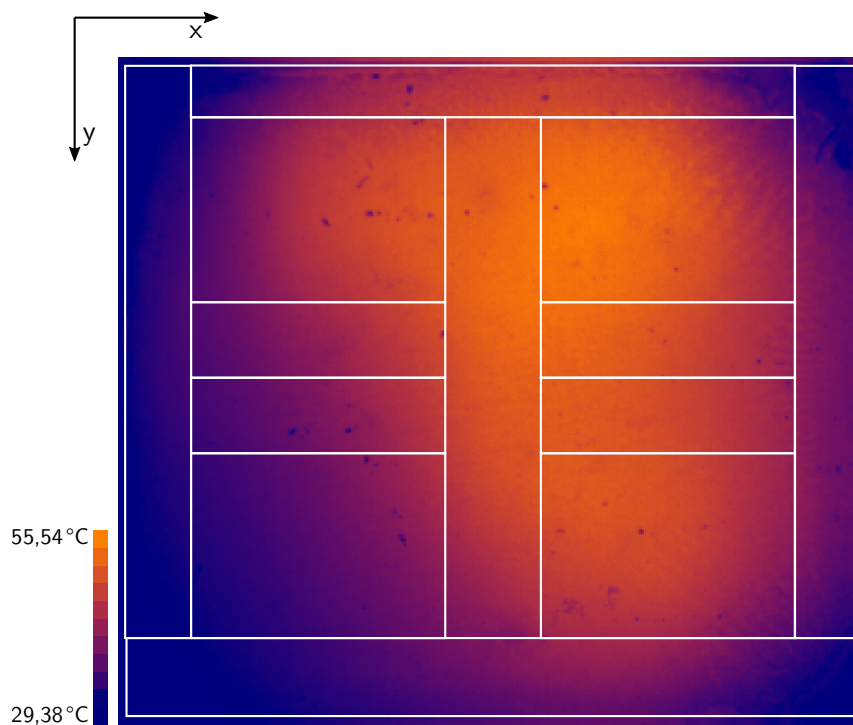
---

## 2.3. Termikus mérési módszerek

A fejlesztési folyamat összetettsége és a gyártási költségek miatt minimalizálni kell a prototípusokon végzett méréseket (*first silicon success*). Bizonyos esetekben azonban szükségessé válhat a kész áramkör termikus viselkedésének mérésel történő meghatározása, ami összetett IC-k esetén különösen nehéz feladat.

### Hőkamerás mérések

Komplex digitális IC-k a nagy disszipáció miatt megfelelő hűtés nélkül azonnal tönkremennek. Emiatt a tokozatlan lapka hőmérsékleti viszonyai nem mérhetőek, tok és a hűtőborda alkalmazása esetén pedig a mérőeszköz nem fér hozzá a chiphez. Erre a problémára egy megoldás az áramkör olajba merítése a mérés folyamán. Az olaj áramoltatásával elérhető a szükséges hűtési teljesítmény úgy, hogy közben az áramkör hőmérséklete hőkamerával mérhető marad. Hőkamerás mérést több kutatócsoport is végzett [62, 63], a 2.6. ábrán egy AMD Athlon II X4 610e processzor felületén kialakuló hőmérsékleteloszlás látható három mag terhelése esetén.



2.6. ábra. Modern négymagos processzor felületén kialakuló hőmérsékleti viszonyok működés közben [63]

Megjegyzendő azonban, hogy például az olajba merítés következtében az IC lapka felületi hőmérsékleteloszlása jelentősen különbözik az alkalmazási körülményekre jellemző eloszlástól, ezért az ilyen jellegű mérési eredmények nem vehetőek össze közvetlenül az áramkör normál működése közben kialakuló viszonyokkal. Emiatt a hőkamerás méréseknek és a termikus szimulációnak eltérő szerepe van az integrált áramkörök tervezésében [64].

### Termoreflektanciás mérések

A termoreflektanciás mérési eljárás a vizsgált anyagok által visszavert fény analízisén alapul. Az anyagok által reflektált fény mennyisége függ a fény hullámhosszától és az anyag hőmérsékletétől

is. Ismert hullámhosszú fénnel megvilágított minta esetén mérhető az anyag reflektivitásának változása, és így közvetetten a hőmérséklet is. A módszerrel lehetőség nyílik szubmikronos felbontású hőmérsékleteloszlások mérésére is.

A. Shakouri professzor és csoportja a UCSC-en majd a Purdue Egyetemen hőtérképező módszert fejlesztett ki ezen az elven [65], amelyet később a Microsanj<sup>TM</sup> cég kereskedelmi forgalomba hozott [66]. Ugyancsak ezen az elven alapuló mérési eljárást valósított meg P. E. Raad professzor az SMU-n [67, 68], illetve W. Claeys professzor és munkatársai a bordeaux-i egyetemen [69].

### Folyadékkristályos mérések

A folyadékkristályok hőmérséklet hatására állapotváltozást szenvednek, amely megváltoztatja egyes alapvető tulajdonságaikat. A hőtérképező eljárások a folyadékkristályok polarizációs tulajdonságának változását használják ki. A módszert Székely Vladimír professzor dolgozta ki az Elektronikus Eszközök Tanszékén [70, 71], majd Csender Alpár fejlesztette tovább PhD munkája során [72].

## 2.4. Funkcionális-termikus együttes szimuláció

A mérnöki gyakorlatban a funkcionális tervezés – chip és NYHL szinten egyaránt – villamosmérnöki feladat. A termikus tervezést ettől elkülönülten jellemzően gépészmérnökök végzik, így a két tervezési folyamat között kevés helyen történhet információcsere. Ez részben annak is köszönhető, hogy nem állnak a tervezők rendelkezésére olyan szimulációs módszerek és eszközök, amelyek hatékonyan támogatják az integrált áramkörök tranzisztorszintnél magasabb absztrakciós szintjén történő termikus-funkcionális együttes tervezést.

Az analóg áramkörök a tranzisztorok hőmérsékletfüggő karakterisztikái következtében érzékenyek a hőmérsékletre. Ezt bizonyos esetekben – például hőmérséklet-érzékelő szenzor esetén – kihasználják, egyéb esetekben pedig a tervezés során igyekeznek kiküszöbölni annak hatását a kapcsolás és a layout optimalizálásával. Elektro-termikus szimuláció során az áramkör elektromos és termikus viselkedését egy szimulációs környezetben modellezzük, így vizsgálni lehet a két tartomány egymásra hatását.

Kezdetben a digitális áramkörök a kvantált jelszintek és a kis fogyasztás miatt érzéketlenek voltak a hőmérsékletre, azonban a disszipációsűrűség növekedése (3. oldal, 1.1. ábra) végül az ilyen típusú IC-k esetén is termikus problémákat okozott és okoz ma is. Digitális áramkörök esetén a funkcionális viselkedést egyre erőteljesebben befolyásolják a termikus viszonyok, így napjainkban már ezen áramkörök funkcionális és termikus együttes szimulációjára is szükség lehet a tervezési folyamat során.

Értekezésem ezért az integrált áramkörök funkcionális és termikus viselkedésének együttes vizsgálatával foglalkozik. A vizsgált tartomány a chipre és a tokozásra korlátozódik. Ennek ellenére bizonyos helyzetekben szükségessé válhat a konvekcióval történő hőszállítás modellezése is, például mikrocsonnás hűtési megoldás esetén.

### 2.4.1. Az együttes szimuláció módszerei

Két vagy több tartomány együttes szimulációjának megvalósítására két bevett megoldás létezik, a szimultán iteráció (más néven direkt módszer), és a szimulátorcsatolás (vagy relaxált módszer).

---

## Szimultán iteráció

Szimultán iteráció esetén a termikus és elektromos tartománybeli viselkedést egy egyesített modell jellemzi, így a megoldás a szimuláció minden lépésében pontos, önkonzisztens lesz. Egy ilyen egyesített modellben az elektromos működést a hagyományos villamos hálózati elemek, a termikus viselkedést pedig kompakt modellezés során meghatározott hőellenállások és kapacitások reprezentálják. A hőmérsékletre érzékeny elektromos áramköri elemek – diódák, tranzisztorok – létesítenek kapcsolatot a két tartomány között. Az egyesített modell minden csomópontjára felírható Kirchoff első törvénye:

$$\begin{aligned} I(\bar{V}, \bar{T}) &= 0 \\ P(\bar{V}, \bar{T}) &= 0, \end{aligned} \tag{2.6}$$

ahol  $I$  az elektromos csomópontokra,  $P$  pedig a termikus csomópontokra felírt egyenletrendszer.  $\bar{V}$  és  $\bar{T}$  a keresett csomóponti potenciálok és hőmérsékletek vektora [73]. A szimultán iteráció a nem lineáris (2.6) egyenlet megoldását egyszerre – szimultán – állítja elő a  $[\bar{V}, \bar{T}]$  vektor minden elemére. A megoldó algoritmus alkalmazhatja a Newton-Raphson módszert a megoldás iteratív előállításához:

$$\mathbf{J}_{\mathbf{F}}(\bar{x}_n)(\bar{x}_{n+1} - \bar{x}_n) = -\mathbf{F}(\bar{x}_n), \tag{2.7}$$

amelyben  $\mathbf{F}$  a (2.6) egyenletrendszer mátrixa,  $\mathbf{J}_{\mathbf{F}}$  annak Jakobi-mátrixa,  $\bar{x}_n$  pedig a keresett változók vektora az  $n$ -edik iteráció után. Ennek a módszernek az előnye, hogy a szimuláció során számított változók – csomóponti potenciálok és hőmérsékletek – mindig konzisztensek maradnak az egyesített modell alkalmazása miatt, így szoros csatolású, időben gyors lefolyású elektromos-termikus hatások is vizsgálhatóak.

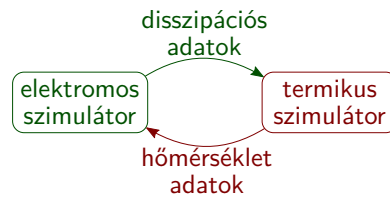
A szimultán iteráció hátránya, hogy mivel áramköri egyenletek felírására van szükség az elektromos oldalon, nincs lehetőség magasabb elvonatkoztatású modelleket alkalmazni az áramköri szintű leírás helyett. Emiatt ez az eljárás nagy méretű és összetett rendszerek – például VLSI digitális áramkörök – elektromos-termikus vizsgálatára csak korlátozottan alkalmas.

## Szimulátorcsatolás

Szimulátorcsatolás során a különböző tartománybeli szimulátor motorok egy közös ciklusban kapcsolódnak. Ennek a ciklusnak egy iterációja során az egyes tartománybeli szimulátor motorok meghívódnak, elvégzik az adott szimulációs lépéshez tartozó számításokat, majd az eredményeket továbbítják a többi szimulátornak. A 2.7. ábra mutatja be az elektromos-termikus szimulációt megvalósító relaxációs módszert. Az elektromos megoldó meghatározza a csomópontok potenciálját az egy lépéssel korábbi hőmérséklet adatok alapján, majd a disszipáció meghatározását követően a termikus motor kiszámítja az új hőmérsékleti értékeket. Az új értékeket a közös szimulációs ciklus következő lépésében használja csak fel az elektromos megoldó.

A szimulátorcsatolás lehetővé teszi, hogy két, önállóan is használható szimulátor programot egy közös ciklusba szervezve a két tartomány egymásra hatása is vizsgálható legyen, azonban ez a módszer nem minden esetben működik megfelelően. Mivel az elektromos és a termikus tartománybeli változókat – csomóponti potenciálokat, csomópontok hőmérsékletét – külön szimulátor határozza meg, nem garantált, hogy azok konzisztensek lesznek, például egy szimulációs hurokban kiszámított hőmérséklet-változás nem lesz hatással az ugyanabban az iterációban már meghatározott csomóponti potenciálokra. Más szavakkal fogalmazva a csatolt rendszer Jakobi mátrixában





2.7. ábra. Szimulátorcsatolás

az ellenkező tartománybeli parciális deriváltak ( $\partial I_m / \partial T_j$  és  $\partial P_n / \partial V_i$ ) nem szerepelnek (lásd a (2.7) egyenletet).

A szimulátorcsatolás előnye, hogy az adott tartománybeli szimulátor motort ki lehet cserélni egy másik, kedvezőbb paraméterekkel rendelkező programra. Elektromos-termikus szimuláció esetén például az egyik áramkör-megoldó programot ki lehet cserélni egy másikra, mindössze a csatoláshoz használt interfészhez kell illeszteni az új szimulátort. Ez a rugalmasság lehetővé teszi az elektromos-termikus együttes vizsgálat más elvonatkoztatási szinteken történő megvalósítását. A termikus szimulátornak a fogyasztási adatokra (2.1.2. fejezet), illetve egy disszipáció eloszlásra van szüksége, a funkcionális szimulátornak pedig a szimulált blokkok hőmérsékleti adataira.

## 2.4.2. Együttes szimulációt megvalósító rendszerek

### Elektro-termikus szimulátorok

Az elektro-termikus szimulátorok a legalacsonyabb elvonatkoztatási szinten valósítják meg az áramkör funkcionalitásának és termikus viselkedésének együttes szimulációját. Egyedül ezen a szinten van lehetőség a szimultán iteráció alkalmazására.

Az Elektronikus Eszközök Tanszékén fejlesztett TRANZ-TRAN a világon az elsők között rendelkezett elektro-termikus szimulációs képességekkel [74, 75, 76]. Később megszületett a program IC tervezőrendszerbe integrált változata (SISSI<sup>42</sup>), amely az integrált áramkör layoutja alapján állította elő a fizikai struktúrához tartozó termikus modellt [77, 78]. Más korabeli elektro-termikus szimulátorok is a szimultán iterációs eljárást alkalmazták az együttes szimuláció megvalósításához [79, 80].

Napjainkban az IC tervezőrendszereket fejlesztő cégek is kínálnak elektro-termikus szimulátorokat. A Mentor® ELDO® programja szimultán iterációs eljárásra támaszkodik [81, 82], a © Keysight HeatWave terméke pedig szimulátorcsatolást alkalmaz [83].

### Logi-termikus szimulátorok

Az úgynevezett logi-termikus szimulációs eljárás a logikai kapuk szintjén realizálja a funkcionális és termikus együttes szimulációt. A stacionárius logi-termikus szimulációs eljárás helyességét – az áramköri lapka felületi hőmérsékleteloszlása tekintetében – a tanszéki kutatócsoport teszttáramkörön végzett mérésekkel is igazolta [73].

Timár András a logi-termikus szimulációs módszert EDA környezetben implementálta [84]. A megvalósított CellTherm nevű keretrendszer az integrált áramkör tranziens szimulációja során a hőmérséklet hatását is figyelembe vette a standard cellák késleltetésének modellezése során [8, 23]. A logi-termikus szimuláció helyességét áramköri szintű SPICE szimulációkkal validálta [85].

<sup>42</sup> SISSI, *Simulator for Integrated Structures by Simultaneous Iteration*

Nagy Gergely a módszert továbbfejlesztette és egy saját fejlesztésű logikai szimulátor motort illesztett a SUNRED termikus motorhoz [86, 87]. Az egyedi logikai motor kapusintú leírásoknál magasabb elvonatkoztatású modelleket is tudott fogadni [88].

### Funkcionális-termikus szimulátorok

Az RTL-nél magasabb elvonatkoztatás az értekezésemben használt terminológiában az algoritmusok absztrakciós szintje, az ezen a szinten megvalósított funkcionális és termikus viselkedés együttes vizsgálatára alkalmas szimulátorokat funkcionális-termikus szimulátoroknak nevezem<sup>43</sup>.

T. Bouhadiba és kollégái SystemC-TLM<sup>44</sup> modellek funkcionális-termikus szimulációját egy SYNCHRO nevű komponenssel tették lehetővé [89]. Az általuk használt eljárás során a disszipáció-modellt a funkcionális TLM modell, az UPF<sup>45</sup> specifikáció és a konkrét hardveren megvalósított mérések alapján hozták létre [90], termikus szimulációkhoz a ATMI<sup>46</sup>-t használták [91].

S. Vinco és munkatársai különböző tartományok együttes szimulációját valósították meg SystemC felhasználásával [92]. A tartományon belüli és azok közötti kommunikációt a modellezéshez használt SystemC és SystemC-AMS<sup>47</sup> nyelvi elemeivel valósították meg. A keretrendszer SystemC-ben és SystemC-AMS-ben implementált funkcionális modellt tud fogadni, a disszipáció-, termikus és megbízhatósági modelleket SystemC-AMS-ben valósították meg.

A Ctherm [12] funkcionális-termikus szimulátor az SoCLib IP könyvtár [93] komponseit használta funkcionális modellként, a termikus szimulációkhoz a HotSpot-ot alkalmazták. Ehhez hasonlóan az SST<sup>48</sup>-t is kibővítették a disszipációt, hőmérsékletet és megbízhatóságot modellező komponensekkel [36]. A Ctherm és az SST funkcionális modellje kötött, nem képes tetszőleges áramköri modell funkcionális-termikus szimulációjára. Ezeknek a keretrendszereknek a célja nem az áramkörök tervezéséhez használt eszköztár bővítése, hanem különböző algoritmusok kiértékelése a fogyasztás/disszipáció és hőmérséklet szempontjából.

### 2.4.3. Az eddigi megoldások korlátai

A 2.4.2. fejezetben bemutatott megoldások egyes szempontok alapján kötöttek, nem általános módszerek az együttes szimuláció megvalósítására.

#### Fix elvonatkoztatási szint

A fenti megoldások az alkalmazási területüket tekintve nem fedik le az IC-k tervezésének egész spektrumát, jellemzően egy elvonatkoztatási szintet támogatnak, legyen az áramköri, kapusintú vagy algoritmus szint. Ez alól kivétel Nagy Gergely megoldása, amely az egyedi logikai motor megvalósításával tetszőleges elvonatkoztatási szintű áramköri modellt fogadhat, azonban ennek hátránya, hogy a logikai motorhoz újra kell implementálni a szimulálandó modellt.

---

<sup>43</sup> Nyelvi korlátok miatt az együttes szimulációs módszert és az algoritmusok szintjén megvalósított szimulátorokat ugyanúgy nevezem. Törekedtem arra, hogy a dolgozatban szöveggörnyezet alapján meg lehessen őket különböztetni.

<sup>44</sup> TLM, *Transaction Level Modeling*

<sup>45</sup> UPF *Unified Power Format*

<sup>46</sup> ATMI, *Analytical model of Temperature in Microprocessors*

<sup>47</sup> AMS, *Analog Mixed Signal*

<sup>48</sup> SST, *Structural Simulation Toolkit*

---

### **Fix modellező nyelv**

A korábbi megoldások általában egy-egy modellező nyelvet támogatnak. Ez szerkezeti jellegű leírások esetén nem hátrány. Például áramköri szintű elektro-termikus szimulátorok, és a kapusintű logi-termikus szimulátorok esetén az elsődleges szempont, hogy az adott megoldás a tervezési folyamatban alkalmazott EDA eszközök által előállított netlistát fogadni tudja. Az általános együttes szimulációs módszerrel szemben azonban követelmény, hogy magasabb elvonatkoztatási szinten is együttműködjön a létező EDA infrastruktúrával, így az RTL és algoritmus szinten használt modellező nyelveket is támogassa. T. Bouhadiba és S. Vinco megoldásai a tartományok összekapcsolásához a SystemC és SystemC-AMS modellező nyelvek eszközeit használják. Ez kötöttséget jelent, ha a módszerüket más magas elvonatkoztatású modellező nyelv – például SystemVerilog – támogatásával szeretnék bővíteni.

### **Fix funkcionális modell**

Bizonyos megoldások – például Ctherm, SST – fix, magas elvonatkoztatású komponenskönyvtárral és processzormodellekkel dolgoznak, így nem tekinthetőek általános együttes szimulációs módszernek. Ezeknek a megoldásoknak a célja elsősorban különböző DTM algoritmusok és a modellen futtatott szoftver viselkedésének kiértékelése, nem az integrált áramkörök tervezési folyamatának közvetlen bővítése.

## 3. fejezet

# Példamodellek és felhasznált eszközök

### 3.1. A dolgozat demonstrációs modelljei

A dolgozat későbbi fejezeteiben bemutatott módszer általános célú, így tetszőleges integrált áramkörüi modellre alkalmazható. A téziseimet alátámasztó eredményeimet három, különböző elvonatkoztatási szinteken implementált processzoros rendszeren keresztül mutatom be. A processzorok előnye a speciális céláramkörökkel szemben az, hogy az architektúrájuk heterogén, azaz tartalmaznak aritmetikai egységeket, regisztereket és regisztertömböket, multiplexereket, a végrehajtás vezérléséhez szükséges állapotgépeket, így ezekkel jól demonstrálhatóak az általam létrehozott szimulációs keretrendszer tulajdonságai. Ez a fejezet rövid áttekintést nyújt a felhasznált demonstrációs modellekről az összetettségük szerinti növekvő sorrendben. A demonstrációs processzorokra görög betűkkel hivatkozom,  $\alpha$  a kis,  $\beta$  a közepes és  $\gamma$  a nagy összetettségű modell.

#### Az $\alpha$ példaprocesszor

Ez a tanszéki fejlesztésű processzormag egyszerű, kilenc utasítást tartalmazó, háromcímes utasításkészlettel, 8 bites nem pipeline-osított adatúttal és egy 16 elemű általános célú regisztertömbbel rendelkezik. A dolgozatban használt modellt SystemC-ben regiszter-átviteli szinten implementáltam Horváth Péter modellje alapján, a kapusintű reprezentációját ugyancsak SystemC nyelven valósítottam meg. A kapusintű modell körülbelül ezer cellából áll, a kapukhoz tartozó fogyasztási és terület információkat a tanszéken rendelkezésre álló, az AMS<sup>1</sup> 0,35  $\mu\text{m}$ -es CMOS technológiájához készített PDK-ból nyertem ki. A szimulációk során a processzor egész számok osztását hajtotta végre.

#### A $\beta$ példaprocesszor

A második processzor ugyancsak Horváth Péter fejlesztése [7], azonban jelentősen összetettebb, mint az előző modell. A  $\beta$  processzor 64 utasítása között megtalálhatóak háromcímes és kétcímes utasítások is. Előbbiek a processzor 32 bites egész-aritmetikás ALU<sup>2</sup> egységén, utóbbiak a fixpontos 64 bites DSP<sup>3</sup> ALU-n hajtódnak végre. Az utasítások végrehajtása egyszeres kibocsajtású, ötfokozatú pipelineban történik. A pipeline-os felépítésből adódó hazárdok hatását elágazásbecslő és adatvisszacsatoló egységek csökkentik. A processzor 128 elemű általános célú regisztertömbbel rendelkezik, amelyhez hozzáférés egy ablakozó mechanizmuson keresztül, egy időben 32 regiszterhez

---

<sup>1</sup> AMS, *AustriaMicroSystems*

<sup>2</sup> ALU, *Arithmetic Logic Unit*

<sup>3</sup> DSP, *Digital Signal Processor*

lehetséges. A kétcímes fixpontos utasítások eredménye egy 64 bites akkumulátorba kerül, ahonnan külön utasítással lehet az adatot a regisztertömbbe visszaírni.

Horváth Péter VHDL-ben implementált modelljét SystemC-ben is megvalósítottam, a kapusintű leírást az AMS 0,35  $\mu\text{m}$ -es CMOS technológiájára történő RTL szintézissel állítottam elő. A kapusintű leírás hozzávetőleg 25 ezer standard cellából áll.

## A $\gamma$ példaprocesszor

A dolgozatban használt harmadik processzormodell a Sniper x86 processzorszimulátor [94], amelyet az Intel<sup>®</sup> Xeon<sup>®</sup> X5550 paraméterei alapján konfiguráltam. A szimulátor a processzor disszipációját a McPAT modell felhasználásával becsüli meg, amelyet a szimulátor szerzői az említett processzorral felszerelt szerveren végrehajtott mérésekkel validáltak [13].

A szimulátor alkalmas x86 utasításkészletre fordított bináris programok végrehajtására, a legismertebb benchmark csomagokat pedig integráltan tartalmazza (SPLASH-2<sup>4</sup> [95], SPEC<sup>5</sup> CPU2006 [96], PARSEC<sup>6</sup> 2.1 [97]). A  $\gamma$  processzor szimulációihoz a SPLASH-2 csomag FFT<sup>7</sup> programját használtam.

## 3.2. A felhasznált tervező szoftver eszközök

Munkám során a hagyományos integrált áramkörök tervezési folyamatában is alkalmazott tervező eszközöket használtam: különböző modellezési nyelveket, szimulátorokat és további IC tervezői eszközöket, amelyek mind szabványos megoldások.

### Modellező nyelvek, szimulátorok

Példaáramkörök funkcionális modellezéséhez SystemC, SystemC-AMS könyvtárakat és Verilog nyelvet, szimulációhoz SystemC-t, a Mentor<sup>®</sup> Questa<sup>™</sup> Simet és a Cadence<sup>®</sup> IES-t, analóg szimulációkhoz Cadence<sup>®</sup> Virtuoso<sup>®</sup>-t használtam. A fentiekén kívül felhasználtam a magas elvonatkoztatású Sniper x86 utasításkészletű processzorszimulátort is az új együttes szimulációs eljárás rugalmas alkalmazhatóságának bemutatására.

### Szintézis programok, PDK

Az RTL és fizikai szintézist a Cadence<sup>®</sup> RTL Compiler illetve Encounter programjával végeztem, megvalósítási (*target*) technológiaként az AMS 0,35  $\mu\text{m}$ -es CMOS standard cellakönyvtárat megadva. A LEF<sup>8</sup>/DEF<sup>9</sup> adatbázisokkal specifikált layout beolvasásához, valamint a PDK-ban található Liberty<sup>™</sup> adatbázis feldolgozásához nyílt forráskódú szoftvereket vettem igénybe.

### Termikus szimulátorok

Az áramkör funkcionális modellezésére használt standard eszközökkel szemben az egyes termikus szimulációs megoldások csak sokkal szűkebb körben elterjedtek. Az IC tervező cégek az

---

<sup>4</sup> SPLASH, *Stanford Parallel Applications for Shared Memory*

<sup>5</sup> SPEC, *Standard Performance Evaluation Corporation*

<sup>6</sup> PARSEC, *Princeton Application Repository for Shared-Memory Computers*

<sup>7</sup> FFT, *Fast Fourier Transform*

<sup>8</sup> LEF, *Library Exchange Format*

<sup>9</sup> DEF, *Design Exchange Format*

---

igényeiknek legmegfelelőbb CAD eszközöket használják a termikus viselkedés modellezésére, valamint akadémiai kutatások is foglalkoznak az IC lapkák termikus szimulációjának problémakörével. Utóbbi kategóriába sorolhatóak az általam is használt szimulátorok, a szukcesszív hálózatredukciós algoritmusra épülő SUNRED, valamint a nyílt forráskódú, szabadon elérhető 3D-ICE és HotSpot. Megvalósítottam egy minta termikus szimulátor motort is, amihez a szabadon is hozzáférhető PARALUTION lineáris algebra könyvtárat [98] használtam fel.

### **Programozási nyelvek**

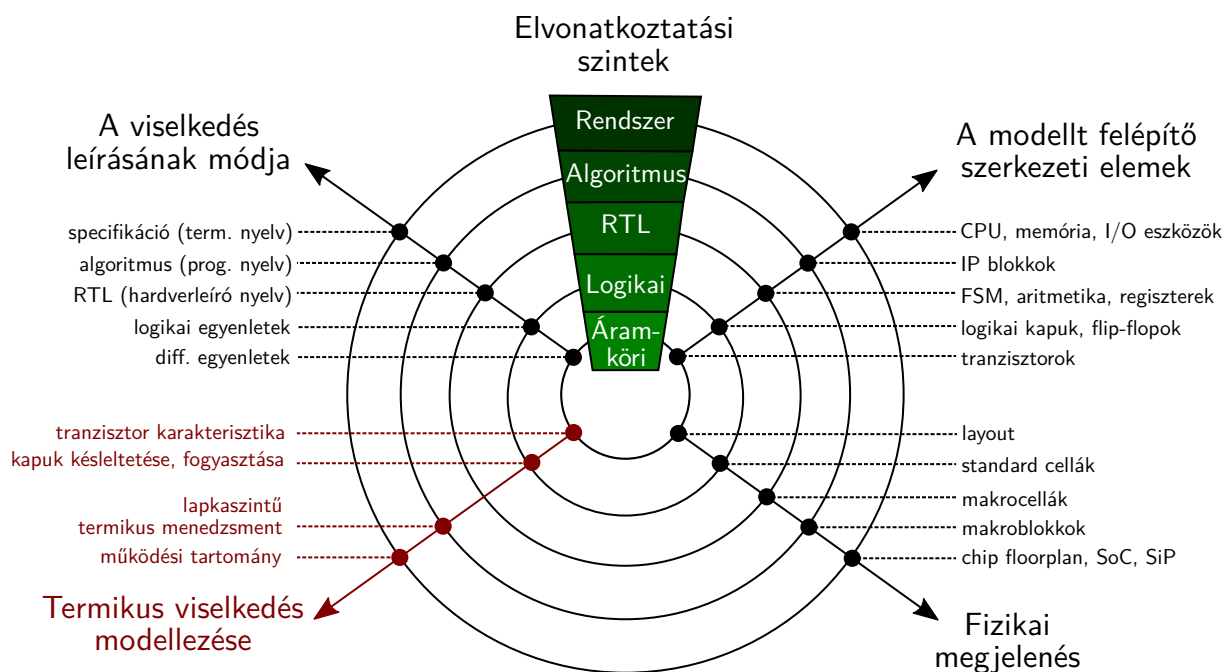
A kutatási eredmények alapján megvalósított keretrendszer implementálásához C++ és Python programozási nyelveket használtam, a szimulációs eredmények feldolgozásához és szemléltetéséhez az R nyelvet vettem igénybe.

## 4. fejezet

# A szimulációs tartományok egyesítése

Kutatásom célja egy olyan általános szimulációs módszer kidolgozása volt, amely független az integrált áramkörök tervezési folyamatában alkalmazott modellező nyelvektől és absztrakciós szintektől. Így tetszőleges elvonatkoztatású és nyelvű áramköri modell funkcionális-termikus viselkedése vizsgálhatóvá válik.

### 4.1. Termikus viselkedés modellezése



4.1. ábra. Termikus viselkedés modellezésével bővített GK diagram

A 4.1. ábrán a Gajski-Kuhn diagram módosított változata látható. A negyedik ágon az egyes elvonatkoztatási szinteken modellezhető hőmérséklet által befolyásolt működési jellemzőket mutatom be. Magasabb elvonatkoztatási szinteken rendszerszintű tulajdonságok vizsgálhatóak, például a hőmérsékleti működési tartomány vagy a lapkaszintű termikus menedzsment algoritmusok működése.

A hőmérsékleti működési tartomány a specifikáció alapján határozható meg. Például amíg a kereskedelmi célú processzorokat 0–85 °C hőmérsékleti tartományra tervezik, a katonai célú eszkö-

zöknek a  $-55-125\text{ }^\circ\text{C}$  hőmérsékleteken is működőképesnek kell lenniük, úreszközök esetén pedig akár specifikusan egy adott misszió paramétereit figyelembe véve határozzák meg ezt a követelményt. Termikus menedzsment algoritmusok célja, hogy az integrált áramkör működése közben a lapka hőmérséklete ne haladja meg a helyes működéshez szükséges határértékeket. Napjainkban egyre kifinomultabb algoritmusokat implementálnak a processzorgyártók, hogy megakadályozzák a túlmelegedést, és közben növeljék a számítási teljesítményt részterhelés – kevés programszálal alkalmazó programok – esetén.

Alacsonyabb elvonatkoztatási szinteken már több információ áll rendelkezésre az áramkör tényleges fizikai implementációjáról, így lehetőség nyílik a fizikai paraméterek – például a kapuk késleltetésének és fogyasztásának, vagy a tranzistorok karakterisztikáinak – hőmérsékletfüggő modellezésére is.

A célkitűzések – tetszőleges elvonatkoztatású leírás és modellező nyelv támogatása – miatt az együttes szimulációs módszerek közül kizárólag a relaxációs módszer/szimulátorcsatlós alkalmazható az új eljárás megvalósításához. A funkcionális és termikus tartományok összekapcsolásához, az áramköri modell két tartománybeli viselkedésének együttes szimulációjához egymáshoz kell rendelni a modell fizikai megjelenésének egységeit és a viselkedési leírás egyes részeit.

## 4.2. Absztrakt interfész kidolgozása

A funkcionális ( $F$ ) és termikus ( $T$ ) szimulációs tartományok összekapcsolásához bevezetem a *funkcionálismodell-komponens* ( $F_i$ ), a *disszipációforrás* ( $p_i$ ) és *layout alakzat* ( $L_i$ ) fogalmakat.

**Funkcionálismodell-komponens.** A funkcionális modellt  $n$  darab diszjunkt részhalmoz, a funkcionálismodell-komponensek alkotják. Egy ilyen funkcionálismodell-komponens ( $F_i$ ) hőmérsékletét annak  $T_i$  paramétere jellemzi.  $F_i$  tulajdonságai formálisan a (4.1) definíció szerint foglалhatóak össze.

$$\begin{aligned} F &= \{F_1 \dots F_n\} \\ \forall (i, j), i \neq j \quad F_i \cap F_j &= \emptyset \\ \forall F_i \quad F_i &\mapsto T_i \end{aligned} \tag{4.1}$$

**Disszipációforrás.** A  $p_i$  disszipációforrások tulajdonságai formálisan a (4.2) definíció szerint foglалhatóak össze. A disszipációforrások a  $P$  disszipációmodell elemei. Minden  $F_i$  funkcionális komponenshez pontosan egy disszipációforrás rendelhető, amely annak a komponensnek a teljesítményfelvételét becsli.

$$\begin{aligned} P &= \{p_1 \dots p_n\} \\ \forall F_i \subset F \quad F_i &\mapsto p_i \text{ ahol } p_i \in P \end{aligned} \tag{4.2}$$

**Layout alakzat.** A layout része a  $T$  termikus modellnek. A layoutot alkotó  $l_{ij}$  téglalapok  $\{x, y, z\}_p$ <sup>1</sup> pozícióval és  $\{x, y\}_a$ <sup>2</sup> mérettel rendelkeznek, a fizikai struktúrában elfoglalt pozíció és méret alapján a layout primitívek hőmérséklete meghatározható. Az  $L_i$  layout alakzat  $m$  darab ilyen téglalap uniója<sup>3</sup>, amely pontosan egy  $F_i$  funkcionálismodell-komponenshez rendelhető.

<sup>1</sup>  $\{x, y, z\}_p$  pozíciót jellemző számhármass

<sup>2</sup>  $\{x, y\}_a$  a felületet jellemző számpár

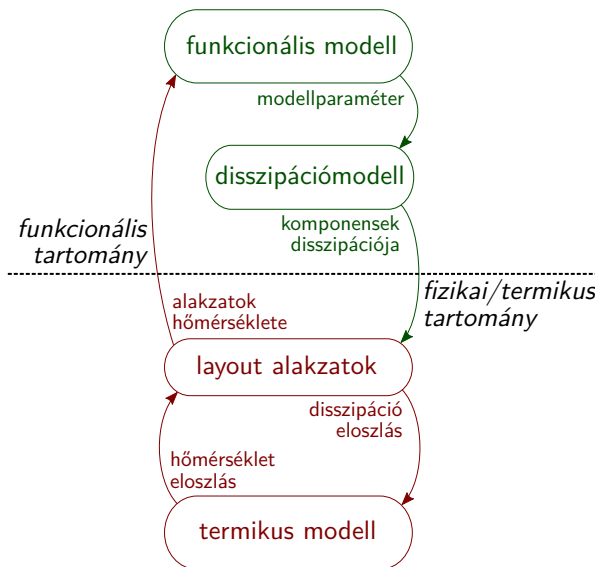
<sup>3</sup> Ezáltal több téglalap tarozhat egy adott funkcionális komponenshez, ami lehetővé teszi összetettebb layout alakzatok létrehozását is.



A layout alakzatok páronként diszjunktak. Ezeket a tulajdonságokat foglalja össze formálisan a (4.3) definíció.

$$\begin{aligned}
 \{l_{11} \dots l_{nm}\} &= L \in T \\
 \forall l_{ij} \quad l_{ij} &\mapsto \{x, y, z\}_p, \{x, y\}_a \\
 L_i &= \bigcup_{j=1}^m l_{ij} \subset L \\
 \forall (i, j), i \neq j \quad L_i \cap L_j &= \emptyset \\
 \forall F_i \subset F \quad F_i &\mapsto L_i \text{ ahol } L_i \subset L
 \end{aligned} \tag{4.3}$$

### Az interfész működése



4.2. ábra. Az interfész által megvalósított szimulátorcsatolás

A két tartomány összekapcsolásának célja a funkcionális modell hőmérsékleti információval való bővítése, lehetővé téve a hőmérsékletfüggő viselkedés modellezését. A 4.2. ábrán látható az együttes szimuláció relaxációs módszerrel való megvalósítása a bemutatott fogalomrendszer felhasználásával. A szimulátorcsatolás ciklusának egy iterációjában a funkcionális modell működése alapján a disszipációmodell kiértékelődik. Ez az információ átadódik a funkcionális modell komponenseihez tartozó  $L_i$  layout alakzatoknak. A layoutot alkotó alakzatoknak ismert a méretük és a pozíciójuk, így a termikus modell a disszipáció alapján meg tudja határozni minden  $L_i$  hőmérsékletét<sup>4</sup>. Ezt a hőmérsékletet adja vissza a termikus modell a funkcionális modell komponenseinek.

Különböző elvonatkoztatási szinteken a disszipációmodell és layout reprezentációja eltérő lehet. Magas elvonatkoztatási szinten a fogyasztást és a területigényt inkább becsülni lehet, mint pontosan

<sup>4</sup> Mivel a téglalapok rendelkeznek mérettel és pozícióval, így meghatározható a hőmérsékletük. A primitívek uniójával alkotott alakzat hőmérséklete többféleképpen képezhető, például téglalapok felületének arányában vett súlyozott átlagolással, a hőmérsékleti értékek mediánjával, vagy az alakzat középpontjának hőmérsékletével.

modellezni<sup>5</sup>. A fejlesztés előrehaladtával egyre részletesebb információ áll rendelkezésre az áramkörtől, így a disszipációmodell és a layout is finomítható. Ezeket a modelleket hagyományos tervezési folyamatban is gyakran előállítják a fejlesztés egyéb résztvevői és a tervezés későbbi fázisai számára.

**I. Tézis.** *Absztrakt interfészt dolgoztam ki az integrált áramkörök funkcionális és termikus tartománybeli modelljeinek összekapcsolásához. Az új interfész felhasználható szimulátorcsatlózással megvalósított funkcionális-termikus együttes szimulációs eljáráshoz, és tetszőleges nyelven implementált, tetszőleges elvonatkoztatású modellel dolgozó szimulátor programhoz illeszthető [S1, S2, S3, S4, S5, S6].*

### 4.3. Összehasonlítás a korábbi megoldásokkal

A 4.2. fejezetben bemutatott absztrakt interfész fogalmi absztrakciós szinttől és modellező nyelvtől függetlenek, de megfeleltethetők a 4.1. ábrán szereplő tetszőleges elvonatkoztatási szintű modell elemeinek. Az interfésszel a 2.4.2. fejezetben említett tetszőleges funkcionális-termikus együttes szimulációs módszer megvalósítható, a 4.1. táblázat foglalja össze az egyes fogalmak jelentését különböző szimulációs módszerek esetén.

Szimulációs módszer	Az interfész elemei
Elektro-termikus	$F_i$ áramkörtörök elemek (tranzisztorok, ellenállások stb.) $p_i$ hálózatgyenletek megoldásából számítható $L_i$ áramkörtörök elemek layout rajzolata
Logi-termikus	$F_i$ logikai kapuk/standard cellák $p_i$ cellakönyvtár fogyasztási adatai $L_i$ a kapuk layout rajzolatai
Funkcionális-termikus	$F_i$ összetett funkciójú elemek, IP blokkok $p_i$ disszipációt becslő modell (McPAT, Wattch stb.) $L_i$ makroblokkok becsült mérete és elrendezése

4.1. táblázat. Különböző absztrakciós szinteken alkalmazott együttes szimulációs módszerek megvalósítása a bemutatott interfésszel

A módszer valós alkalmazásához erre alkalmas programozási nyelven implementálni kell az interfész elemeit és integrálni kell a használni kívánt szimulátorokba. A 7. fejezetben a szimulációs módszer egy lehetséges megvalósítását is bemutatom.

<sup>5</sup> Magas elvonatkoztatási szinten a funkcionális modell tetszőleges programozási nyelven megvalósítható. A disszipációmodell és a layout ilyen esetekben a specifikáció és korábbi ismeretek alapján becsülhető és az alkalmazott programozási nyelv felhasználásával a viselkedési leíráshoz illeszthető.

## 5. fejezet

# A tervezési folyamat bővítése

A 4.3. fejezetben bemutatottam, hogy az I. tézisben megfogalmazott interfésszel megvalósíthatóak a korábban elvonatkoztatási szintek szerint elválasztott, különböző együttes szimulációs módszerek. A 9. oldalon a 2.2. ábrán látható tervezési folyamat bővítéséhez az interfészhez kell illeszteni a tervezés során alkalmazott szimulátorokat, formális nyelveket. A dolgozat fókuszja a digitális és kevert jelű (*mixed-signal*) SoC áramkörök tervezési folyamatának bővítése, így az ezeknél alkalmazott hardverleíró nyelvek támogatása volt az elsődleges cél.

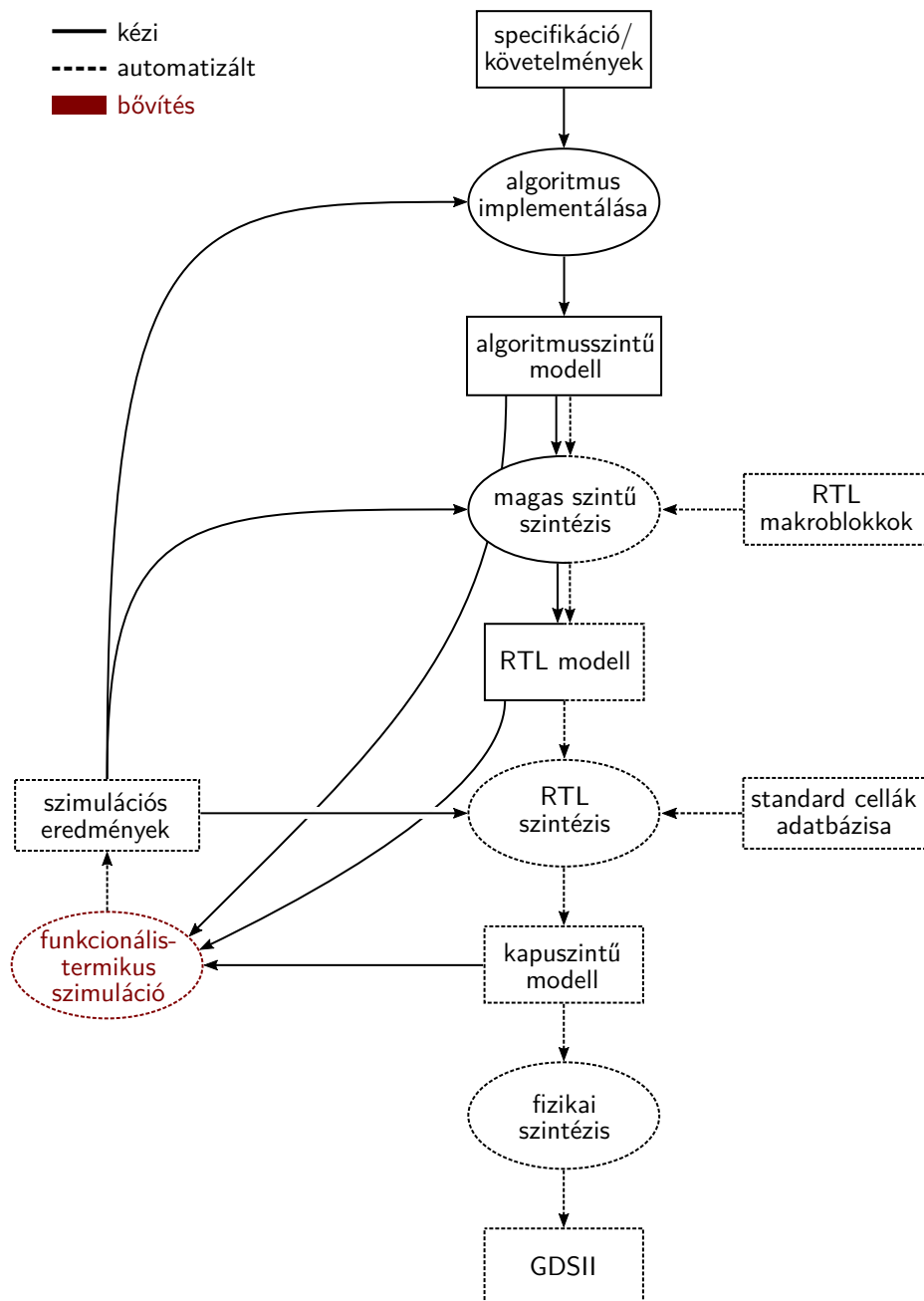
### 5.1. A tervezés során használt formális nyelvek

A 33. oldalon az 5.1. ábrán látható a funkcionális-termikus szimulációval bővített tervezési folyamat. Az I. tézisben leírt interfész gyakorlati alkalmazásához a  $p_i$  disszipációforrások, az  $L_i$  layout alakzatok és az  $F_i$  funkcionálismodell-komponensek reprezentációját biztosítani kell a tervezés több lépésében, különböző elvonatkoztatási szinteken. A tervezés során a funkcionális modellt jellemzően valamilyen modellező vagy hardverleíró nyelven implementálják. Az 5.1. táblázat a funkcionális modellt felépítő nyelvi eszközöket foglalja össze a legismertebb, általános célú hardverleíró nyelvek esetén. A funkcionális modellt hierarchiába szerveződő elemek alkotják, aminek a működését párhuzamosan kiértékelődő folytonos értékadások és szekvenciális utasításblokkok írják le.

Nyelvi elem	Verilog	VHDL	SystemC
hierarchiába szervezhető elem	module	entity	sc_module
szekvenciális utasításblokk	always, initial	process	sc_thread, sc_method
folytonos értékadás	assign	<=	-

5.1. táblázat. Funkcionális modellt felépítő nyelvi elemek ismert HDL-ekben

A modell felbontása/granularitása függ az alkalmazott elvonatkoztatási szinttől. Finom felbontású modellek nagy számú, önmagukban egyszerű felépítésű elemekből állnak, míg a durva granularitású modelleket kevesebb, de összetettebb hierarchikus elem alkotja. Magasabb absztrakciójú modellek esetén az összetett hierarchikus elemeket kódolási stílus függvényében alkothatják rövidebb vagy hosszabb szekvenciális utasításblokkok is. A hardverleíró nyelvekben a szekvenciális utasításblokkok és folytonos értékadások kaphatnak egyedi azonosítót, azonban ezek általában nem



5.1. ábra. Funkcionális-termikus szimulációval bővített tervezési folyamat

kötelezőek. Emiatt az EDA eszközökben – például RTL/logikai szintézerekben – a funkcionális modell legkisebb, egyértelműen azonosítható alkotóelemei a hierarchiába szerveződő elemek.

A 4. fejezetben leírt módszer általános jellegének megőrzése céljából az  $F_i$  funkcionális modell komponenseként a HDL-ek hierarchiába szerveződő elemét választottam. Az ehhez tartozó  $p_i$  disszipációforrások és  $L_i$  layout alakzatok hagyományos EDA eszközök – magas szintű, RTL/logikai és fizikai szintézer programok – illetve a szakirodalomban ismert disszipációmokellek (2.1.2. fejezet) implementálásával állíthatóak elő.

## Logikai szint

Logikai elvonatkoztatási szint esetén az új szimulációs módszer megfelel a logi-termikus szimulációs eljárásnak (31. oldal, 4.1. táblázat). A standard cellákat tartalmazó leírásban minden egyes logikai kapu a funkcionális modell egy diszjunkt komponense ( $F_i$ ), amihez egy disszipációforrás ( $p_i$ ) és egy layout alakzat ( $L_i$ ) rendelhető. A disszipációforrás a logikai szimulációs motorban regisztrált kapcsolási aktivitás, és a standard cellákat tartalmazó adatbázis energia modellje alapján adja a termikus motor számára a gerjesztést. A layout alakzatok méretét és pozícióját a fizikai szintézer-programból kinyerhető layout leíró fájlok tartalmazzák. A kapuszintű logi-termikus szimulációt az  $\alpha$  processzor felhasználásával mutatom be.

## Regiszterátviteli szint

Regiszterátviteli szinten nem áll rendelkezésre közvetlen információ az áramkör fizikai megvalósításáról. Az általam javasolt automatizálható módszer a disszipációforrások paramétereinek meghatározásához egy előzetes szintézis eredményét, valamint a kapott hálózat kényszerített véletlen gerjesztésével kinyert disszipáció-értékeit használja. Az RTL szintű funkcionális modell szintézisét követően egyértelműen azonosítható, hogy az egyes standard cellák a generált kapuszintű modellben az eredeti funkcionális modell melyik komponenséhez tartoznak<sup>1</sup>. A layout alakzatok méretét és helyzetét ugyancsak az előzetes szintézis eredményeit felhasználva lehet meghatározni, mivel az RTL/logikai szintézerek a felhasznált cellák alapján meg tudják becsülni az eredeti komponensek méretét. Az ilyen elvonatkoztatású funkcionális-termikus szimulációhoz szükséges adatok előállítását az  $\alpha$  és  $\beta$  processzormodelleken keresztül mutatom be a fejezet további részében.

## Algoritmusszint

Az algoritmusszintű leírások nem tartalmaznak információt a leendő áramkör szerkezetére vagy fizikai megvalósítására vonatkozóan, így ilyen elvonatkoztatású általános modellek funkcionális-termikus szimulációja körülményes. Magas szintű szintézerek rendelkezésre állása esetén az RTL leírásoknál alkalmazott módszer általános SystemC-TLM modellekre is alkalmazható. Speciális modellek – például processzor szimulátorok – esetén ismert lehet a disszipáció (2.1.2. fejezet) és a funkcionális komponensek elrendezése is. A  $\gamma$  modell esetén a Sniper szimulátor Intel® Xeon® X5550 processzor paramétereit alkalmaztam. A keretrendszer algoritmusszintű funkcionális-termikus szimulációs képességét ennek a processzor szimulátornak a felhasználásával mutatom be.

## 5.2. Szükséges többlet információ előállítása

Az előző pontban bemutatott bővített tervezési folyamatban a funkcionális modell mellett szükséges a layout és a disszipáció adott elvonatkoztatási szintjű modelljének előállításához<sup>2</sup>. Ezzel a problémakörrel a szakirodalomban fellelhető számos publikáció foglalkozott, így az azokban közölt eredményekre támaszkodom.

---

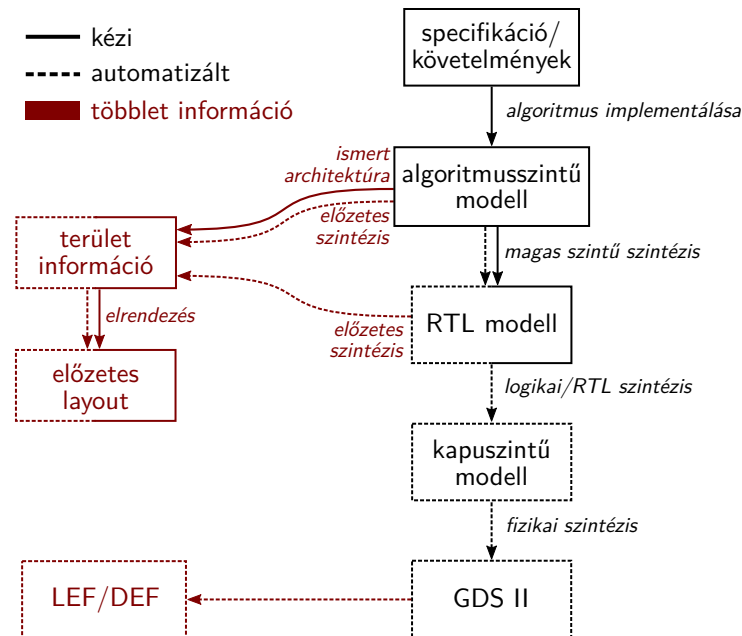
<sup>1</sup> Ezért volt szükség az  $F_i =$  hierarchiába szervezhető elem választásra

<sup>2</sup> A fejlesztés során korábban is szükséges volt különböző absztrakciós szinten a fogyasztást és területigényt az elérhető részletességgel modellezni. Ez az információ a folyamat egyéb résztvevői és a tervezés későbbi fázisainak bemeneteként szolgált.

---

### 5.2.1. Layout reprezentáció

Funkcionális-termikus szimuláció futtatásához az adott elvonatkoztatási szintű modell mellett elő kell állítani az áramköri komponensek elrendezését a leendő szilícium lapkán. Logikai kapukat tartalmazó leírás alapján lehetőség van a tényleges layout megtervezésére a fizikai szintézis során (5.2 ábra).



5.2. ábra. A layout reprezentáció előállítása különböző elvonatkoztatási szinteken

Magasabb elvonatkoztatási szintek – RTL vagy algoritmusszintű modellek – esetén az egyes komponensek területigényére és az elrendezésre becslés adható. RTL leírásokból előzetes RTL/logikai szintézissel kinyerhető az RTL modellt alkotó komponensek becsült területigénye, amelyek már felhasználhatóak előzetes elrendezést generáló algoritmusok, mint például a HotFloorplan [34] bemeneteként. Kiszámú alakzat esetén az előzetes elrendezés akár kézi úton is előállítható.

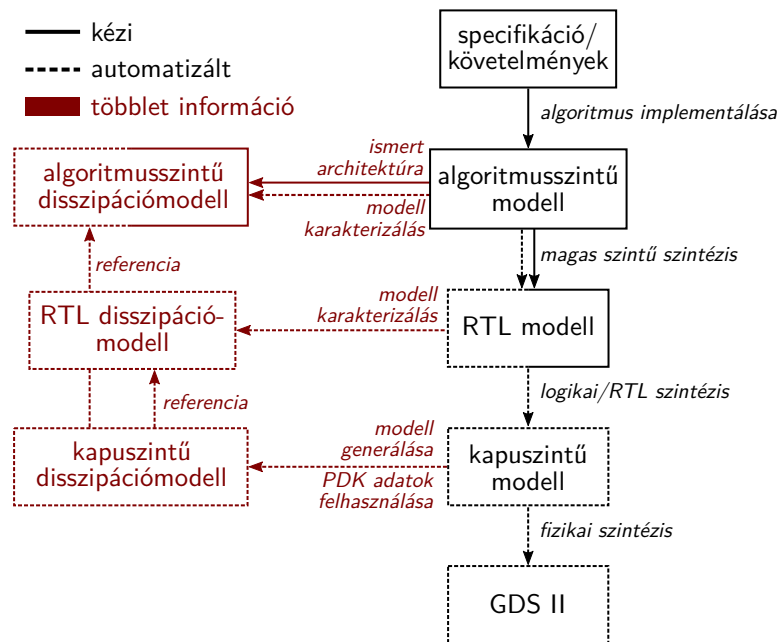
Algoritmusszintű modellek esetén, amennyiben rendelkezésre áll magas szintű szintézer eszköz, ugyanúgy lehet eljárni, mint RTL szintnél. A magas szintű szintézerrel automatizált módon állítható elő RTL és kapus szintű leírás, amikből már következtetéseket lehet levonni a leendő áramköri komponensek területigényére vonatkozóan. Amennyiben az alkalmazott tervezési folyamatban manuálisan áll elő az RTL modell, akkor az egyes komponensek területigényére vonatkozóan korábbi ismeretekre – már megvalósított áramkörök adataira – kell támaszkodni.

### 5.2.2. Disszipációmodellek előállítása

Funkcionális-termikus együttes szimuláció futtatásához az  $L_i$  alakzatok területigényének becslése mellett modellezni kell az  $F_i$  komponensek disszipációját is. Az alkalmazott absztrakciós szint függvényében különböző disszipációmodellekre lehet szükség, ami adott időpillanatokban megbecsüli a fogyasztást.

A disszipáció becslésére a szakirodalmi összefoglaló 2.1.2. fejezetében bemutatott fogyasztási modellek mind alkalmasak. Standard cellákat tartalmazó logikai szintű leírás esetén a fogyasztás számítására felhasználhatóak a PDK adatai, vagy ennek hiányában előállíthatóak SPICE szimulációk sorozatával, hasonlóan a Timár András PhD disszertációjában bemutatott módszerhez [8].

Regiszterátviteli szinten alkalmazhatóak a 2.1.2. fejezetben bemutatott disszipáció-makromodellek, algoritmus szinten pedig a makromodellek mellett használhatóak az integrált McPAT, Wattch vagy CACTI keretrendszerek is. Az 5.3 ábrán látható a disszipációforrások előállításának folyamata.



5.3. ábra. A disszipációmodellek előállítása különböző elvonatkoztatási szinteken

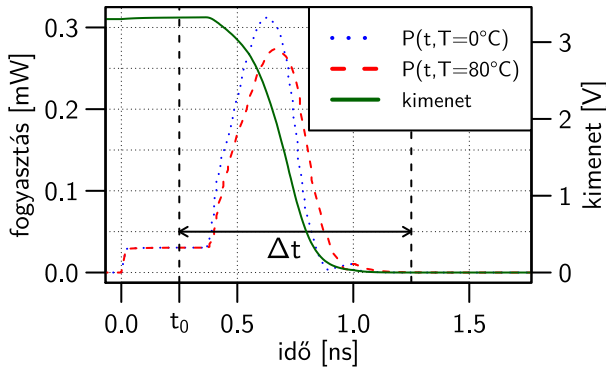
### Disszipáció modellezése logikai szinten

Kapuszintű leírás esetén használható a standard cellákat tartalmazó adatbázis fogyasztásra vonatkozó információja. Ezt az adatbázist a félvezetőgyártó állítja elő és a PDK-ban teszi elérhetővé IC tervezők számára. Minden cellához tartozik egy energiamodell, amely különböző paraméterek függvényében képes modellezni a kérdéses kapu disszipációját. A Liberty<sup>TM</sup> fájlformátum és különböző teljesítmény modellek részletei Timár András értekezésében megtalálhatóak [8]. A dolgozatban bemutatott módszer a logikai kapuk absztrakciós szintjén azonosnak tekinthető az általa megvalósított szimulátor keretrendszerrel. A szabványos EDA eszközöket használó CellTherm logi-termikus szimulátorhoz hasonlóan az általam megvalósított rendszer is fel tudja dolgozni a Liberty<sup>TM</sup> adatbázist a cellák fogyasztási adatainak kinyerése érdekében.

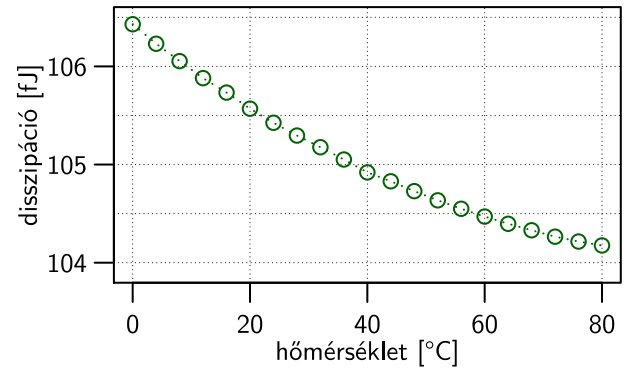
Kapuszintű disszipációmodelleket sorozatos SPICE szimulációkkal is meg lehet határozni [8, 23]. Az 5.4(a) ábrán egy NAND20 cella fogyasztása látható 0 °C és 80 °C hőmérséklet esetén a kimenet 1–0 váltásakor. Az (5.1) egyenlet alapján meghatároztam a  $\Delta t$  idő alatt lezajló jelváltás során disszipált energiát a hőmérséklet függvényében, amelynek eredménye az 5.4(b) ábrán látható. A dinamikus fogyasztás 2%-ot változik 0–80 °C tartományon, így a modellezés során ezt a hőmérsékletfüggést elhanyagoltam.

$$W(T) = \int_{t_0}^{t_0 + \Delta t} P(t, T) dt \quad (5.1)$$

Az 5.5. ábrán a NAND20 cella SPICE szimulációval meghatározott szivárgási árama látható a hőmérséklet függvényében. A szivárgási áram exponenciális összefüggésben áll a hőmérséklettel, így az ebből adódó statikus disszipáció is erősen hőmérsékletfüggő. Az AMS 0,35  $\mu\text{m}$  CMOS technológiájával megvalósított cellák statikus fogyasztása több nagyságrenddel kisebb a dinamikus

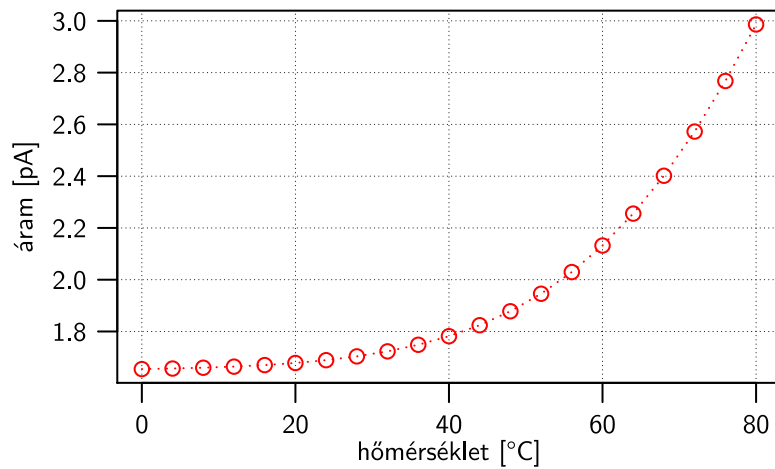


(a) A cella fogyasztása a kimenet 1-0 váltásakor



(b) A disszipált energia a hőmérséklet függvényében

5.4. ábra. NAND20 cella dinamikus fogyasztása a kimenet 1-0 váltásakor



5.5. ábra. NAND20 cella szivárgási árama

fogyasztásnál, így az erre a gyártástechnológiára tervezett áramkörök esetén a statikus fogyasztás elhanyagolható<sup>3</sup>.

### Disszipáció modellezése regiszterátviteli szinten

Jellemzően az automatizált szintézis eszközök bemenete a regiszterátviteli szinten megfogalmazott leírás [6], de újabban megjelentek algoritmusszintű modellekből RTL modellt előállító eszközök is. Ebben az esetben a szintézis eredménye kevésbé kézben tartható a tervezők részéről, mert a magas elvonatkoztatási szintű modellekben nehezebben fejezhető ki az áramkör architektúrája [7]. Emiatt digitális áramkörök implementációja jelenleg is főleg RTL-en történik.

Kódolási stílus és az áramkör szerkezetére vonatkozó információtartalom tekintetében még ezen az elvonatkoztatási szinten is van szórás. Viselkedési RTL modellnek azt a stílust nevezem, ahol a leírásban nincs szétválasztva a vezérlés és az adatút, illetve egy-egy tervezési egység több funkciót is ellát. Strukturális RTL modell az a kódolási stílus, amelynél a vezérlés és az adatút külön tervezési egységekbe van szétválasztva, az egyes tervezési egységeknek egyedi funkciójuk van.

<sup>3</sup> A C függelékben az AMS 0,35  $\mu\text{m}$ -es gyártástechnológia módosított változatán mutatom be a statikus disszipáció hatását.



Az utóbbi kódolási stílus kedvezőbb a funkcionális-termikus szimulációhoz, mert a szintézer programok részletesebb információt tudnak biztosítani az egyes tervezési egységek becsült terület-igényéről, ezért olyan lineáris regresszió alapuló disszipációmodellt alkalmaztam, ami elsősorban ilyen RTL leírásokhoz illeszthető [19].

Az áramkör statikus és dinamikus disszipációját lehetséges külön-külön modellezni. Digitális áramköröknél alkalmazott CMOS kapcsolástechnika esetén a statikus fogyasztás egyszerűen számítható az előzetesen szintetizált netlistát alkotó kapuk szivárgásból adódó disszipációjá<sup>4</sup> alapján:

$$P_{statikus} = \sum_{i=1}^N p_{si}, \quad (5.2)$$

ahol  $P_{statikus}$  a teljes áramkör statikus disszipációja,  $N$  a cellák száma és  $p_{si}$  az  $i$ . standard cella statikus disszipációja, amelyet a standard cellakönyvtárból lehet kinyerni<sup>5</sup>.

Az alkalmazott disszipáció-makromodellben az RTL modellek dinamikus fogyasztását egy többszörös lineáris regressziós modell becsüli meg:

$$\hat{P}_{dinamikus} = \sum_{i=1}^N \left( p_{i0} + \sum_{j=1}^{M_i} \alpha_{ij} \cdot p_{ij} \right), \quad (5.3)$$

amelyben  $\hat{P}_{dinamikus}$  a teljes áramkör becsült dinamikus fogyasztása,  $N$  az előzetesen szintetizált netlista celláinak száma,  $M_i$  az  $i$ . cella portjainak a száma,  $\alpha_{ij}$  az  $i$ . cella  $j$ . portjához tartozó kapcsolási aktivitás és  $p_{ix}$  a regressziós modell együtthatója.

A teljes fogyasztás a statikus és dinamikus fogyasztás összegeként számolható:

$$P_{teljes} = P_{statikus} + P_{dinamikus} = P_{statikus} + \hat{P}_{dinamikus} + \varepsilon, \quad (5.4)$$

ahol  $\varepsilon$  a dinamikus összetevő becslésének hibája.

A regressziós modell együtthatóinak meghatározása előzetes karakterizálással történik. Egy kapusintű modellre gerjesztést adva meghatározható annak fogyasztása és a portok aktivitása minden időpillanatban. Kellően nagyszámú mintával –  $k \gg M$ , ahol  $M$  a portok száma és  $k$  a minták száma – olyan együtthatók határozhatóak meg, amelyekkel a hiba minimális lehet [19].

## Disszipáció modellezése algoritmusszinten

A dolgozatban demonstrációs céllal használt  $\gamma$  processzor a Sniper szimulátor programhoz illesztett McPAT keretrendszer használja disszipáció-modellként. A McPAT az architektúra komponenseinek fogyasztását azok aktivitása alapján külön-külön becsli, az aktivitási adatokat a szimulátor program biztosítja. A  $\gamma$  modell esetén használt konfigurációt a szimulátor készítői valós processzoron végzett mérések alapján validálták [13].

Az algoritmusszintű modellekhez a disszipáció makromodelljét automatizáltan is elő lehet állítani magas szintű szintézerek segítségével (2.1.2. fejezet). A dolgozatban bemutatott módszer algoritmusszinten nem korlátozódik a speciális  $\gamma$  processzormodellre, általános algoritmusszintű leírások funkcionális-termikus szimulációjára ugyanúgy használható a megfelelő infrastruktúra rendelkezésre állása esetén.

<sup>4</sup> Ezt az információt tartalmazza a standard cellák Liberty™ adatbázisa.

<sup>5</sup> A statikus disszipáció hőmérséklet, tápfeszültség, bemeneti kombináció, és további paraméterek függvénye lehet. Ezeket nem tüntettem fel, de figyelembe vehetőek a disszipáció modellezésének ezen lépésében.

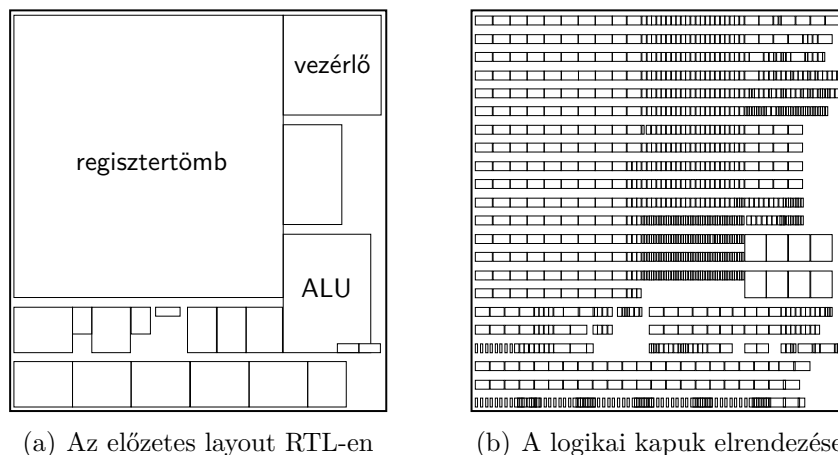
**II. Tézis.** *Eljárást dolgoztam ki az I. tézisben bemutatott interfészt alkalmazó funkcionális-termikus együttes szimulációs módszernek az integrált áramkörök tervezési folyamatába való illesztéséhez. A bővített eljárásban a disszipációforrások és layout alakzatok modellezéséhez a szakirodalomban ismertetett tetszőleges eljárás alkalmas, amely a hagyományos tervezési folyamatban alkalmazott formális nyelvekhez, illetve az alkalmazott egyedi funkcionális szimulátorhoz illeszthető. Az általam megvalósított eljárás alkalmazásával a fejlesztés több lépésében ugyanabban a szimulációs környezetben válik vizsgálhatóvá az áramkör termikus viselkedése. [S4, S6, S7].*

### 5.3. A példa processzorok disszipáció és layout modelljei

A funkcionális-termikus szimulációs módszer demonstrációjához megvalósítottam a 3.1. fejezetben leírt példaprocesszorok layoutját/floorplanjét, illetve disszipáció modelljét. Ezekre az ismeretekre egy áramkör tervezése során egyébként is szükség van.

#### 5.3.1. A megvalósított fizikai elrendezések

##### Az $\alpha$ processzor



5.6. ábra. Az  $\alpha$  processzor layoutja RTL és logikai szinteken

layout alakzat	terület
regisztertömb	0,155 mm <sup>2</sup>
vezérlő	0,02 mm <sup>2</sup>
ALU	0,021 mm <sup>2</sup>

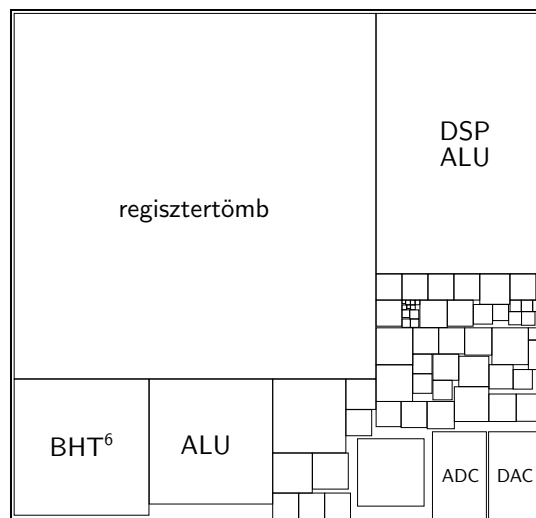
5.2. táblázat. Az  $\alpha$  processzor layout komponenseinek mérete

A kis összetettségű  $\alpha$  processzormag regiszter átviteli szintű komponenseinek elrendezése látható az 5.6(a) ábrán. Az egyes layout alakzatok területét a kapuszintű modellen elvégzett fizikai szintézis eredménye alapján határoztam meg, azaz ebben az esetben a rendelkezésre álló kész layout

után valósítottam meg az RTL komponensek elrendezését. Az 5.6(b) ábrán láthatóak a layouton elhelyezett standard cellák. Az 5.2. táblázat tartalmazza az 5.6. ábrán megnevezett alakzatok területét. A processzormag mérete ezzel az elrendezéssel  $0,525\text{ mm} \times 0,559\text{ mm}$ .

### A $\beta$ processzor

A közepes összetettségű  $\beta$  processzor kézzel tervezett előzetes layoutja látható az 5.7. ábrán. Az egyes layout alakzatok területe a szintézerből kinyert eredményeken alapul a kevert jelű ADC és DAC kivételével, amelyek területigényét létező ADC és DAC modulok mérete alapján [99] becsültem meg. Az 5.3. táblázat tartalmazza az 5.7. ábrán névvel feltüntetett alakzatok területét. A processzormag mérete ezzel az elrendezéssel  $2,46\text{ mm} \times 2,352\text{ mm}$ .



5.7. ábra. A  $\beta$  processzor előzetes layoutja RTL szinten

layout alakzat	terület
regisztertömb	$2,812\text{ mm}^2$
BHT	$0,391\text{ mm}^2$
ALU	$0,328\text{ mm}^2$
DSP ALU	$0,938\text{ mm}^2$
ADC	$0,1\text{ mm}^2$
DAC	$0,1\text{ mm}^2$

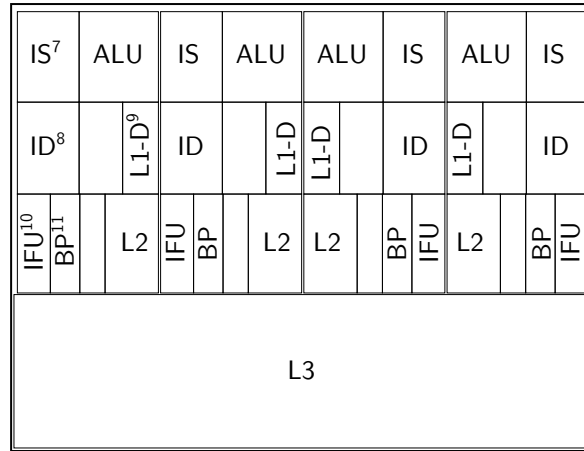
5.3. táblázat. A  $\beta$  processzor layout komponenseinek mérete

### A $\gamma$ processzor

A  $\gamma$  processzor layoutja látható az 5.8. ábrán. Mivel a Sniper szimulátornak része egy, az Intel® Xeon® X5550 processzor tulajdonságain alapuló konfiguráció, így egy olyan elrendezés megvalósítására törekedtem, amely hasonlít az említett processzor layoutjára [100]. Az 5.4. táblázat tartalmazza az 5.8. ábrán feltüntetett layout komponensek méretét. Ezzel az elrendezéssel a processzormag mérete  $17,2\text{ mm} \times 13,2\text{ mm}$ , tehát kismértékben eltér a valóságos lapka  $19,6\text{ mm} \times 13,5\text{ mm}$  méretétől.

<sup>6</sup> BHT, *Branch History Table*

A szimulátorban alkalmazott McPAT disszipációmodell a processzormagok belső komponenseire és a harmadik szintű gyorsítótár disszipációjára ad becslést, a lapka fotóján [100] szereplő I/O illetve memóriavezérlő egységekre nem, így ezeket nem tüntettem fel a layouton.



5.8. ábra. A  $\gamma$  processzor layoutja algoritmus/architektúra szintjén

layout alakzat	terület
IS	4,81 mm <sup>2</sup>
ALU	6,17 mm <sup>2</sup>
ID	6,17 mm <sup>2</sup>
L1-D	2,83 mm <sup>2</sup>
IFU	2,81 mm <sup>2</sup>
BP	2,51 mm <sup>2</sup>
L2	4,53 mm <sup>2</sup>
L3	76,5 mm <sup>2</sup>

5.4. táblázat. A  $\gamma$  processzor layout komponenseinek mérete

### 5.3.2. A felhasznált disszipáció modellek

#### Az $\alpha$ processzor

Az  $\alpha$  processzor RTL disszipációmodelljének létrehozásához szükséges karakterizálási folyamat számára a processzort alkotó komponensek kis száma miatt minden komponens típusához külön-külön tesztkönyvet valósítottam meg. A bemenetekre egyenletes eloszlású véletlen számokat adva szimuláltam a különböző típusok kapusintű modelljét, közben elmentve minden ki- és bemenet aktivitását, valamint a fogyasztást. Ezekből az adatokból egy R nyelven megvalósított szkript határozta meg az RTL disszipációmodell egyes együtthatóit. Ez a módszer hasonló a funkcionális verifikáció során alkalmazott úgynevezett *unit* tesztekhez, amely során a modellt

<sup>7</sup> IS, *Instruction Scheduler*

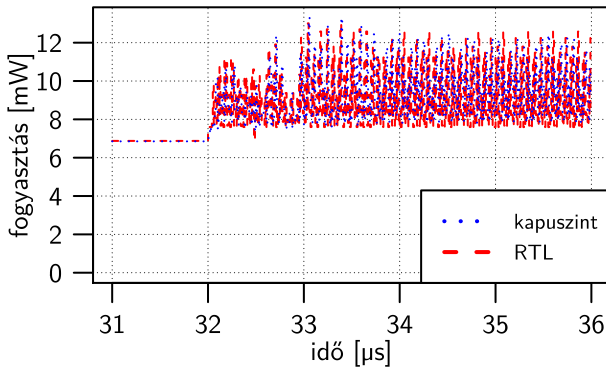
<sup>8</sup> ID, *Instruction Decode*

<sup>9</sup> L1-D, *L1 Data cache*

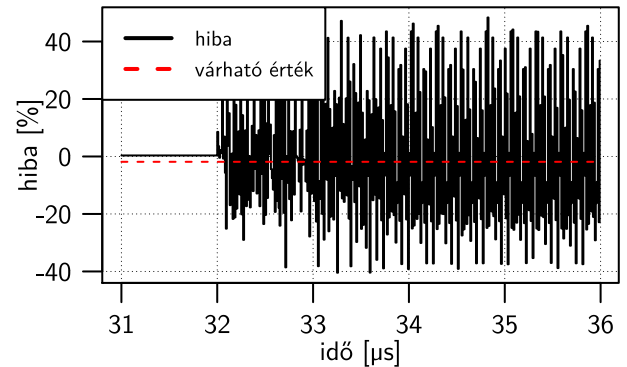
<sup>10</sup> IFU, *Instruction Fetch Unit*

<sup>11</sup> BP, *Branch Predictor*

alkotó egyes komponenseket külön-külön verifikálják kényszerített véletlenszerű (*constraint random*) gerjesztéssel.



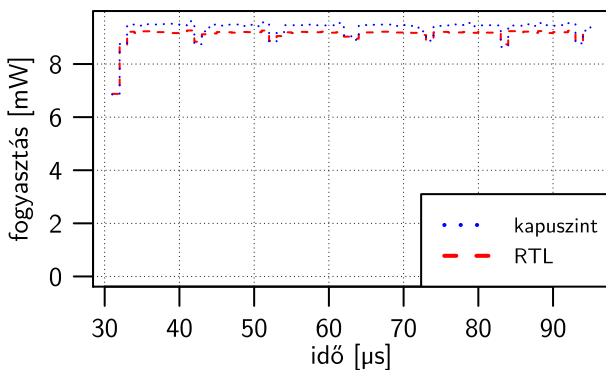
(a) Fogyasztás kapusztinten és a becslés RTL-en



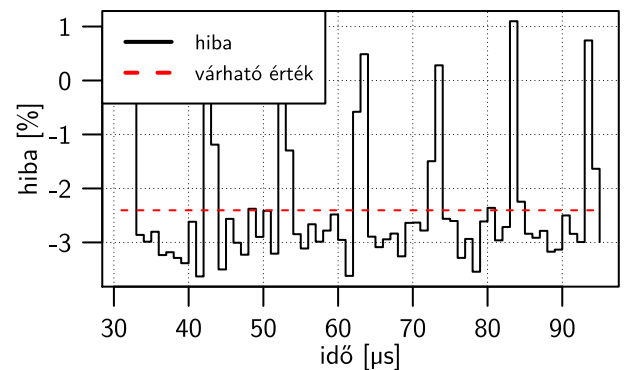
(b) A becslés hibája és annak várható értéke

5.9. ábra. Az  $\alpha$  processzor fogyasztásának becslése és annak hibája órajelenként

Az 5.9(a) ábrán a kis méretű  $\alpha$  processzormag disszipációja látható kapusztinten és a becsült fogyasztás RTL-en. Az 5.9(b) ábra a becslés hibáját mutatja be, ahol a referencia a kapusztintű modell esetén megfigyelt disszipáció. A pillanatnyi eltérés meghaladhatja a 40%-ot, azonban ez nem okoz ekkora eltérést a termikus szimuláció során, mert a termikus időállandók jelentősen nagyobbak a processzor órajelének periódusidejénél<sup>12</sup>.



(a) Fogyasztás kapusztinten és a becslés RTL-en



(b) A becslés hibája és annak várható értéke

5.10. ábra. Az  $\alpha$  processzor fogyasztásának becslése és annak hibája 1  $\mu$ s-onként kiértékelve

Az 5.10. ábra a becsült fogyasztást és annak hibáját mutatja be 1  $\mu$ s-os intervallumokban átlagolva. Láthatóan a hiba jelentősen csökkent, nem haladja meg a 4%-ot.

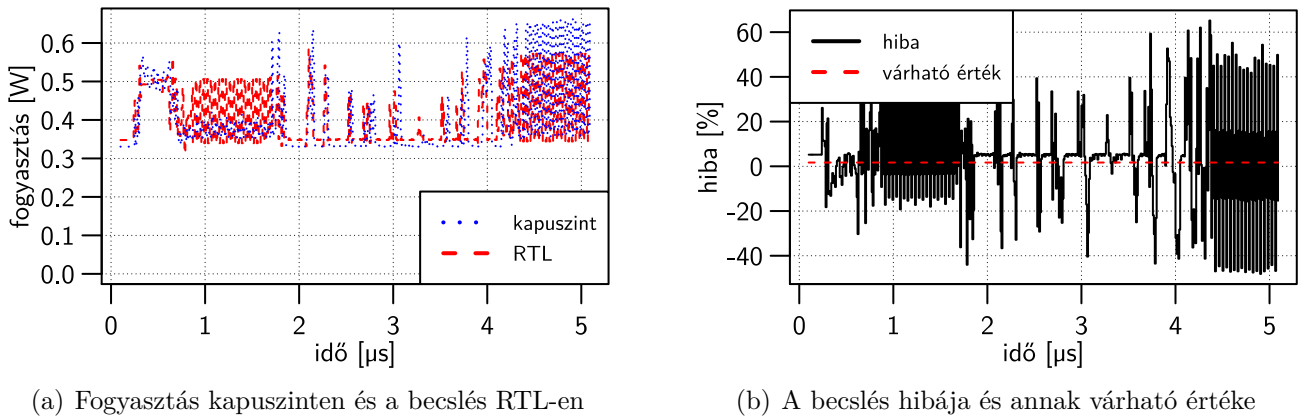
## A $\beta$ processzor

A közepes összetettségű  $\beta$  processzormag disszipációforrásainak karakterizálása nem lett volna lehetséges az  $\alpha$  processzor esetén alkalmazott módszerrel. Ebben a processzorban az utasítások egy öt fokozatú pipeline-ba kerülnek, amelynek vezérlése összetett, számtalan visszacsatolás érkezik az egyes pipeline fokozatokból, így ennek a komponensnek a disszipációmodelljét nem lehet egyszerű

<sup>12</sup> A processzorok órajelének periódusideje 0,2–10 ns tartományba esik, míg a termikus időállandók jellemzően 10  $\mu$ s–1 s nagyságrendbe esnek

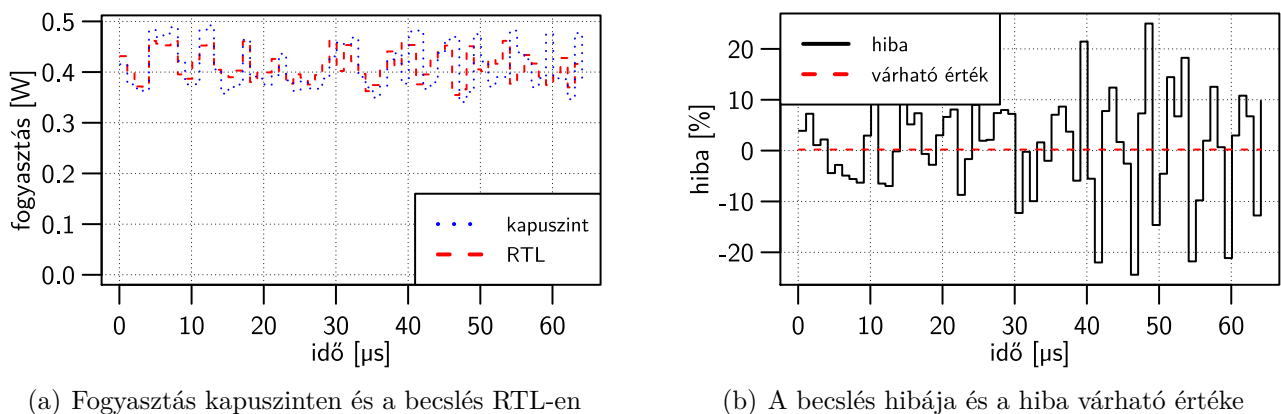
véletlen gerjesztéssel karakterizálni, mert ahhoz gyakorlatilag a teljes processzormagot meg kellett volna újra valósítani verifikációs komponensként.

Ezért ehhez a processzormodellhez egy olyan célprogramot készítettem, amely tetszőlegesen hosszú véletlen utasítássorozatot képes generálni. Ezt felhasználva egy szimuláció futtatásával lehet karakterizálni az összes komponens disszipáció forrását.



5.11. ábra. A  $\beta$  processzor fogyasztásának becslése és annak hibája órajelenként

Az 5.11(a) ábra a  $\beta$  processzormag disszipációját mutatja be kapuszenten és a becslést fogyasztást RTL-en. Az 5.11(b) ábrán a becslés hibája látható, ahol a referencia a kapuszentű modell esetén megfigyelt disszipáció. A pillanatnyi eltérés meghaladhatja a 60%-ot, azonban ez nem okoz ekkora eltérést a termikus szimuláció során<sup>13</sup>.



5.12. ábra. A  $\beta$  processzor fogyasztásának becslése és annak hibája 1  $\mu$ s-onként kiértékelve

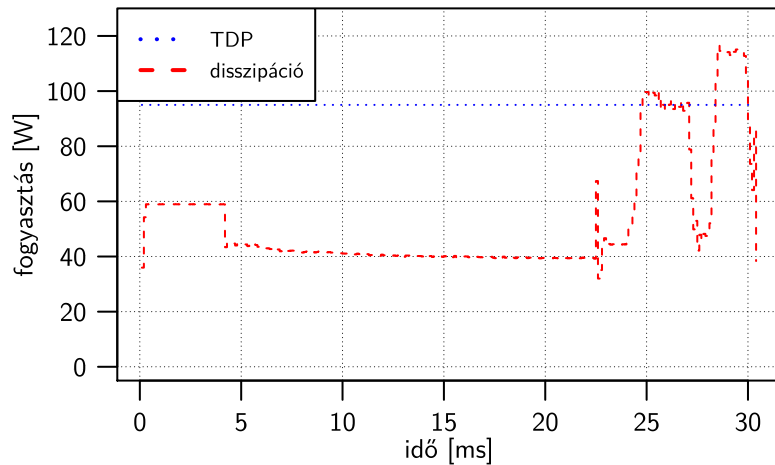
Az 5.12. ábra 1  $\mu$ s-os intervallumokban átlagolva mutatja be a közepes méretű  $\beta$  processzor becslést fogyasztását és annak hibáját. Láthatóan a hiba nagyobb, mint az előző, kis összetettségű processzor esetén. Ennek egyik oka, hogy a processzor disszipációmódeljének karakterizálása véletlen utasítássorozattal kedvezőtlenebb a komponensenkénti karakterizálásnál, a gyorsabb, kevesebb munkával járó módszer pontatlanabb is. A másik ok az áramkörök összetettségében keresendő, a  $\beta$  processzor körülbelül 25 ezer logikai kapuból áll, míg a kis processzormag csak ezből. A  $\beta$  processzor egyes komponensei – ALU, DSP ALU, regisztertömb stb. – jóval több kapuból állnak,

<sup>13</sup> A processzorok órajelének periódusideje 0,2–10 ns tartományba esik, míg a termikus időállandók jellemzően 10  $\mu$ s–1 s nagyságrendbe esnek

ezeknél a komponenseknél a disszipációmodellben alkalmazott feltevés, hogy a portok kapcsolási aktivitása jól tükrözi a fogyasztást, már nem egészen fedí a valóságot.

### A $\gamma$ processzor

A nagy összetettségű  $\gamma$  processzormag fogyasztásának becsléséhez a szimulátor tervezői a McPAT [11] modellt karakterizálták [13]. Mivel ebben az esetben nem állt rendelkezésemre kapusintű illetve RTL implementáció, ezért a becslés pontosságát csak a hivatalos TDP értékéhez tudtam hasonlítani.



5.13. ábra. A  $\gamma$  processzor hivatalos TDP értéke és McPAT modellel becsült fogyasztása

Az 5.13. ábrán látható egy rövid tesztprogram futtatása során becsült fogyasztás, és a hivatalos TDP érték. Láthatóan a becsült disszipáció bizonyos helyzetekben meghaladja a TDP-t, ami nem feltétlen jelent pontatlan becslést. A TDP definíció szerint az a hőmennyiség, aminek az elvezetésére méretezni kell a lapkát hűtő megoldást, azonban a pillanatnyi fogyasztás meghaladhatja ezt.

## 6. fejezet

# Új termikus motor megvalósítása

A 4.2. fejezetben bemutatott funkcionális-termikus interfész működéséhez szükséges egy termikus szimulátor, amely a disszipációeloszlás alapján meghatározza a hőmérsékleteloszlást. Az új SloTh termikus szimulációs motor tervezése és megvalósítása során prioritás volt a gyors működés és az integrálhatóság a funkcionális-termikus együttes szimulációs rendszerembe.

### 6.1. Tokozás szintű termikus modellezés FVM-mel

Az új termikus szimulátor tokozás szintű termikus szimulációkhoz készült. Ezen a szinten a cél a lapka hőmérsékleteloszlásának meghatározása, a hőút tokozáson kívül eső része nem képezi a vizsgálat tárgyát, így jellemzően a hőt elszállító megoldások a tok vagy lapka által behatárolt szimulációs tartomány peremfeltételeként jelennek meg. A szimulátorban – mérlegelve a különböző diszkretizálási eljárások előnyeit – a véges térfogatok módszerét alkalmaztam. Ez a fejezet röviden bemutatja az FVM-en alapuló termikus modellezés egyes lépéseit.

#### 6.1.1. Szimulációs tartomány diszkretizálása

A 14. oldalon olvasható (2.4) egyenletet a  $V$  térfogatú diszkretizációs testre<sup>1</sup> integrálva kapható a térfogategységre alkalmazott energiamegmaradási törvény:

$$\int_V c_v \frac{\partial T}{\partial t} dV = \int_V \operatorname{div}(\lambda \cdot \operatorname{grad}(T)) dV + \int_V p dV, \quad (6.1)$$

amelyben  $c_v$  az anyag térfogategységre eső hőkapacitása,  $\lambda$  a hővezetés,  $T$  a hőmérséklet,  $p$  a disszipációsűrűség és  $t$  az idő. A Gauss-tétel alapján a divergencia térfogati integrálja átalakítható a térfogatot bezáró zárt  $A$  felszínre történő integrálással:

$$\int_V c_v \frac{\partial T}{\partial t} dV = \oint_A \lambda \cdot \operatorname{grad}(T) dA + \int_V p dV. \quad (6.2)$$

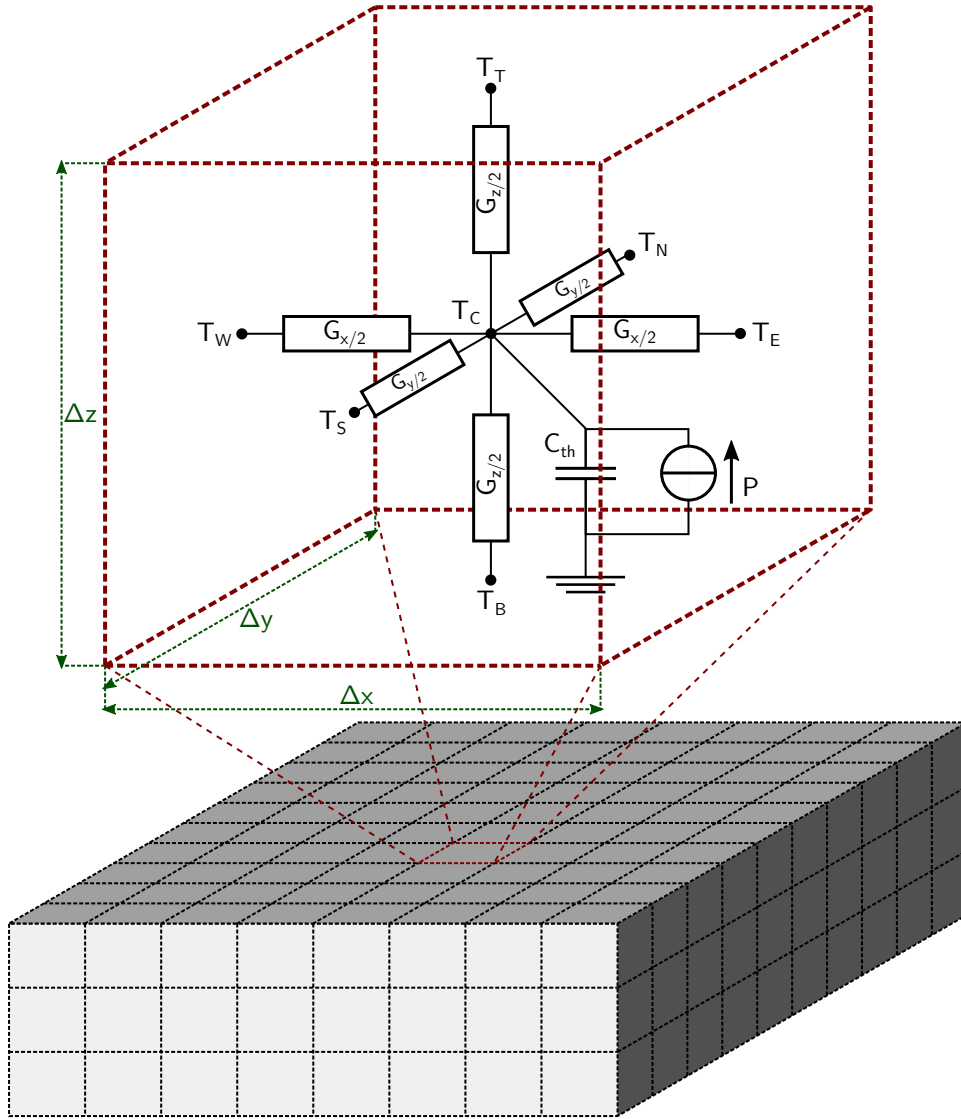
Az előző egyenletet Descartes koordináta-rendszerben kifejezve és a térfogatban felszabaduló  $p$  Joule-hő sűrűséget átrendezve jutunk a következő alakhoz:

$$\int_V c_v \frac{\partial T}{\partial t} - p dV = \oint_A \left( \lambda \frac{\partial T}{\partial x} dydz + \lambda \frac{\partial T}{\partial y} dxdz + \lambda \frac{\partial T}{\partial z} dxdy \right). \quad (6.3)$$

---

<sup>1</sup> A továbbiakban szinonimaként alkalmazom a kontrolltérfogat és cella szavakat.





6.1. ábra. Szimulációs tartomány diszkretizálása véges térfogatok módszerével

A felületi integrálban szereplő differenciáhányadosok a cella oldalai és a középpont közötti differenciahányadossal közelíthetők, a térfogatintegrálok esetén egyponthoz integrálközelítés alkalmazható, így a (6.3) egyenlet a 6.1. ábrán látható hőmérséklet jelölésekkel következő egyenletté alakítható:

$$\begin{aligned} \Delta x \Delta y \Delta z c_v \frac{dT}{dt} - \Delta x \Delta y \Delta z p = & \lambda \left( \frac{T_W - T_C}{\Delta x/2} + \frac{T_E - T_C}{\Delta x/2} \right) \Delta y \Delta z + \\ & \lambda \left( \frac{T_N - T_C}{\Delta y/2} + \frac{T_S - T_C}{\Delta y/2} \right) \Delta x \Delta z + \\ & \lambda \left( \frac{T_T - T_C}{\Delta z/2} + \frac{T_B - T_C}{\Delta z/2} \right) \Delta x \Delta y. \end{aligned} \quad (6.4)$$

A 6.1. ábrán látható termikus modell paraméterei a (6.5) egyenlet összefüggései szerint határozhatóak meg, ahol  $G_{x/2}$ ,  $G_{y/2}$ , és  $G_{z/2}$  a cella középpontja és az oldalak közötti konduktív hővezetés,  $C_{th}$  a cella hőkapacitása, és  $P$  a cellában felszabaduló Joule-hő.

$$\begin{aligned}
G_{x/2} &= \lambda \frac{\Delta y \Delta z}{\Delta x/2} \\
G_{y/2} &= \lambda \frac{\Delta x \Delta z}{\Delta y/2} \\
G_{z/2} &= \lambda \frac{\Delta x \Delta y}{\Delta z/2} \\
C_{th} &= c_v \Delta x \Delta y \Delta z \\
P &= p \Delta x \Delta y \Delta z
\end{aligned} \tag{6.5}$$

Felhasználva az így kifejezett értékeket, a (6.4) egyenlet a következő alakra hozható:

$$\begin{aligned}
C_{th} \frac{dT_C}{dt} - P &= G_{x/2} (T_W - T_C) + G_{x/2} (T_E - T_C) + \\
&G_{y/2} (T_N - T_C) + G_{y/2} (T_S - T_C) + \\
&G_{z/2} (T_T - T_C) + G_{z/2} (T_B - T_C),
\end{aligned} \tag{6.6}$$

amelyben  $T_W, T_N, T_E, T_S, T_T$  és  $T_B$  a cella lapközepeinek,  $T_C$  pedig a középpontjának hőmérséklete,  $\Delta x, \Delta y$  és  $\Delta z$  a cella éleinek hossza (6.1. ábra). Az egyenletben szereplő  $T \cdot G$  szorzatok hőáram dimenziójú mennyiségek.

A teljes háromdimenziós szimulációs tartomány diszkretizálását követően az egyes kontrolltérfogatok az  $(x, y, z)$  irányú koordinátájával lehet azonosítani. A cella oldalain átfolyó hőáram felírható a szomszédos cellák középpontjainak hőmérséklete alapján (48. oldal, 6.2. ábra). Két szomszédos kontrolltérfogat középpontja közötti konduktancia kifejezhető, homogén anyag esetét következő összefüggés írja le<sup>2</sup>:

$$\begin{aligned}
G_x &= \frac{G_{x/2} G_{x/2}}{G_{x/2} + G_{x/2}} = G_{x/2} \times G_{x/2} \\
G_y &= \frac{G_{y/2} G_{y/2}}{G_{y/2} + G_{y/2}} = G_{y/2} \times G_{y/2} \\
G_z &= \frac{G_{z/2} G_{z/2}}{G_{z/2} + G_{z/2}} = G_{z/2} \times G_{z/2}.
\end{aligned} \tag{6.7}$$

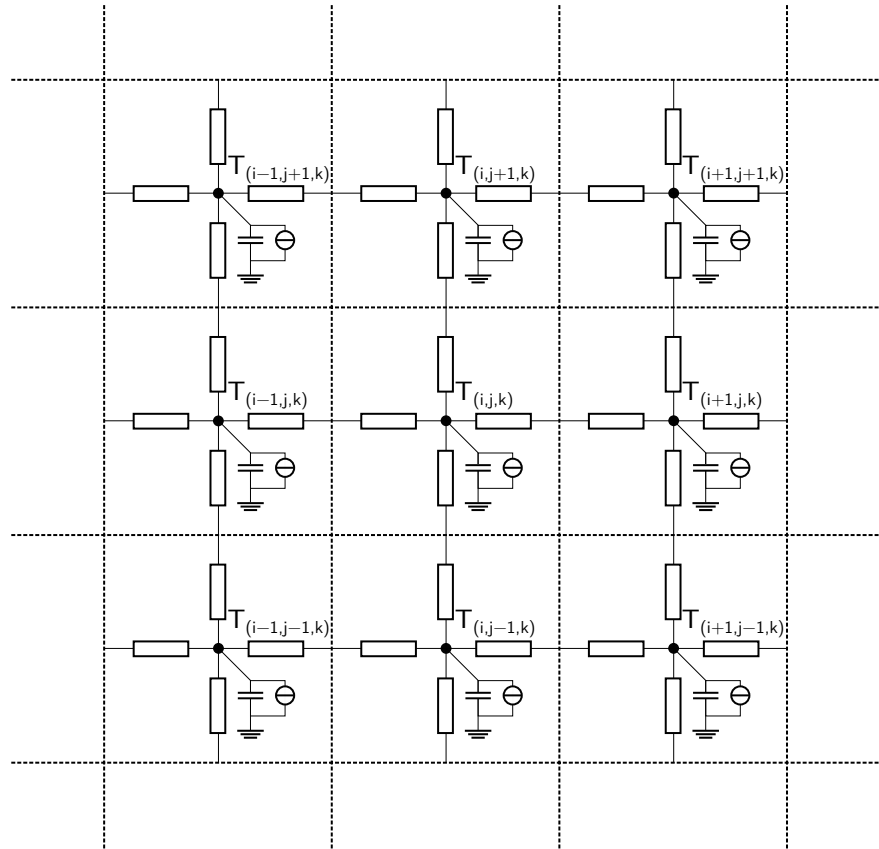
A (6.7) egyenletben kifejezett értékeket használva a (6.6) egyenlet alapján az  $(i, j, k)$  koordinátájú cellára felírható az alábbi egyenlet:

$$\begin{aligned}
C_{th(i,j,k)} \frac{dT_{(i,j,k)}}{dt} - P_{(i,j,k)} &= G_x (T_{(i-1,j,k)} - T_{(i,j,k)}) + G_x (T_{(i+1,j,k)} - T_{(i,j,k)}) + \\
&G_y (T_{(i,j-1,k)} - T_{(i,j,k)}) + G_y (T_{(i,j+1,k)} - T_{(i,j,k)}) + \\
&G_z (T_{(i,j,k-1)} - T_{(i,j,k)}) + G_z (T_{(i,j,k+1)} - T_{(i,j,k)}).
\end{aligned} \tag{6.8}$$

A hőmérsékletek kiszámításához minden cellára fel kell írni a 6.8. közönséges differenciálegyenletet (ODE<sup>3</sup>) és meg kell oldani az így kapott egyenletrendszert. Többféle numerikus módszer használható közönséges differenciálegyenlet-rendszer megoldásához. Az általam vizsgált három módszer tulajdonságai, és a (6.8) egyenletből meghatározott egyenletek a B függelékben olvashatóak.

<sup>2</sup> Inhomogén anyag esete hasonlóan levezethető.

<sup>3</sup> ODE, *Ordinary Differential Equation*



6.2. ábra. Véges térfogatok módszerével felosztott 2 dimenziós tartomány

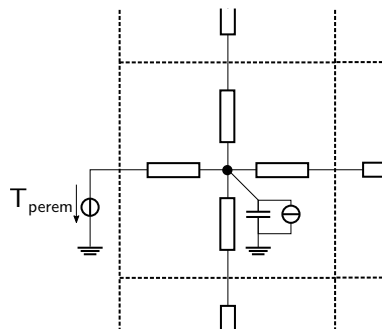
### 6.1.2. Peremfeltételek beépítése a differenciálegyenletbe

A 6.8. ODE egyértelmű megoldhatóságának feltétele, hogy a keresett változó viselkedése ismert legyen a vizsgált tartomány határán. Termikus modellezés során a következő peremfeltételek fordulnak elő.

#### Dirichlet peremfeltétel

A szimulációs tartomány határán a felületi hőmérséklet konstans:

$$T_{perem} = \text{konstans.} \quad (6.9)$$



6.3. ábra. Dirichlet peremfeltétel

A (6.8) egyenlet a következő szerint módosul, amennyiben a  $(i, j, k)$  cella bal szomszédja  $(i - 1, j, k)$  Dirichlet peremfeltétel (6.3. ábra):

$$\begin{aligned} C_{th(i,j,k)} \frac{dT_{(i,j,k)}}{dt} - P_{(i,j,k)} &= G_{x/2} (T_{perem} - T_{(i,j,k)}) + G_x (T_{(i+1,j,k)} - T_{(i,j,k)}) + \\ &G_y (T_{(i,j-1,k)} - T_{(i,j,k)}) + G_y (T_{(i,j+1,k)} - T_{(i,j,k)}) + \\ &G_z (T_{(i,j,k-1)} - T_{(i,j,k)}) + G_z (T_{(i,j,k+1)} - T_{(i,j,k)}). \end{aligned} \quad (6.10)$$

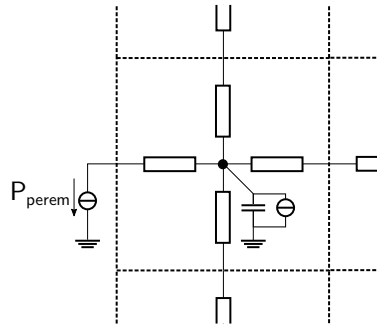
Látható, hogy a  $G_{x/2} \cdot T_{perem}$  szorzat ismert áram dimenziójú mennyiség, így ezt a tagot át lehet vinni az egyenlet bal oldalára.

### Neumann peremfeltétel

A keresett függvénynek a felület normálisa szerinti deriváltja konstans, termikus esetben ez konstans hőáramot jelent:

$$\frac{\partial T}{\partial n} = P_{perem} = \text{konstans}. \quad (6.11)$$

Ennek speciális esete az adiabatikus peremfeltétel, amelynél a hőáram a környezet felé nulla.



6.4. ábra. Neumann peremfeltétel

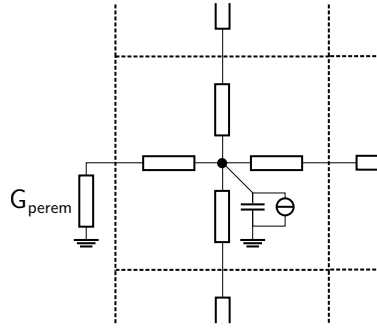
Bal oldali Neumann peremfeltétel esetén (6.4. ábra) a (6.8) egyenletben a szomszéd irányába felírt  $G_x (T_{(i-1,j,k)} - T_{(i,j,k)})$  hőáram a peremfeltételben megadott árammal helyettesítendő:

$$\begin{aligned} C_{th(i,j,k)} \frac{dT_{(i,j,k)}}{dt} - P_{(i,j,k)} &= P_{perem} + G_x (T_{(i+1,j,k)} - T_{(i,j,k)}) + \\ &G_y (T_{(i,j-1,k)} - T_{(i,j,k)}) + G_y (T_{(i,j+1,k)} - T_{(i,j,k)}) + \\ &G_z (T_{(i,j,k-1)} - T_{(i,j,k)}) + G_z (T_{(i,j,k+1)} - T_{(i,j,k)}). \end{aligned} \quad (6.12)$$

Mivel ennek a peremfeltétel értéke ismert, így átvihető az egyenlet bal oldalára.

### Robin peremfelétel

A felületi hőmérséklet, és a hőmérséklet felületi normális szerinti deriváltjának (hőáramsűrűség) hányadosa állandó, ami konduktív peremfeltételnek felel meg:



6.5. ábra. Robin peremfeltétel

$$\frac{\partial T}{\partial n} = \frac{P_{perem}}{T_{perem}} = \text{konstans.} \quad (6.13)$$

Robin peremfeltétel alkalmazásakor (6.5. ábra) az adott peremfelület konduktív tagon keresztül kapcsolódik a környezeti hőmérsékletet modellező „termikus föld” csomóponthoz, így az a (6.8) egyenlet módosul:

$$\begin{aligned} C_{th(i,j,k)} \frac{dT_{(i,j,k)}}{dt} - P_{(i,j,k)} &= G_{perem} \times G_{x/2} (T_{föld} - T_{(i,j,k)}) + G_x (T_{(i+1,j,k)} - T_{(i,j,k)}) + \\ &G_y (T_{(i,j-1,k)} - T_{(i,j,k)}) + G_y (T_{(i,j+1,k)} - T_{(i,j,k)}) + \\ &G_z (T_{(i,j,k-1)} - T_{(i,j,k)}) + G_z (T_{(i,j,k+1)} - T_{(i,j,k)}). \end{aligned} \quad (6.14)$$

A  $(G_{perem} \times G_{x/2}) T_{föld}$  tag ebben az esetben is átvihető az egyenlet túloldalára.

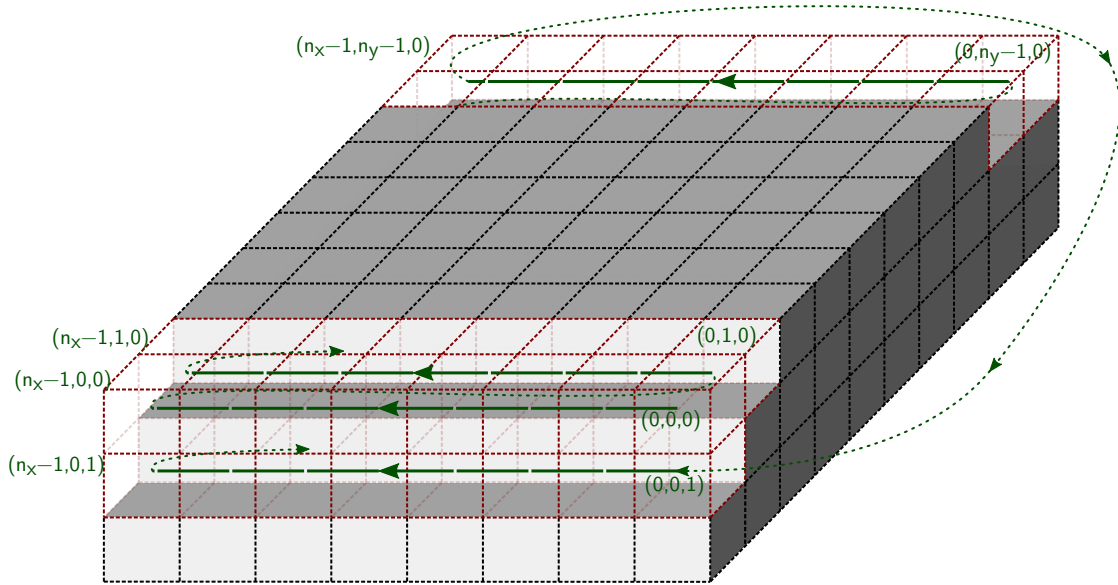
## 6.2. A megoldandó differenciálegyenlet-rendszer

A szimulációs tartomány minden kontrolltérfogatra felírható a peremfeltételekkel módosított (6.8) egyenlet. Az így meghatározott egyenletek megfelelő indexeléssel egy  $n$  darab egyenletből álló egyenletrendszerre alakíthatóak, ahol  $n$  a kontrolltérfogatok – így a  $T_{(i,j,k)}$ -k – száma. A 6.6. ábrán zölddel jelölt bejárást választva a szimulációs tartomány háromdimenziós koordinátarendszere egydimenzióssá alakítható: amennyiben a tartomány  $x, y, z$  koordinátatengelyek mentén rendre  $n_x, n_y, n_z$  részre van osztva, tetszőleges  $(i, j, k)$  koordináta megegyezik az egydimenziós  $(i + j \cdot n_x + k \cdot n_x \cdot n_y)$  koordinátával.

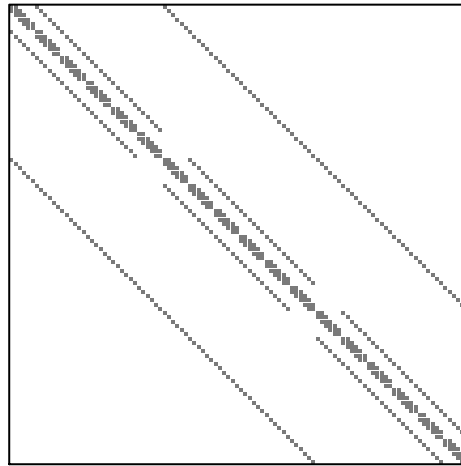
A (6.8) egyenletet minden kontrolltérfogatra felírva a 6.6. ábrán jelölt sorrendben a kontrolltérfogatok  $T_{(i,j,k)}$  hőmérsékletéből  $\bar{T}$ , a gerjesztések és a peremfeltételek összevonásával  $\bar{P}$  oszlopvektor, a  $C_{th(i,j,k)}$  hőkapacitásokból és  $G_x, G_y, G_z$  hővezetésekéből pedig  $\mathbf{C}_{th}$  és  $\mathbf{G}$  mátrixok képezhetőek. A  $\mathbf{C}_{th}$  mátrix a főátlóban tartalmazza az egyes  $C_{th(i,j,k)} > 0$  értékeket. A  $\mathbf{G}$  egy szimmetrikus sávmátrix (6.7. ábra), főátlója negatív, a sor illetve oszlop többi nem nulla eleme pozitív. A sorösszegek nullával egyeznek meg, azaz teljesül az energiamegmaradás.

Az új jelölésekkel a (6.8) egyenlet az alábbi egyenletté alakítható:

$$\mathbf{C}_{th} \frac{d\bar{T}}{dt} - \bar{P} = \mathbf{G}\bar{T}. \quad (6.15)$$



6.6. ábra. Háromdimenziós koordinátarendszer transzformálása egydimenzióssá

6.7. ábra.  $\mathbf{G}$  mátrix nem nulla elemei

A hőmérsékletek kiszámításához ezt a differenciál-egyenletrendszert kell megoldani ismert perem- és kezdeti feltételekkel. Átrendezve ezt az egyenletrendszert kifejezhető a hőmérsékletvektor idő szerinti deriváltja:

$$\frac{d\bar{T}}{dt} = \mathbf{C}_{th}^{-1} (\mathbf{G}\bar{T} + \bar{P}). \quad (6.16)$$

Az  $\mathbf{C}_{th}^{-1}\mathbf{G}$  mátrix domináns főátlójú, a diagonális elemei negatívak, a Gersgorin körök tétele következtében a sajátértékek a bal félsíkon vannak ( $Re(\lambda_i) < 0$ ). A fenti egyenletrendszer megoldására három numerikus eljárást – előrelépő Eulert, hátralépő Eulert és trapéz módszert – vizsgáltam és valósítottam meg a termikus motorban. Az előrelépő Euler módszer alkalmazása esetén a hőmérsékletek meghatározásához egyszerű mátrix-vektor szorzást és vektor-vektor összeadást kell elvégezni, hátralépő Euler és trapéz módszer esetén lineáris egyenletrendszert kell megoldani. Az alkalmazott numerikus módszerek stabilitási tulajdonságai a B függelékben, kvantitatív összehasonlításuk pedig a 6.3.2. fejezetben olvasható.

### 6.3. Az új termikus szimulációs motor vizsgálata

A termikus motort C++ nyelven implementáltam. A lineáris egyenletrendszerek megoldásához számtalan lineáris algebra könyvtár áll rendelkezésre, például ATLAS<sup>4</sup> [101], LAPACK<sup>5</sup> [102], és SuperLU<sup>6</sup> [103]. A fejlesztés során első számú szempont volt, hogy az új termikus szimulációs motor képes legyen kihasználni a modern többmagos CPU és GPU processzorarchitektúrákat. A PARALUTION [98] és a ViennaCL [104] képes számításokat futtatni CPU-n és GPU<sup>7</sup>-n is, végül a dokumentáció alapján az előbbi mellett döntöttem.

#### 6.3.1. A megoldó algoritmus kiválasztása és paraméterezése

A PARALUTION könyvtár megoldó algoritmusainak előzetes tesztje alapján a konjugált gradiens módszer bizonyult a leggyorsabbnak, így ezt használtam a termikus szimulációs motorban<sup>8</sup>. Ez a módszer tekinthető iteratív, illetve direkt módszernek is, mivel a számábrázolási pontosságból adódó hibáktól eltekintve az algoritmus  $n$  lépést követően előállítja a pontos megoldást, ahol  $n$  az egyenletrendszert leíró mátrix rangja<sup>9</sup>. A konjugált gradiens módszer azonban ennél kevesebb lépést követően is elérheti a kívánt pontosságot. Ebben az esetben az iteráció leállításának feltétele, hogy a legnagyobb maradvány kisebb legyen egy előre beállított  $\varepsilon$  értéknél:

$$\begin{aligned} \bar{r} &= \left| \bar{P} - \mathbf{M} \cdot \bar{T} \right| \\ \max(\bar{r}) &< \varepsilon, \end{aligned} \quad (6.17)$$

ahol  $\mathbf{M}$  a lineáris egyenletrendszert leíró mátrix. Hátralépő Euler módszer esetén ez lehet

$$\mathbf{M} = -\mathbf{G} + \frac{\mathbf{C}_{\text{th}}}{\Delta t}, \quad (6.18)$$

vagy trapéz módszer esetén

$$\mathbf{M} = -\mathbf{G} + \frac{2\mathbf{C}_{\text{th}}}{\Delta t}. \quad (6.19)$$

A maradvány fizikai jelentése az adott kontrolltérfogatot bezáró felületen átfolyó hőáram eltérése a pontos megoldás esetén átfolyó hőáramtól ( $\mathbf{M} \cdot \bar{T}_0 = \bar{P}$ ). A pontos megoldást ( $\bar{T}_0$ ) visszahelyettesítve a (6.17) egyenletbe kifejezhető a számított megoldás és a pontos megoldás távolsága:

$$\bar{r} = \left| \mathbf{M} \cdot \bar{T}_0 - \mathbf{M} \cdot \bar{T} \right| = \left| \mathbf{M} \cdot \Delta \bar{T} \right|. \quad (6.20)$$

Legkedvezőtlenebb esetet feltételezve a teljes  $r_{(i,j,k)}$  eltérés a termikus modellben található legkisebb konduktancián keresztül folyik el szomszédos csomópontba vagy a környezet felé, a legnagyobb  $\Delta T_{(i,j,k)}$ -t okozva:

$$\min \left( G_{x_{(i,j,k)}}, G_{y_{(i,j,k)}}, G_{z_{(i,j,k)}}, G_{C_{(i,j,k)}} \right) \cdot \Delta T_{(i,j,k)} \leq r_{(i,j,k)} < \varepsilon, \quad (6.21)$$

<sup>4</sup> ATLAS, *Automatically Tuned Linear Algebra Software*

<sup>5</sup> LAPACK, *Linear Algebra Package*

<sup>6</sup> SuperLU, *Supermodal LU*

<sup>7</sup> GPU, *Graphics Processor Unit*

<sup>8</sup> Előrelépő Euler módszer esetén erre nincs szükség

<sup>9</sup> A mátrix rangja megegyezik a csomópontok számával.

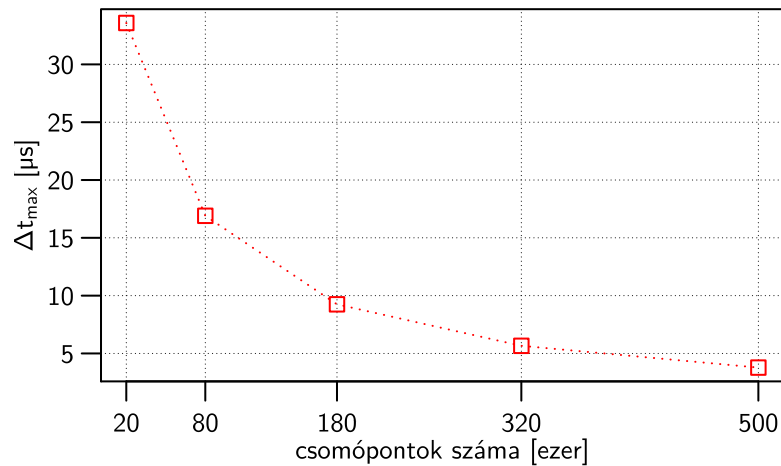
amelyben  $G_{C_{(i,j,k)}}$  a  $C_{th_{(i,j,k)}}$  helyettesítőképekének konduktív eleme. Az alkalmazott numerikus módszer – visszalépő Euler vagy trapéz módszer – függvényében  $C_{th_{(i,j,k)}}$ / $\Delta t$  vagy  $2C_{th_{(i,j,k)}}$ / $\Delta t$ . Ezt a felülbecslést alkalmazva  $\varepsilon$  meghatározásakor garantált, hogy a megadott  $\Delta T$  hőmérsékleti pontosságnál minden  $\Delta T_{(i,j,k)}$  kisebb lesz:

$$\min_{\forall(i,j,k)} \left( G_{x_{(i,j,k)}}, G_{y_{(i,j,k)}}, G_{z_{(i,j,k)}}, G_{C_{(i,j,k)}} \right) \cdot \Delta T = \varepsilon. \quad (6.22)$$

### 6.3.2. Numerikus módszer kiválasztása

Az új termikus szimulációs motorba a már említett három numerikus eljárást építettem be a (6.8) egyenletrendszer megoldásához. Ebben a fejezetben stabilitás, pontosság és futási idő szempontjából hasonlítom össze ezeket. Az eredményeket a validációnál is használt elrendezés<sup>10</sup> mellett mértem egy hatmagos AMD Phenom II X6 1055T processzoral, 8 GiB memóriával, és egy AMD Radeon HD 6970 videokártyával felszerelt PC-n.

#### Az előrelépő Euler módszer stabilitása



6.8. ábra. Előrelépő Euler módszer stabilitásának határhelyzete

A B függelékben olvasható a három numerikus módszer stabilitásának feltétele. A hátralépő Euler és a trapéz módszer esetén az időlépés elméletileg tetszőlegesen nagyra választható, azonban az előrelépő Euler módszer esetén teljesülnie kell a B.4. feltételnek. Ezért a szimulációs motorba beépítettem egy eljárást, ami az adott struktúra és térbeli felbontások alapján meg tudja határozni a stabilitás határhelyzetét, vagyis az alkalmazható legnagyobb időlépést. Ehhez a szabadon hozzáférhető Spectra [105] és Eigen [106] könyvtárakat használtam fel. A 6.8. ábra illusztrálja, hogy a felbontás növelése<sup>11</sup> milyen hatással van a legnagyobb megengedett időlépésre. Látható, hogy részletes hőterkép esetén egyre kisebb időlépést kell alkalmazni a stabilitás megtartásához, ami korlátozza ennek a módszernek az alkalmazhatóságát.

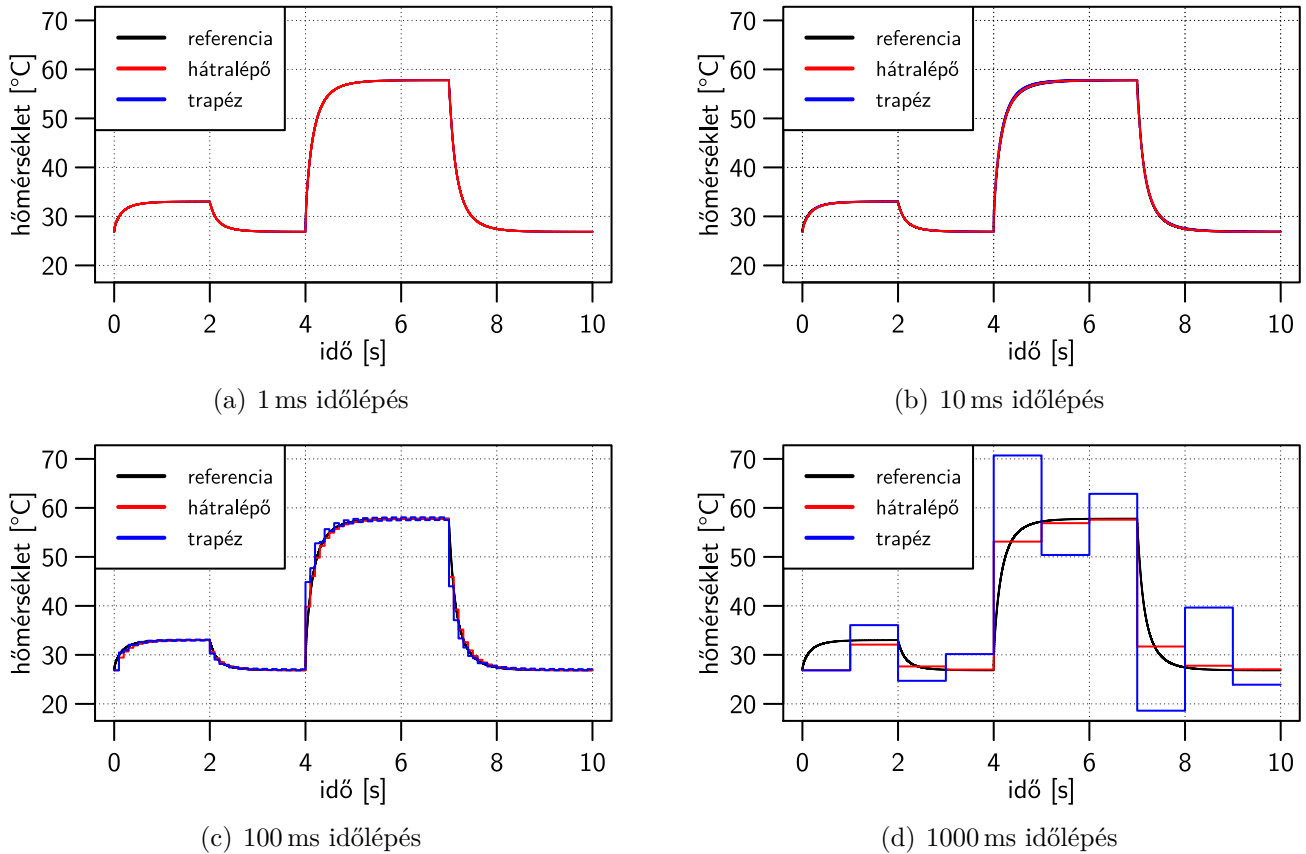
<sup>10</sup> A termikus motor validációja az 56. oldalon a 6.3.3. fejezetben olvasható.

<sup>11</sup> A lapka minden esetben két réteget tartalmazott, a felbontást laterális irányban változtattam 100×100-tól 500×500-ig.



### A hátralépő Euler és trapéz módszerek pontossága

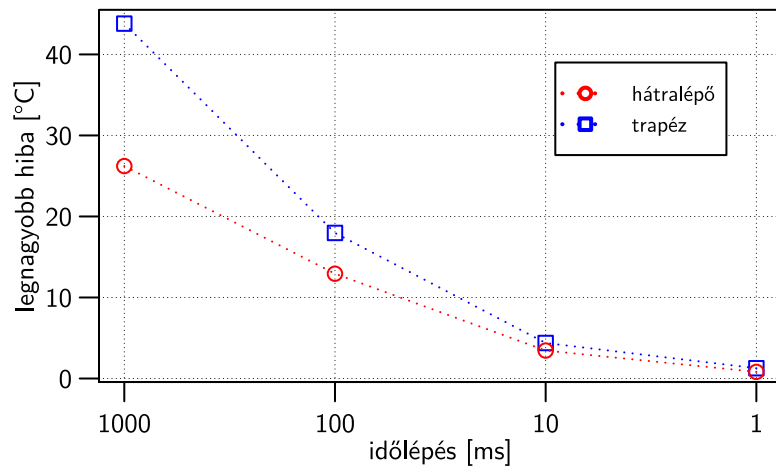
A hátralépő Euler és a trapéz módszerek pontosságát megvizsgáltam különböző időlépések mellett. A 6.9. ábrán látható a lapka maximális hőmérséklete 1 ms, 10 ms, 100 ms és 1000 ms esetén. A szimulációk során 80 ezer csomópontos térbeli felbontást használtam, a gerjesztés megegyezik a termikus motor validációja során alkalmazott profillal (6.13(a) ábra). Referenciának az 1  $\mu$ s időlépést alkalmazó hátralépő Euler módszerrel kapott eredményt vettem. Az időlépés növekedése láthatóan nagyobb hibát okoz a trapéz módszer esetén, kis mértékű oszcilláció már a 10 ms esetben is látható, és 1000 ms esetén már jelentős eltérést okoz a referenciához képest.



6.9. ábra. Numerikus módszerek pontossága különböző időlépés mellett

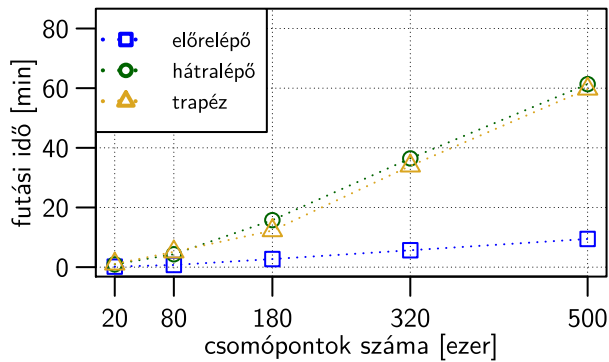
A 6.10 ábra illusztrálja a két módszer legnagyobb hibáját az időlépés függvényében. A trapéz módszernek 1000 ms időlépésnél jelentősen nagyobb a hibája, de az időlépés csökkentése a hibát is jobban csökkenti a hátralépő Euler módszerhez viszonyítva (B függelék). Annak ellenére, hogy a trapéz és a hátralépő Euler módszer is stabil, előbbi alkalmazása nagyobb hibát és időben lecsengő oszcillációt eredményez nagyobb ( $>1$  ms) időlépések mellett.

A későbbiekben referenciaként használt SUNRED és 3D-ICE termikus szimulátorok hátralépő Euler módszert alkalmaznak az ODE megoldásához. Az irodalmi áttekintésben a 2.2.1. fejezetben említett HotSpot termikus motor Runge-Kutta módszert használ erre a célra. A Runge-Kutta módszer egy negyedrendű módszer, egy időlépésen belül négy ponton számítja ki a keresett függvény meredekségét. Pontossága kedvezőbb lehet az általam alkalmazott első- és másodrendű módszerekhez képest, azonban az implementációja is bonyolultabb.

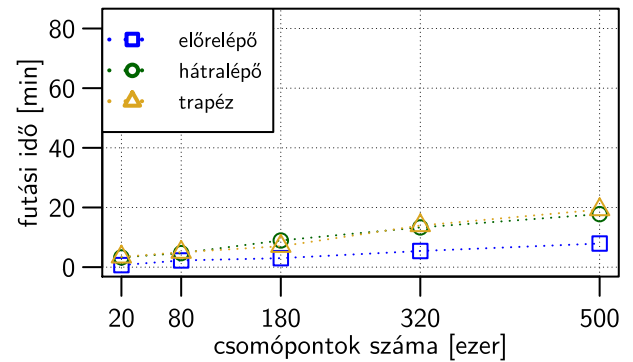


6.10. ábra. A hátralépő Euler és a trapéz módszer hibája az alkalmazott időlépés függvényében

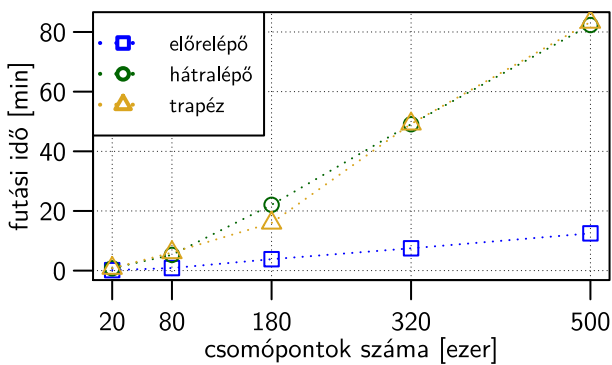
### Numerikus módszerek futási idejeinek vizsgálata



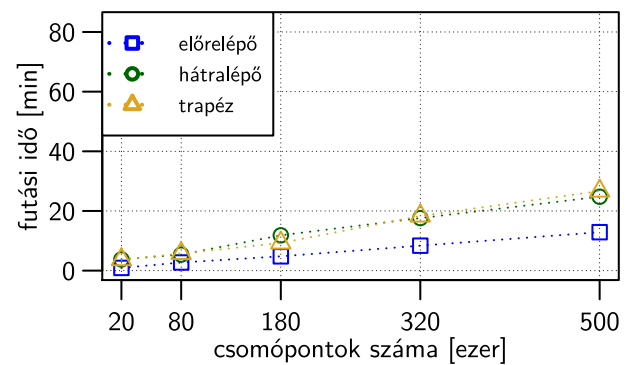
(a) CPU-s megoldó egyszeres pontosság esetén



(b) GPU-s megoldó egyszeres pontosság esetén



(c) CPU-s megoldó dupla pontosság esetén



(d) GPU-s megoldó dupla pontosság esetén

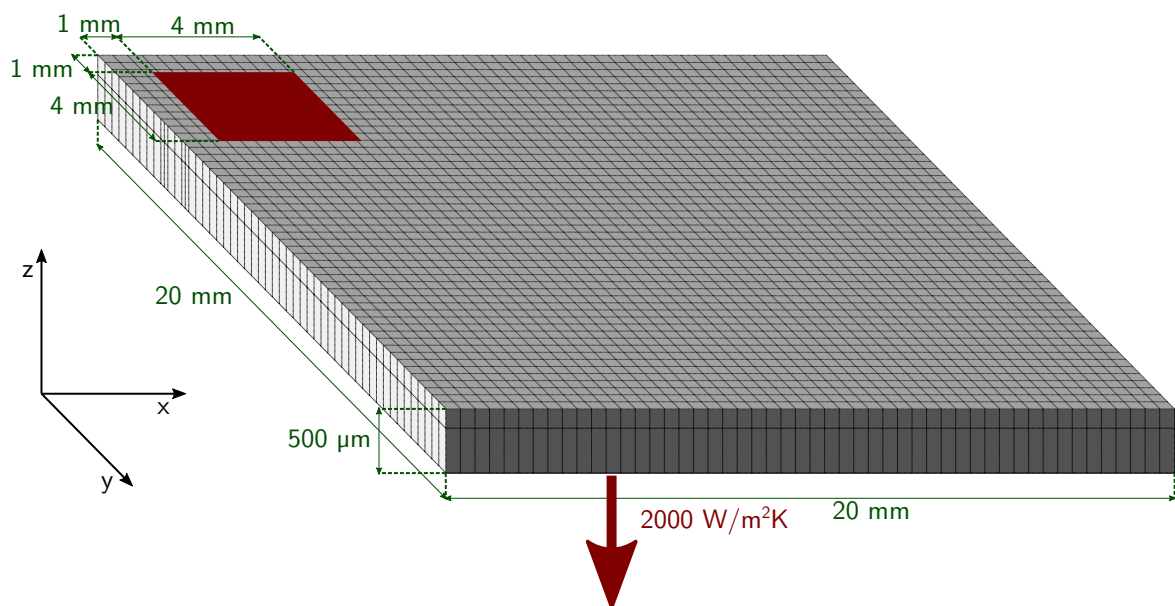
6.11. ábra. Numerikus módszerek sebessége különböző felbontások mellett

A 6.11. ábrán látható a termikus szimulációk futási ideje a térbeli felbontás, alkalmazott numerikus módszer, pontosság és a processzor architektúrájának függvényében. A szimulációk során egységugrás gerjesztést és  $1\ \mu\text{s}$  szimulációs lépésközt alkalmaztam az ábrán feltüntetett

térbeli felbontások mellett<sup>12</sup>. Az előrelépő Euler módszer számottevően gyorsabb a másik két módszernél ilyen lépésköz mellett, amely nem meglepő a kis számításigény ismeretében – egy lépésben csak két mátrix-vektor szorzásra és összeadásra van szükség. A hátralépő Euler és a trapéz módszer esetén azonban lehetőség van a szimuláció időlépését a kívánt időbeli felbontáshoz igazítani. Érdekesség, hogy a GPU-s megoldó esetén az előrelépő Euler előnye sokkal kisebb a CPU-s esethez viszonyítva. Ennek az az oka, hogy GPU használata esetén az adatok mozgatása a rendszermemória és a videokártya memóriája között időbe telik, ami csökkenti a kis számításigényű módszer hatékonyságát.

### 6.3.3. A termikus motor validációja

Az új termikus motor pontosságát a SUNRED és a 3D-ICE szimulátorokhoz hasonlítottam dupla pontosságú lebegőpontos számábrázolás mellett<sup>13</sup>. Mindhárom szimulátorban a 6.12. ábrán látható struktúrát hoztam létre.



6.12. ábra. Az új termikus motor validációjához használt elrendezés<sup>14</sup>

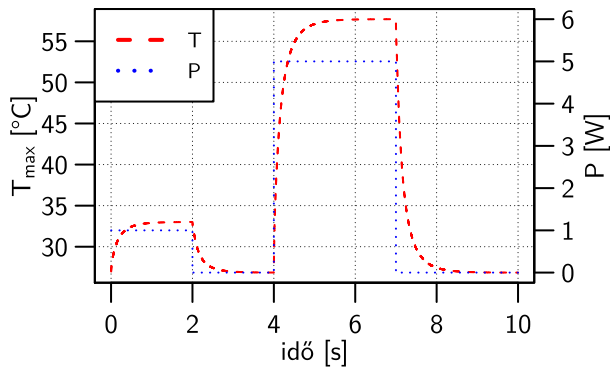
A chip  $20\text{ mm} \times 20\text{ mm} \times 500\text{ }\mu\text{m}$  méretű, amelynek a sarkában egy  $4\text{ mm} \times 4\text{ mm}$  méretű disszipáló elem található. A lapkát vertikálisan két részre, egy  $10\text{ }\mu\text{m}$  és egy  $490\text{ }\mu\text{m}$  vastag rétegre osztottam, laterális irányban  $200 \times 200$ -as ekvidisztáns felbontást használtam. Adiabaticus peremfeltételeket alkalmaztam az alsó oldalt kivéve, ahol  $2000\text{ W/m}^2\text{ K}$  hőátadási együtthatót állítottam be. A validáció során hátralépő Euler módszert választottam  $1\text{ ms}$  szimulációs időlépéssel.

A validáció során a 6.13(a) ábrán látható gerjesztési profilt használtam. A különböző termikus szimulációs motorok által számolt eredmények közötti legnagyobb hőmérsékleteltérést a 6.13(b) ábra mutatja be. A 3D-ICE és SloTh között a legnagyobb eltérés a szimuláció negyedik másodpercénél jelentkezett, a hőmérsékletkülönbség  $\sim 0,05\text{ }^\circ\text{C}$ . A SUNRED-hez viszonyítva a legnagyobb eltérés  $\sim 0,2\text{ }^\circ\text{C}$ , ami a szimuláció hetedik másodpercénél lépett fel. A 6.13(c) és a 6.13(d) ábrák a hőtérképek közötti különbséget mutatják be a legnagyobb különbségek esetén.

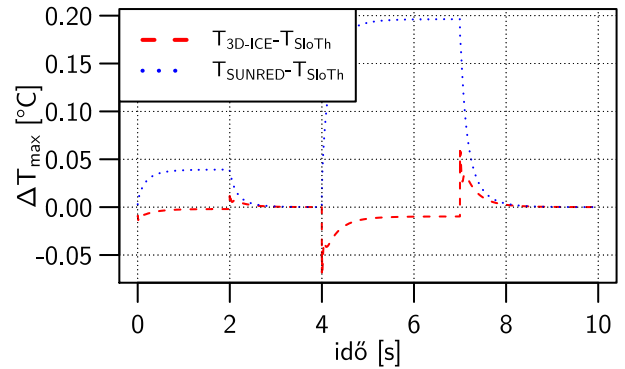
<sup>12</sup> A lapka minden esetben két réteget tartalmazott, a felbontást laterális irányban változtattam  $100 \times 100$ -tól  $500 \times 500$ -ig.

<sup>13</sup> Egyszeres pontosság mellett kapott eredmények  $\sim 0,1\text{ }^\circ\text{C}$ -al tértek el.

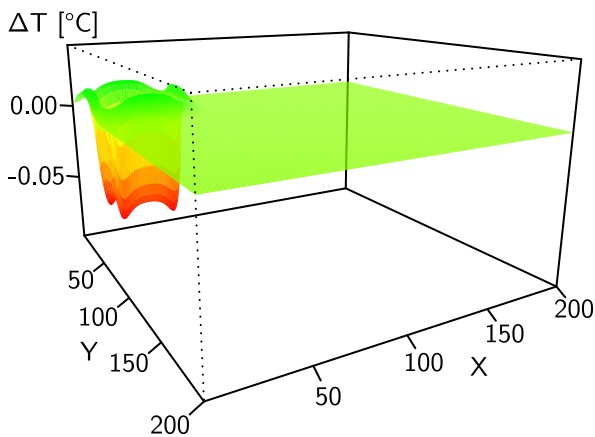
<sup>14</sup> Az ábra nem méretarányos.



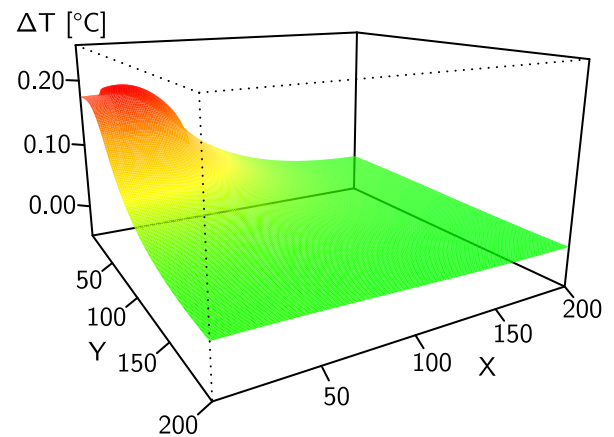
(a) A validáció során alkalmazott gerjesztés és válasz



(b) Hőmérsékletek közötti legnagyobb eltérés



(c) Legnagyobb különbség 3D-ICE és SloTh között



(d) Legnagyobb különbség SUNRED és SloTh között

6.13. ábra. Az új termikus motor validációja

### 6.3.4. A futási idő vizsgálata

Az új termikus szimulátor futási idejét összehasonlítottam a SUNRED és 3D-ICE szimulátorok futási idejével a 6.3.2. fejezetben leírt PC konfigurációval. A szimulációkhoz a validációnál is használt struktúrát és beállításokat használtam fel. A 6.14. ábrán a termikus szimulációs motorok futási ideje látható 100×100-tól 500×500-ig változó ekvidisztáns laterális felbontás mellett<sup>15</sup>.

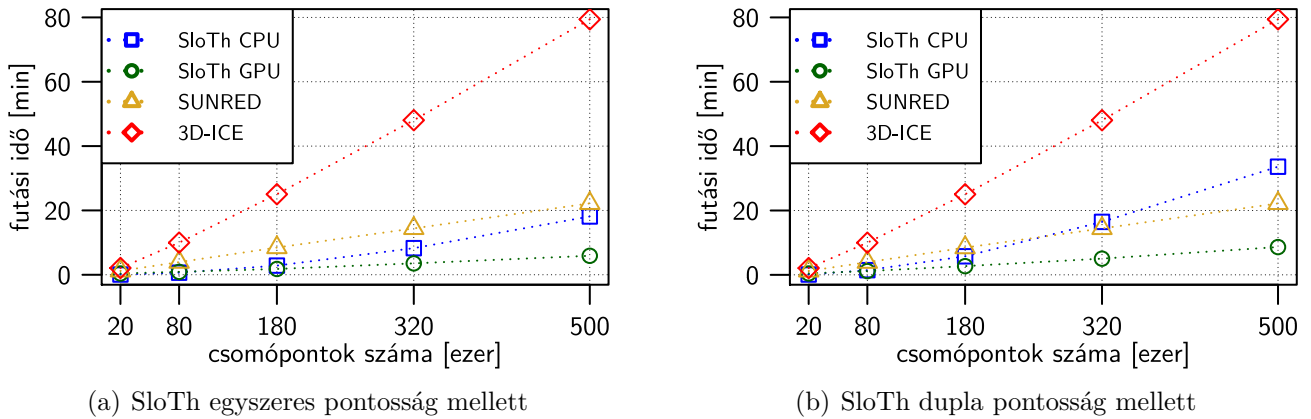
Az optimalizált Intel® MKL<sup>16</sup> [107] és OpenBLAS<sup>17</sup> [108] lineáris algebra könyvtárak használata ellenére a 3D-ICE termikus szimulációs motor bizonyult a leglassabbnak a három vizsgált szimulátor közül. Egy lehetséges szűk keresztmetszet az egyszálú végrehajtásra tervezett SuperLU könyvtár használata. Ennek létezik többszálú változata is, azonban a két verzió programozói interfésze nem azonos, így utóbbira áttérni a 3D-ICE motor mélyebb ismerete nélkül nem triviális. A motor nyilvános változata az egyszálú megoldást alkalmazza.

A SUNRED termikus szimulációs motor képes kihasználni több processzormagot, a mérések alapján átlagosan körülbelül három magot használt a hatból. Ezen kívül nagyon jól skálázódik a probléma méretével, megközelíti az  $O(n)$ -t konstans időlépték mellett. Pohl László doktori disszertációjában részletesebb összefüggés is található a csomópontok száma és a futási idő között [31].

<sup>15</sup> Ezeknél a szimulációknál is két rétegű lapkát alkalmaztam.

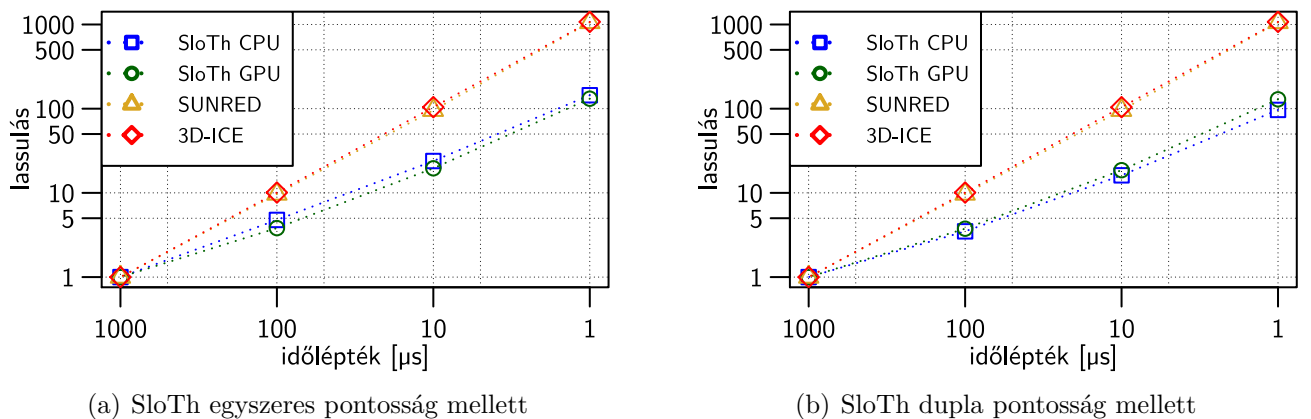
<sup>16</sup> Intel® MKL, Intel® *Math Kernel Library*

<sup>17</sup> OpenBLAS, *Open Basic Linear Algebra Subprograms*



6.14. ábra. Termikus motorok futási ideje a csomópontok számának függvényében

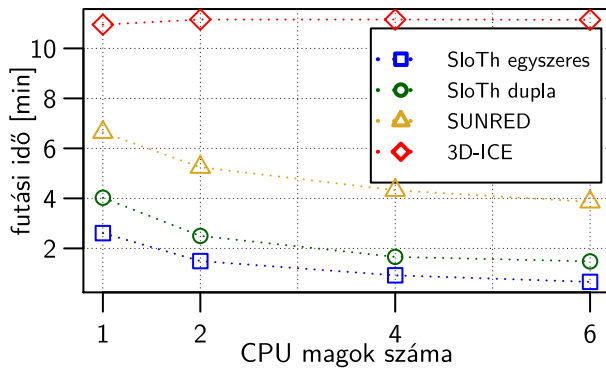
A SloTh CPU-s implementációjának előnye kis felbontás mellett körülbelül 10-szeres a SUNRED-hez képest. A probléma méretével ez az előny csökken a gyengébb skálázódás miatt, nagyon nagy térbeli felbontás esetén pedig akár lassabb is lehet mint a SUNRED. A SloTh GPU-s megvalósítása kisebb felbontásokon lassabb mint a CPU-s, azonban sok csomópontot tartalmazó szimulációk esetén jelentősen gyorsabb.



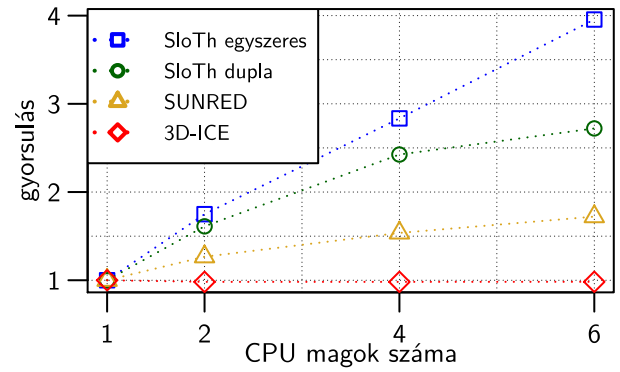
6.15. ábra. Az időlépés hatása a szimuláció sebességére

A 6.15. ábrán az időlépés hatása látható a szimuláció sebességére. Referenciának az előző szimulációs eredmények előállításakor is alkalmazott 1 ms lépésközt választottam, a termikus modell 80 ezer csomópontot tartalmazott. Látható, hogy a direkt megoldó algoritmust használó 3D-ICE és SUNRED motorok futási ideje ezerszeresre nő 1  $\mu$ s lépésköz esetén, az iteratív konjugált gradiens módszert használó SloTh termikus motor esetén százszoros futásidő növekedést tapasztaltam. Ez annak köszönhető, hogy sűrűbb lépésköz esetén nem változnak olyan nagy mértékben a hőmérsékletek két szimulációs lépés között, így a következő időpillanatra kevesebb iterációval lehet előállítani az ugyanolyan pontosságú eredményt.

A 6.16. ábrán a termikus szimulációs motorok futási ideje és az egymagos futásidőhöz képest tapasztalt gyorsulás látható a processzormagok számának függvényében. A SloTh és a SUNRED termikus motorok gyorsulnak több mag alkalmazásával, a 3D-ICE kis mértékben (~1%) lassul. Ez utóbbit valószínűleg az okozza, hogy az operációs rendszer ütemezője többmagos végrehajtás



(a) Futási idő CPU magok számának függvényében



(b) Gyorsulás CPU magok számának függvényében

6.16. ábra. Termikus motorok futási ideje és a gyorsulás CPU magok számának függvényében

esetén változthatja a szimulátor folyamatát a magok között, ez pedig ronthatja a gyorsítótárak hatásfokát.

## 6.4. A szimuláció időlépésének megválasztása

Relaxációs módszer alkalmazása esetén fontos a szimulációs motorok közötti adatsere gyakoriságának helyes megválasztása. A funkcionális szimulációs motor két egymást követő termikus szimuláció között konstans hőmérsékletet feltételez. Amennyiben az együttes szimulációs ciklus túl nagy időlépést alkalmaz, a hőmérsékletek nagy mértékben változhatnak a soron következő termikus szimulációs lépésben, ami hibás eredményre vezethet.

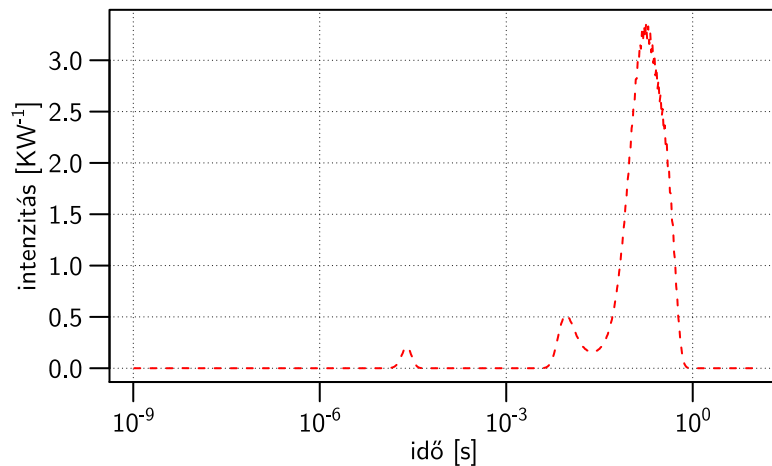
A probléma azonban kiküszöbölhető, ha a funkcionális és termikus motorok közötti adatsere egy iterációja a közös ciklusban jelentősen rövidebb szimulációs időközönként történik, mint a termikus modellben lévő layout alakzatok ( $L_i$ ) legkisebb termikus időállandója. Ez a feltétel azt jelenti, hogy két iteráció között olyan kevés idő telik el, hogy a layout alakzatok, és így a funkcionális modell komponenseinek ( $F_i$ ) hőmérséklete csak kis mértékben változik, ami a (2.7) egyenlet  $\mathbf{J}_F$  Jakobi mátrixának hiányzó parciális deriváltjai ellenére kis hibát okoz<sup>18</sup>.

### 6.4.1. Konstans időlépés

A konstans időlépés megválasztásakor kompromisszumot kell kötni a pontosság és a szimuláció sebessége között. A termikus időállandó-spektrumot minden layout alakzatra ( $L_i$ ) meghatározva kiválasztható az a legkisebb termikus időállandó, amelynek még van szerepe a funkcionális és termikus tartományok csatolásában. Elhagyható az a termikus időállandó, amely olyan  $L_i$ -hez tartozik, ahol az  $F_i$ -ben nincs hőmérsékletfüggés modellezve.

A 6.17. ábrán látható folytonos időállandó-spektrumot NID módszerrel [58] határoztam meg az 56. oldalon a 6.12. ábrán látható elrendezésre. A vízszintes tengelyen az időállandó látható logaritmikus skálán, a függőleges tengelyen a hozzá tartozó intenzitás szerepel.

<sup>18</sup> A  $(\partial I_m / \partial T_j) \cdot \Delta T_j$  szorzat kicsi.



6.17. ábra. A validáció során alkalmazott elrendezés időállandó-spektruma

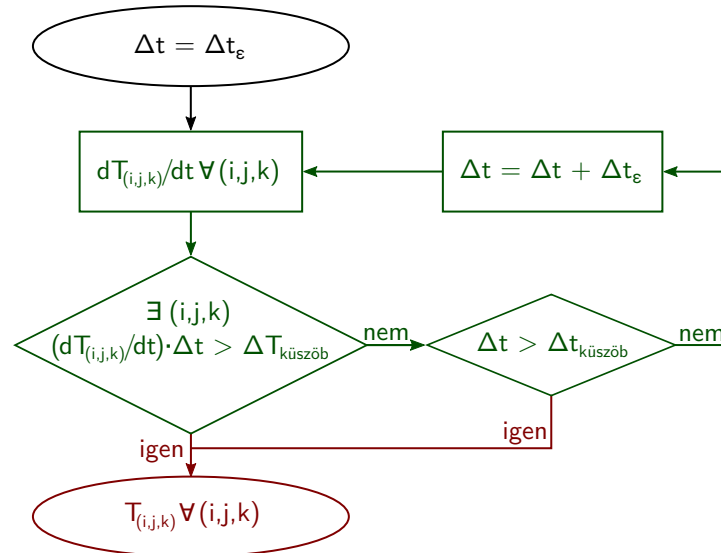
### 6.4.2. Adaptív időlépés

Gyors hőmérséklet-változások lekövetéséhez és az állandósult állapot közelében eltérő időlépést érdemes alkalmazni. Tranziensek elején a hőmérséklet nagyobb mértékben változik, ezért rövid időközönként kell megoldani az egyenletrendszert a pontos eredmény érdekében. Termikus rendszerek szimulációja és mérése során bevett gyakorlat a szimuláció lépésközét illetve a mintavételek időpontját logaritmusos időskálán felvenni, mert az egységugrású gerjesztésre exponenciális jellegű a rendszer válasza. Ez a megközelítés nem alkalmazható funkcionális-termikus szimuláció során, mivel a gerjesztés időben nem állandó, jövőbeli értéke ismeretlen.

A kutatásom során vizsgált HotSpot, 3D-ICE és SUNRED termikus motorok nem alkalmaznak olyan algoritmust, amely a gerjesztés alapján automatikusan választja meg a termikus szimuláció időlépését. A HotSpot és a 3D-ICE esetén kizárólag konstans időlépés alkalmazható, a SUNRED esetén a lépésköz módosítható futás közben is, ebben az esetben a termikus szimulátort hívó programnak kell biztosítani az új időlépést is.

A TRANZ-TRAN elektromos és elektro-termikus szimulátor olyan adaptív algoritmussal rendelkezik, amely túl nagy szimulációs lépésköz esetén csökkenti azt [109]. Nagy időlépés esetén előfordulhat, hogy az egyenletmegoldó algoritmus nem a megoldást állítja elő, hanem egy másik lokális minimumot talál. A TRANZ-TRAN ebben az esetben eldobja az így előállított eredményt, csökkenti a szimuláció lépésközét, és az állapottér egy véletlenszerűen kiválasztott pontjából kiindulva újratekinti a számítást. Ehhez hasonló módszert alkalmaznak a 2.2.1. fejezetben említett ANSYS® termékek is. Szimulátorcsatoláson alapuló együttes szimulációs eljárás esetén akkor használható ez az adaptív algoritmus, ha az egyes szimulátor motorokat „vissza” is lehetne léptetni, amennyiben a számított eredmények nem eléggé pontosak vagy divergálnak. Ezt az alkalmazni kívánt eseményvezérelt szimulátorok és a dolgozatban használt  $\gamma$  processzormodell nem támogatja.

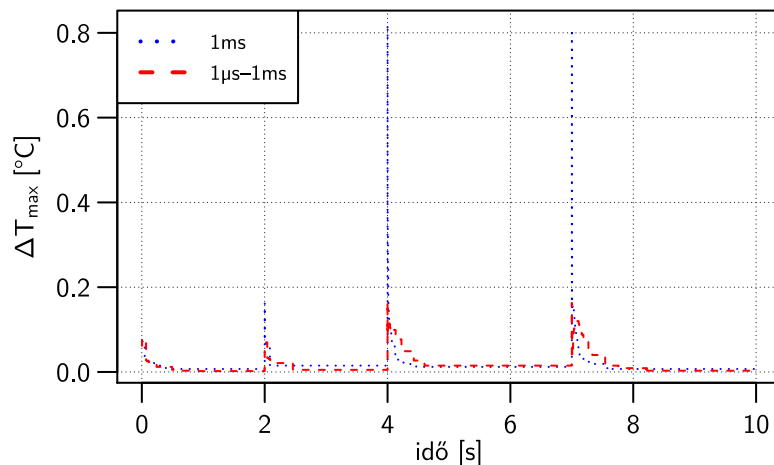
A fenti okok miatt egy olyan algoritmust valósítottam meg, amely a hőmérséklet idő szerinti deriváltja (lásd a (6.16) egyenletet) és az utolsó számítás óta eltelt szimulációs idő alapján megbecsüli a hőmérsékletváltozást. A 6.18. ábrán látható az adaptív időlépést megvalósító algoritmus folyamatábrája. A fő ciklus  $t_e$  szimulációs időközönként értékelődik ki. A  $\Delta t$  változó a legutóbbi hőmérsékletek meghatározása óta eltelt szimulációs időt tárolja,  $\Delta T_{\text{küszöb}}$  és  $\Delta t_{\text{küszöb}}$  változók az algoritmus bemenő paraméterei. Az első paraméter az a küszöbérték, amit ha meghalad a becült hőmérsékletváltozás, akkor a termikus motor meghatározza az új hőmérsékleteket. A második pedig az a szimulációs idő, amely elteltével az új hőmérsékleteket garantáltan meghatározza a



6.18. ábra. Az adaptív időlépést meghatározó algoritmus

szimulátor. Az algoritmus a becsült hőmérsékletváltozást előrelepő Euler módszerrel határozza meg, azaz a becslés a tényleges változásnál mindig nagyobb lesz.

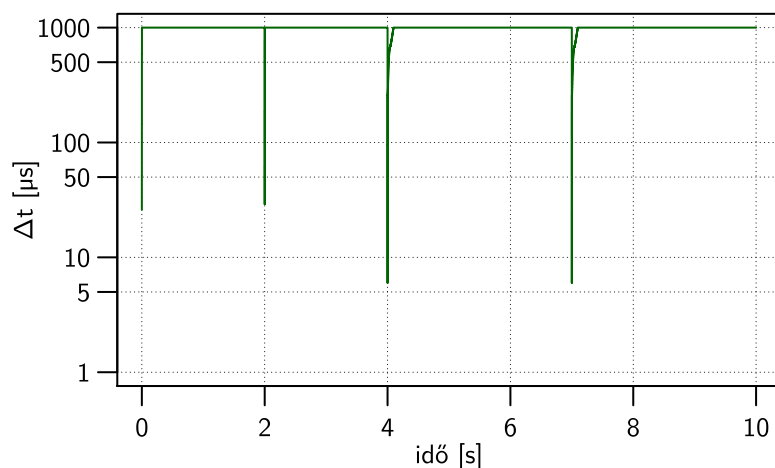
Az 56. oldalon a 6.12. ábrán látható elrendezést és a 6.13(a) ábra szerinti gerjesztést alkalmazva megvizsgáltam a konstans 1 ms és az adaptív – 1  $\mu$ s és 1 ms között változó – időlépés hibáját a referenciának tekintett 1  $\mu$ s esethez képest (6.19 ábra). Az adaptív módszerrel jelentősen csökkenteni lehet a referenciához viszonyított hőmérsékleteltérést.

6.19. ábra. A maximum hőmérsékletek hibája 1  $\mu$ s időlépéshez viszonyítva

A 6.20. ábra illusztrálja a változó időlépést az adaptív algoritmust alkalmazó szimuláció alatt. A gerjesztés megváltozása miatt fellépő hőmérsékletváltozás hatására az algoritmus csökkenti az időlépést, így a lapka hőmérséklete nem változik ugrásszerűen.

A 6.1. táblázat foglalja össze a három szimuláció futási idejét. Az 1 ms időlépést alkalmazva a szimuláció nagyon gyors, azonban a nagy hőmérsékletváltozás hibát okozhat funkcionális-termikus együttes szimuláció során, amennyiben a vizsgálni kívánt jelenség lefutása gyorsabb az időlépésnél. Az 1  $\mu$ s időlépést alkalmazva a hőmérséklet változása egymást követő lépésekben csekély, azonban ennek az ára a százszorosára növekedett futási idő. Az adaptív megközelítés előnye, hogy a gerjesz-





6.20. ábra. Az időlépés változása a szimuláció során

téshez automatikusan igazodik az időlépés, így az állandósult állapot közelében nagy, a tranziens elején kicsi időlépést alkalmaz.

Időlépés	Futási idő	Hiba
1 $\mu$ s	8,77 h	
1 $\mu$ s – 1 ms	1,32 h	0,167 °C
1 ms	5,53 min	0,82 °C

6.1. táblázat. Futási idők és a számított hiba összehasonlítása

A jelenlegi megvalósításban az adaptív időlépték tartományának meghatározása manuálisan történik. A bemutatott példa esetén a tartomány alsó határát a legkisebb időállandó-intenzitás csúcsnál kisebb értéknek választottam. Konkrét logi-termikus vagy funkcionális-termikus szimulációnál az alkalmazott logikai/funkcionális modell granularitásától függ, hogy milyen időállandókat érdemes figyelembe venni. Finom felbontású modell esetén érdemes lehet a legkisebb időállandó alá menni, hogy a termikus szimuláció a lehető legrészletesebb legyen. Durva felbontású modell és nagy időbeli átfogás esetén a legkisebb időállandókat figyelmen kívül lehet hagyni, amennyiben az alkalmazott modell kellően robusztus és nem okoz jelentős hibát.

**III. Tézis.** *Új termikus szimulátort dolgoztam ki, amely modern processzorarchitektúrákon jelentősen gyorsabb a hasonló, jelenlegi megoldásoknál. Az új termikus motor a bemenetként kapott gerjesztések alapján kiszámítja a becsült hőmérsékletváltozást, ami alapján adaptív módon változtatja a termikus szimuláció időlépését. Ez a módszer lehetővé teszi, hogy a gerjesztés bekapcsolásának pillanatában fellépő nagy hőmérsékletváltozást automatikusan detektálja és rövidebb időközönként számítsa ki az áramkör felületén kialakuló hőmérsékleteloszlást, igazodva a kisebb termikus időállandókhöz, míg állandósult állapot közelében nagyobb időközönként számítja ki az új hőmérsékleti értékeket, csökkentve ezzel a funkcionális-termikus szimuláció számítási igényét [S8].*

## 6.5. Az új motor tulajdonságainak összefoglalása

Az általam kifejlesztett új termikus motor egyik előnye a HotSpot, 3D-ICE és SUNRED motorokhoz képest, hogy számottevően kedvezőbb a futási ideje. A ritka mátrixokon végrehajtott műveletek és a lineáris egyenletmegoldó algoritmusok jól párhuzamosíthatóak, így a SloTh képes kihasználni a többmagos CPU-k és GPU-k párhuzamos számítási képességeit.

Az új termikus szimulációs motor a hátralépő Euler és a trapéz módszer alkalmazása esetén iteratív megoldó algoritmust használ a termikus modell egyenletrendszerének megoldásához. Ilyen megoldóalgoritmusok esetén aritmetikai<sup>19</sup> hibák könnyebben előfordulhatnak mint direkt megoldók esetén, de a bemutatott módszer szerint választott leállási feltétel esetén ennek valószínűsége minimalizálható. A SloTh motor konfigurálható egyszeres és dupla pontosságú lebegő pontos számábrázolással, előbbi esetén a szélsőségesen kicsi időlépés és kis felbontás pontatlanságot eredményezhet.

Adaptív időlépés alkalmazásával csökken a szimuláció számítási igénye a konstans szimulációs lépésközhez képest. Akkor érdemes az adaptív időlépés szabályozó algoritmust alkalmazni, ha a vizsgálni kívánt funkcionális-termikus effektusok gyors lefutásúak, és a termikus modell nagy számú csomópontot tartalmaz. A jelenlegi implementációban az időlépés alsó és felső határának megválasztása az időállandóspektrum alapján kézi úton történik.

Az új termikus szimulációs motor jelenleg csak konduktív hőtranszporton alapuló hatásokat tud szimulálni, de a termikus modell alapja azonos a SUNRED termikus modelljével. Az ahhoz kifejlesztett kompakt mikrosatorna-modellek beépíthetőek a SloTh-ba is, így alkalmassá tehető mikrosatornás hűtési megoldásokat alkalmazó 3D elrendezésű IC-k szimulációjára is.

---

<sup>19</sup> Hibásan megválasztott leállási feltétel miatt a megoldó által szolgáltatott eredmény nem éri el a kívánt pontosságot. Tranziens szimuláció esetén ezen az eredményen alapul a következő szimulációs időpontban számított új értékek, így a pontatlanság akkumulálódhat. Ez direkt megoldóalgoritmusok esetén nem probléma, mert a számított eredmény csak a számábrázolásból adódó pontatlansággal terhelt.

## 7. fejezet

# Funkcionális-termikus verifikációs módszer

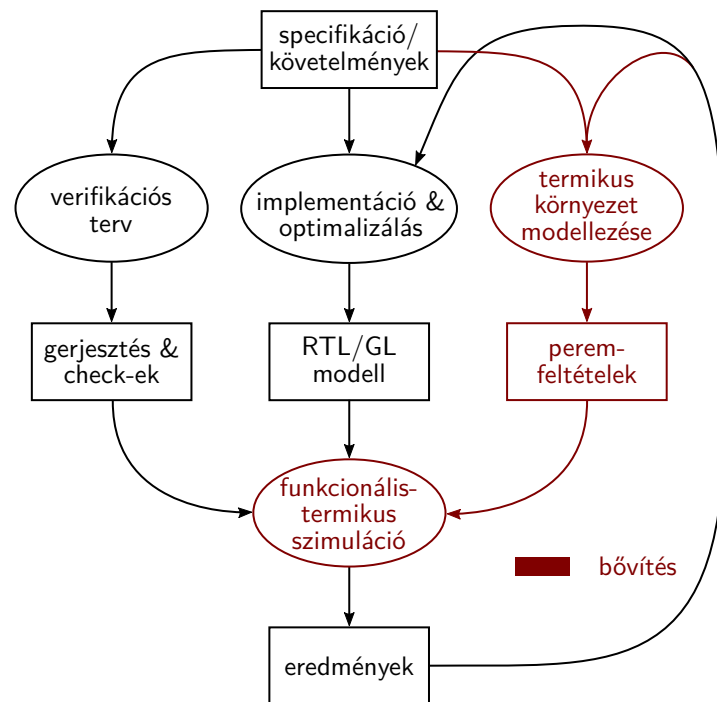
A hagyományos funkcionális verifikáció során a tervezők arra a kérdésre keresik a választ, hogy az áramköri modell működése nem mond-e ellen a specifikációnak, de egy ilyen ellenőrzéssel nem lehet bizonyítani a modell hibátlan voltát. Ezért a verifikációs folyamatot valamilyen önkényesen választott szempont elérésével tekintik befejezettnek. Gyakran a kódlefedésre vonatkozóan fogalmaznak meg követelményeket a verifikáció befejezéséhez. Például ha a kódlefedés eléri 100 %-ot, és nincs ellentmondás a specifikációval, akkor az áramköri modellt megfelelőnek tekintik.

Hagyományos verifikáció során nincs lehetőség termikus effektusok figyelembevételére, mert a jelenlegi modellező és szimulációs eszközök nincsenek erre felkészítve. A 4. fejezetben leírt módszerrel kiegészíthetők a megszokott szimulátorok.

### 7.1. Termikus modellel bővített verifikáció

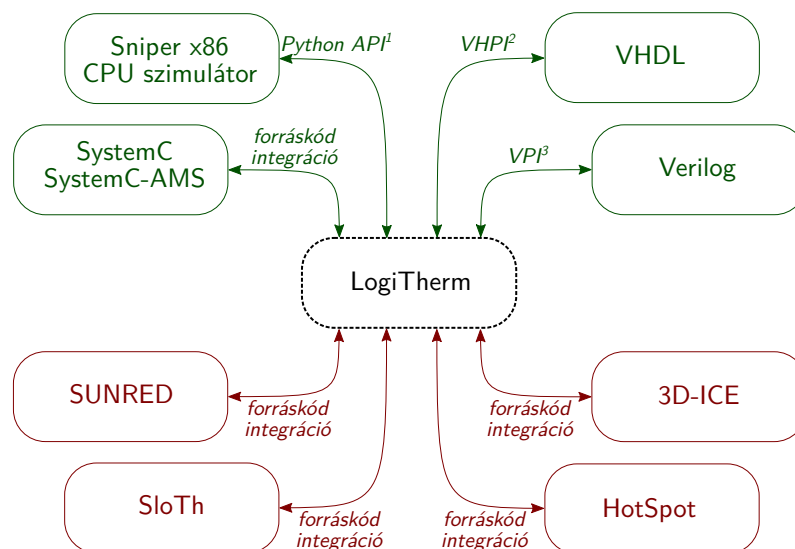
A hagyományos funkcionális verifikáció során a követelmények alapján készülő verifikációs terv és az áramkör tényleges implementációja egymással párhuzamosan készülhet. A szimuláció során az áramkörre a verifikálandó funkció eléréséhez szükséges gerjesztést adva ellenőrzik az áramköri modell működésének helyességét. A 7.1. ábrán látható a termikus szempontokat is figyelembe vevő funkcionális verifikáció folyamata. Az általam javasolt verifikációs eljárásban az áramkör implementálásával párhuzamosan, közvetlenül a követelmények alapján el lehet kezdeni az áramkör termikus környezetének modellezését, a termikus szimulációhoz szükséges peremfeltételek meghatározását.

Funkcionális-termikus verifikáció során a hagyományos verifikációval feltárható hibákon kívül – megfelelően felkészített modellel – lehetséges a hőmérsékletből fakadó funkcionális hibák feltárása, az áramkör hőmérsékletfüggő paramétereinek modellezése, és az azokból adódó viselkedés összevetése a specifikációval. Ilyen jellegű funkcionális hibák digitális áramköröknél adódhatnak a késleltetés hőmérsékletfüggéséből vagy a hőmérséklet alapján döntést hozó vezérlőáramkörök – például DTM illetve DVFS – hibás működéséből. Analóg áramkörök esetén a hőmérsékletfüggés még hangsúlyosabb, annak csökkentése kritikus szempont lehet.



7.1. ábra. Funkcionális verifikáció a termikus viszonyok figyelembevételével

## 7.2. A keretrendszer felépítése és működése



7.2. ábra. A keretrendszerbe beépített és illesztett szimulátorok

A 4. és az 5. fejezetben leírt eredményeket felhasználva funkcionális-termikus verifikációra alkalmas keretrendszert valósítottam meg. Objektumorientált programozási paradigmát használva C++ nyelven olyan őosztályokat implementáltam, amelyek az I. tézisben megfogalmazott absztrakt

<sup>1</sup> API, *Application Programming Interface*

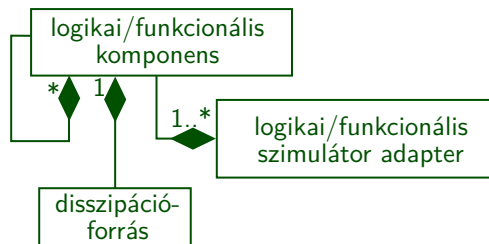
<sup>2</sup> VHPI, *VHDL Procedural Interface*

<sup>3</sup> VPI, *Verilog Procedural Interface*

interfészt realizálják. Funkcionális/logikai, vagy termikus szimulációs motor beépítéséhez ezeket az őszosztályokat kell kibővíteni az adott szimulátor igényének megfelelően.

A 7.2. ábrán láthatóak a keretrendszerhez kapcsolt modellező nyelvek és termikus szimulációs motorok. A LogiTherm keretrendszer által nyújtott interfész általános, így a funkcionális/logikai és termikus szimulátorok tetszőleges kombinációja használható.

### 7.2.1. Logikai/funkcionális szimulátorok adaptere



7.3. ábra. A logikai/funkcionális szimulátorok illesztéséhez használt típusok UML<sup>4</sup> osztálydiagramja

A 7.3. ábrán láthatóak a logikai/funkcionális szimulátorokhoz, modellező nyelvekhez tervezett interfészt alkotó osztályok UML diagramja. A hierarchiába szervezhető logikai/funkcionális komponenshez egy disszipációforrás ( $p_i$ ) tartozik, és megfelel a 4. fejezetben leírt funkcionálismodell-komponensnek ( $F_i$ ). Látható, hogy a megvalósított keretrendszer is általános fogalomként kezeli a logikai/funkcionális komponenset, így az 5. fejezetben leírt leképezésen kívül más megfeleltetést is meg lehetne valósítani.

Az adapter feladata, hogy a relaxációs módszerrel megvalósított együttes szimuláció során bizonyos szimulációs időközönként meghívja a keretrendszer függvényeit, valamint adminisztrálja a logikai/funkcionális komponenseket. A keretrendszer függvényeit a szimulátor motorokba bejegyezhető callback függvények hívják meg az adapter osztályon keresztül.

### SystemC és SystemC-AMS integráció

A SystemC és SystemC-AMS modellező nyelvek C++ nyelvű osztályok, függvények és makrók gyűjteményei [110, 111], amelyeknek egy implementációja letölthető az Accellera honlapjáról [112]. A SystemC-t és SystemC-AMS-t integráltam a keretrendszerbe, így nem szükséges külön telepíteni őket. Egy SystemC nyelvű modellhez tartozó disszipációmodellt a forráskódba építve, vagy a modell együttműködését tartalmazó CSV<sup>5</sup> fájlban lehet megadni.

### VPI interfész beépítése

A Verilog hardverleíró nyelven implementált modelleket a VPI interfészen [113] keresztül éri el a LogiTherm keretrendszer. A VPI-t támogató szimulátorokban – például Mentor<sup>®</sup> Questa<sup>™</sup> Simben és Cadence<sup>®</sup> IES-ben – az áramköri modell betöltésekor kell megadni a keretrendszer binárisát és az úgynevezett bootstrap függvényt<sup>6</sup> az együttes szimuláció futtatásához. A disszipációmodell származhat a Liberty<sup>™</sup> adatbázisból, vagy kapusztintnél magasabb absztrakció esetén CSV fájlból.

<sup>4</sup> UML, *Unified Modeling Language*

<sup>5</sup> CSV, *Comma Separated Value*

<sup>6</sup> A bootstrap függvény regisztrálja a LogiTherm keretrendszer függvényeit a szimulátor programba.

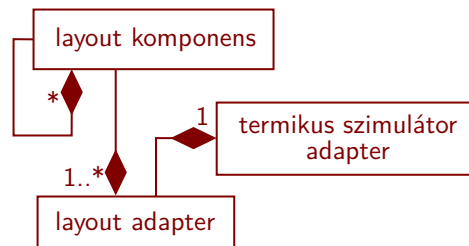
## VHPI interfész megvalósítása

VHDL nyelvű hardvermodellek funkcionális-termikus szimulációja a VHPI interfész [114] segítségével lehetséges. A VHPI-t támogató Cadence® IES szimulátorprogramban hasonlóan a Verilog szimulációhoz az áramköri modell betöltésekor meg kell adni a LogiTherm keretrendszer binárisát és bootstrap függvényét. A disszipációmódellet együttthathatóit CSV fájlban lehet megadni.

## A Sniper x86 processzor szimulátor illesztése Python API-val

A LogiTherm keretrendszerhez illesztettem a Sniper x86 processzor szimulátort [94] is. A szimulátor x86 utasításkészletű processzorok szimulációjához használható, a vizsgálni kívánt architektúra paraméterei konfigurálhatóak. A Sniper része egy Python API, amelyen keresztül hozzá lehet férni a szimulált processzorhoz, ezért a LogiTherm keretrendszert Python nyelvű interfésszel bővítettem ki. Így a funkcionális-termikus együttes vizsgálatot a Sniper Python API-ján keresztül valósítottam meg.

### 7.2.2. Termikus szimulátorok adaptore



7.4. ábra. A termikus szimulátorok illesztéséhez használt típusok UML osztálydiagramja

A 7.4. ábrán látható a termikus szimulátorok illesztéséhez használt osztályok UML osztálydiagramja. A layout komponens hierarchiába szervezhető, a 4. fejezetben leírt layout alakzatnak felel meg ( $L_i$ ). A layout adapter adminisztrálja a layout komponenseket, tárolja azok fizikai elrendezését. A termikus szimulátor adaptore a termikus szimuláció paramétereinek kezeléséért felelős. Az együttes szimuláció időlépéseiben ennek az adapternek a függvénye hívódik meg, ami a termikus szimulátor megfelelő függvényeit hívja meg a hőmérsékletek kiszámításához.

## A 3D-ICE beépítése

A 3D-ICE szimulátor forráskódja nyilvánosan hozzáférhető [42], így azt integráltam a LogiTherm keretrendszerbe. A layout komponensek elrendezése és a termikus szimuláció paraméterei a szimulátor dokumentációjában meghatározott formátumú fájlban adhatóak meg. A gyors szimuláció érdekében a keretrendszerhez használható az OpenBLAS és az Intel® MKL könyvtár is.

## A HotSpot beépítése

A HotSpot termikus szimulátor forráskódja letölthető [35], így azt is beépítettem a LogiTherm keretrendszerbe. A layout komponensek elrendezése és a termikus szimuláció paraméterei egyedi fájlformátumban adhatóak meg a szimulátor számára. A HotSpot is a SuperLU matematikai szubrutincsomagot használja a termikus modell egyenleteinek megoldásához, így a korábban említett

OpenBLAS és az Intel® MKL könyvtárak használata ennél a termikus szimulátornál is csökkenti a szimuláció futási idejét. Lényeges, hogy a LogiTherm keretrendszer általános termikus szimulátor adaptere nem határoz meg követelményeket a termikus modell jellegére vonatkozóan, így használható a HotSpot kompakt és numerikus modellje is.

## A SUNRED beépítése

A SUNRED termikus szimulátor forráskódját Dr. Pohl László biztosította számomra. A layout elrendezést és a termikus szimuláció paramétereit általam specifikált formátumú fájlokban lehet megadni, de SystemC és SystemC-AMS modellek esetén megfelelő függvényhívásokkal a forráskódban is konfigurálható a motor. Az új fejlesztésű mikrocsatorna modellt és a csatorna elrendezését a SUNRED saját formátuma szerint kell meghatározni [S1].

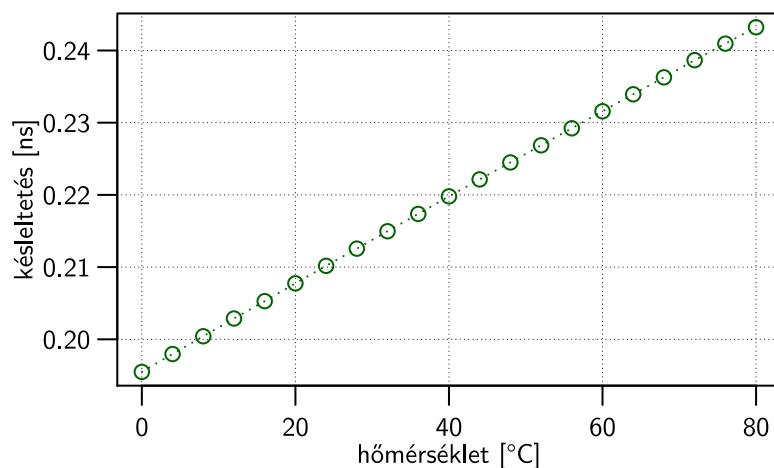
## 7.3. Szimulációs eredmények

A funkcionális-termikus verifikációs módszer lehetőségeit a három demonstrációs rendszeren keresztül mutatom be. A logikai kapuk késleltetésének hőmérsékletfüggése az áramkör működése közben megváltoztathatja az időzítési viszonyokat. Kapusinten korábban is lehetséges volt a hőmérséklet figyelembe vétele, azonban ez adott esetben nagyon lassú lehet. Az általam javasolt módszer lehetővé teszi az áramkör modellezését RTL szinten is, így felgyorsítva a szimulációt. Az így kapott eredményt a referenciának tekintett kapusintű eredmények alapján validáltam.

Modern SoC lapkák digitális részarámkörök mellett tartalmaznak kevert jelű blokkokat is, amelyek érzékenyek lehetnek a hőmérsékletre. A második példarendszer ADC és DAC áramköröket tartalmaz, amelyek pontatlanná válnak a hőmérséklet változásával, így okozva funkcionális hibát.

A harmadik processzoros rendszerben hőmérséklet-menedzsment algoritmus példáján keresztül mutatom be az algoritmusszintű funkcionális-termikus verifikáció lehetőségeit.

### 7.3.1. Hőmérsékletfüggő késleltetés modellezése



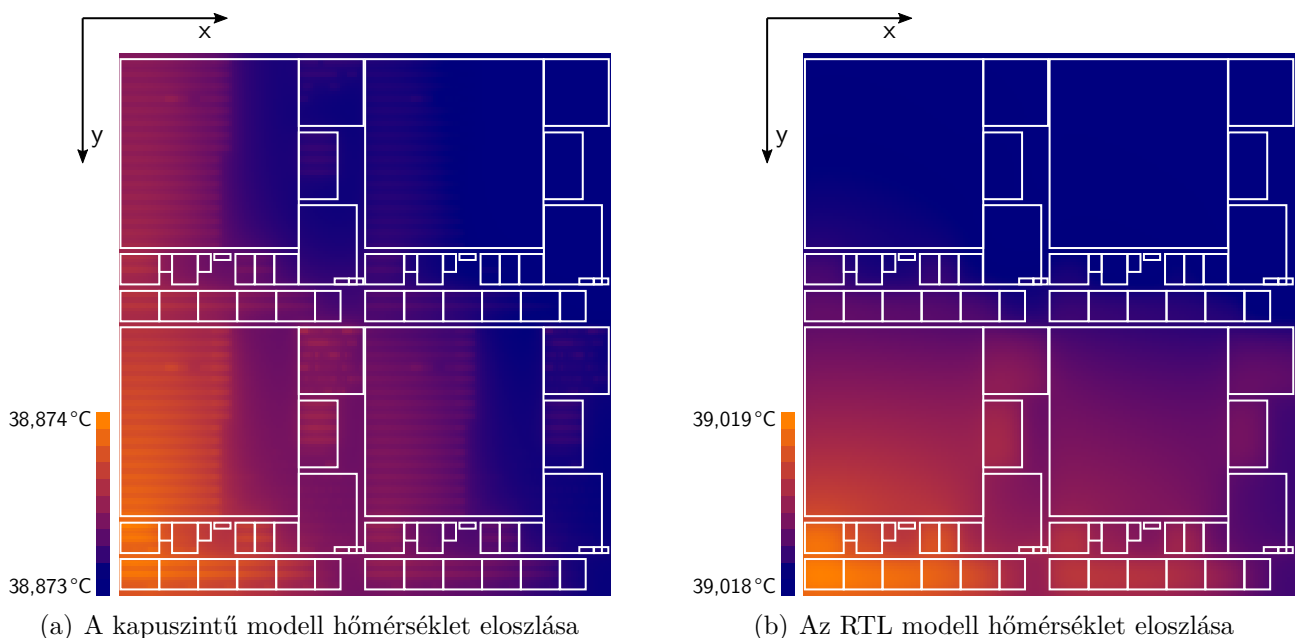
7.5. ábra. NAND20 kapu hőmérsékletfüggő késleltetése

Hagyományos PDK-ban nem áll rendelkezésre a standard cellák hőmérsékletfüggő késleltetésének adatbázisa, azonban előállítható sorozatos SPICE szimulációkkal. Timár András doktori munkája során részletesen foglalkozott ezzel a módszerrel [8]. Ehhez a munkához hasonlóan SPICE

szimulációk segítségével határoztam meg az egyes standard cellák hőmérsékletfüggő késleltetését. A 7.5. ábra az AMS 0,35  $\mu\text{m}$ -es CMOS technológiáján megvalósított NAND20 cella késleltetését mutatja be a hőmérséklet függvényében.

A logi-termikus szimuláció az áramkör működése során a változó hőmérsékletet is figyelembe veszi minden kapu késleltetésének meghatározásakor, de a teljes rendszert kapuszintra kell ehhez szimulálni. A dolgozatban bemutatott új módszer alkalmas kevert elvonatkoztatású modellek funkcionális-termikus szimulációjára, így tetszőlegesen kombinációban használhatóak kapuszintra és magasabb – például RTL – elvonatkoztatású tervezési egységek. Hőmérsékletfüggő késleltetés modellezése adott esetben elegendő lehet azoknál a komponenseknél, ahol az időzítés kritikus.

A hőmérsékletfüggő késleltetés modellezésének demonstrálásához, és a magasabb elvonatkoztatású modell előnyeinek bemutatásához az 5.2.1. és az 5.2.2. fejezetekben bemutatott  $\alpha$  processzormagot használtam fel. Az 5.6. ábrán látható elrendezésű processzorból négyet helyeztem el, amelyek közül az alsó kettőt futtatott programot. A szilícium lapka mérete így  $1,06 \text{ mm} \times 1,15 \text{ mm}$ -nek adódott, a vastagságát  $100 \mu\text{m}$ -nek választottam, termikus peremfeltételnek a lapka alján a levegő természetes konvekciója alapján  $10 \text{ W/m}^2 \text{ K}$  hőátadási tényezőt alkalmaztam.

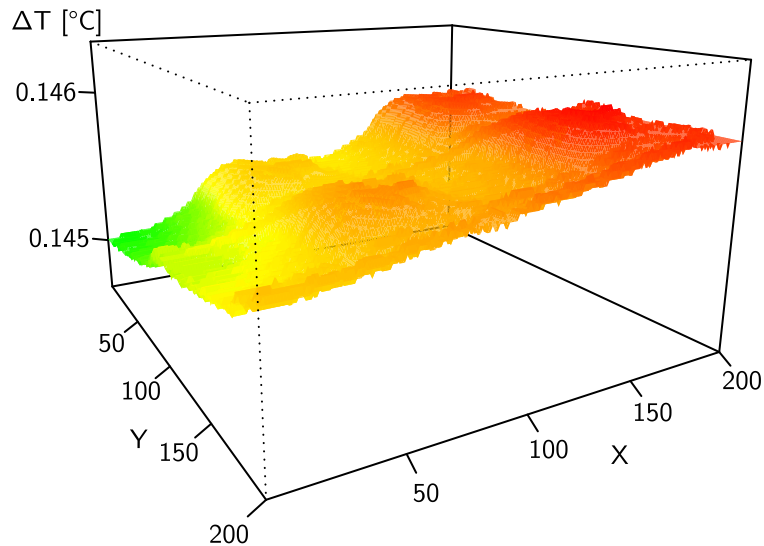


7.6. ábra. A szimuláció végén kialakult hőmérsékleteloszlások

A 7.6(a) ábrán a referenciának tekintett kapuszintra modell, a 7.6(b) ábrán pedig az RTL modell hőmérséklet eloszlása látható. A lapkák felületén kismértékű hőmérsékletkülönbség tapasztalható, ezért a lapkahőmérséklet gyakorlatilag konstansnak tekinthető. Ezt a processzor kis fogyasztása és a kis disszipációgradiens, valamint a lapka vastagságához képest kis laterális mérete okozza. A 7.7. ábrán a két hőtérkép közötti abszolút eltérés látható, amit az RTL disszipációmodell hibája okoz. A szimuláció végén számított  $0,146 \text{ }^\circ\text{C}$ -os eltérés  $1,3 \%$ -os relatív hibát jelent<sup>7</sup>. A kismértékű eltérés lehetővé teszi a két elvonatkoztatási szint keverését, hogy csak azokat a komponenseket kelljen kapuszintra vizsgálni, ahol a standard cellák hőmérsékletfüggő késleltetése szükséges.

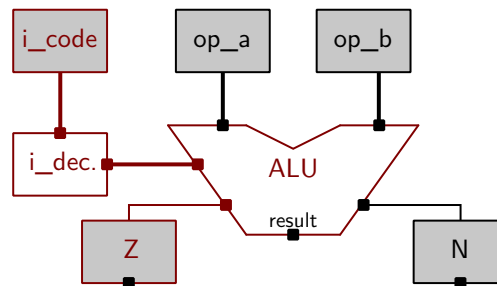
<sup>7</sup> A 41. oldalon az 5.3.2. fejezetben az RTL modell fogyasztása kisebbnek adódott a kapuszintra modellhez képest. Az eltérést a különböző mértékű kapcsolási aktivitás okozza, amely az 5.3.2. fejezetben alkalmazott *zero-delay* szimuláció, és ebben a fejezetben használt hőmérsékletfüggő késleltetés modellezésekor tapasztalható.





7.7. ábra. A kapuszintű és RTL modell hőmérséklet eloszlása közötti abszolút hiba

Szinkron hálózatokban a lokális adatutakat a jelterjedés szempontjából szomszédos regiszterek, és az azokat összekötő kombinációs hálózatok alkotják. A kritikus út az a lokális adatút, amelynek a legnagyobb a késleltetése. Amennyiben erre teljesülnek az időzítési feltételek, akkor az összes többi adatútra is teljesülnek.

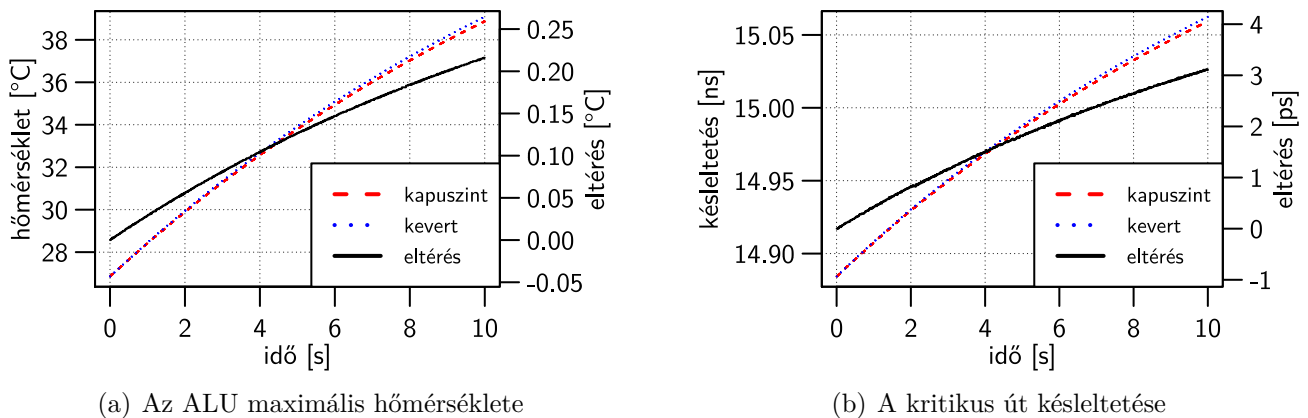


7.8. ábra. A processzor kritikus – legnagyobb késleltetésű – jelútja

A 7.8. ábrán pirossal kiemelve látható az  $\alpha$  processzormag kritikus útja, amelyet statikus időzítés-analízissel határoztam meg. A legnagyobb késleltetésű adatút az utasításregisztertől indul, a zero flag-ben ér véget, a késleltetés legnagyobb részét pedig az ALU-ban található *ripple carry* lánc adja. A kapuszintű és RTL modellek szimulációja során tapasztalt hőmérséklet közötti kis különbség lehetővé teszi olyan kevert elvonatkoztatású modell vizsgálatát, amelyben a kritikus út kapuszinthen, a processzor többi eleme RTL szinten van implementálva.

A 7.9(a) ábrán látható az ALU maximális hőmérséklete a kapuszintű és a kevert elvonatkoztatású modell funkcionális-termikus szimulációja során. A legnagyobb,  $0,2^{\circ}\text{C}$ -os eltérés a szimuláció végén alakult ki a disszipációmodell pontatlansága miatt. A szimuláció során a hőmérséklet emelkedésével nőtt a kritikus út késleltetése is, ami a 7.9(b) ábrán látható.

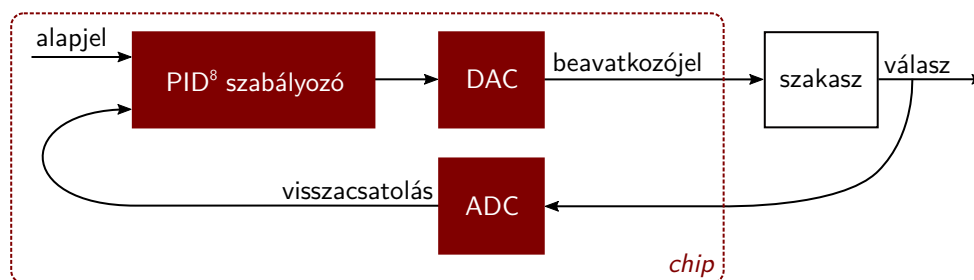
A két szimuláció futási ideje egy nagyságrendben tért el egymástól – a kevert modell esetén 42 perc volt, egységesen kapuszintű modellt alkalmazva pedig 8 óra 20 percre adódott. Eközben a hőmérséklet és a számított késleltetés relatív hibája 2% volt a referenciának tekintett kapuszintű logi-termikus szimulációhoz képest.



7.9. ábra. Az ALU hőmérséklete és a kritikus út késleltetése a kapusztintú és a kevert modell esetén

### 7.3.2. Kevert jelű rendszer hőmérsékletfüggő viselkedése

Az analóg és a kevert jelű áramkörök tipikusan érzékenyebbek a hőmérsékletre, mint a digitális áramkörök. Az SoC lapkákba egyre több, nem kizárólag digitális részegységet integrálnak, így szükség lehet a teljes chip (nem csak az adott részáramkör) funkcionális-termikus együttes szimulációjára.



7.10. ábra. A processzoros rendszer által megvalósított szabályozási rendszer

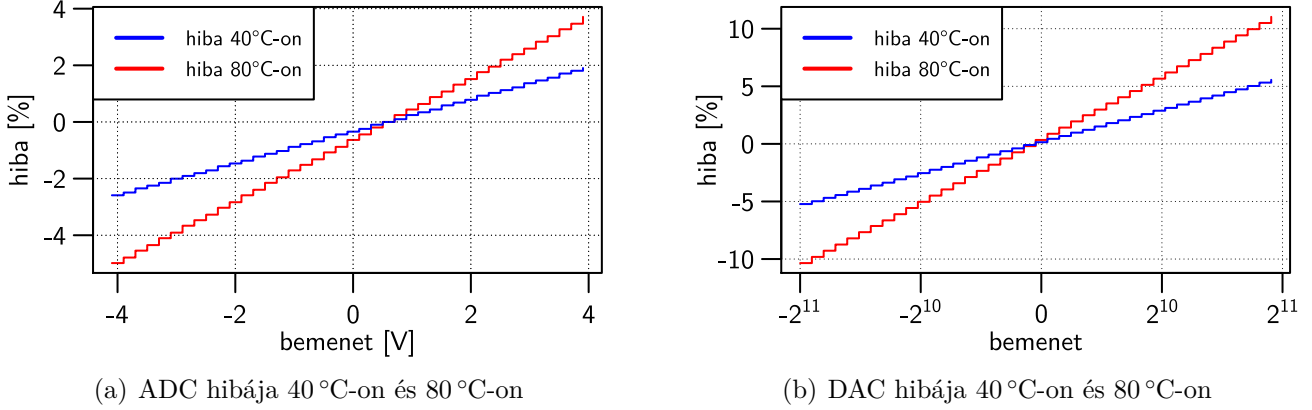
A kevert jelű áramkörök funkcionális-termikus együttes szimulációjának demonstrálása céljából a 7.10. ábrán látható PID szabályozó rendszert valósítottam meg a  $\beta$  processzormag felhasználásával. Az alapjel a processzor egyik regiszterében beállított érték, a szabályozó a szakaszt egy analóg beavatkozójellel szabályozza, a visszacsatolt válaszjelet pedig egy ADC-n keresztül olvassa be a processzor és képez belőle egy hibajelet. A processzormag tanszéki fejlesztésű, így különböző elvonatkoztatású (kapusztintú, RTL) és nyelvű (SystemC, VHDL, Verilog) implementációi is rendelkezésre álltak. A rendszerben található ADC és DAC áramkörök modellezéséhez a TUV AMS könyvtár [115] SystemC-AMS nyelven implementált komponenseit használtam.

drift paraméter	ADC	DAC
Erősítés	400 ppm /°C	-400 ppm /°C
Referencia	1000 ppm /°C	-1000 ppm /°C
Ofszet	300 ppm /°C	-300 ppm /°C

7.1. táblázat. Az ADC és DAC hőmérsékletfüggő paramétere

<sup>8</sup> PID, *Proportional Integral Derivative*

A vizsgált rendszerben a kevert jelű komponensek (ADC és DAC) hőmérsékletfüggését a 7.1. táblázatban látható paraméterekkel modelleztem. A 7.11 ábrán látható az ADC és DAC a karakterisztikák hibája 40 °C-on és 80 °C-on a drift paraméterek figyelembevételével (hőmérsékletfüggetlen hibát nem modelleztem).



7.11. ábra. Az ADC és DAC hibája a hőmérséklet függvényében

A folytonos idejű szabályozott szakaszt harmadrendű átviteli függvénnyel modelleztem:

$$W_{szakasz}(s) = \frac{A_{szakasz}}{(1 + sT_1)(1 + sT_2)(1 + sT_3)}. \quad (7.1)$$

A szakasz paramétereinek a 7.2. táblázatban látható értékeket választottam.

$A_{szakasz}$	$T_1$	$T_2$	$T_3$
5	500 ms	200 ms	50 ms

7.2. táblázat. A szabályozott szakasz paramétereinek

A PID szabályozót a következő egyenlet írja le:

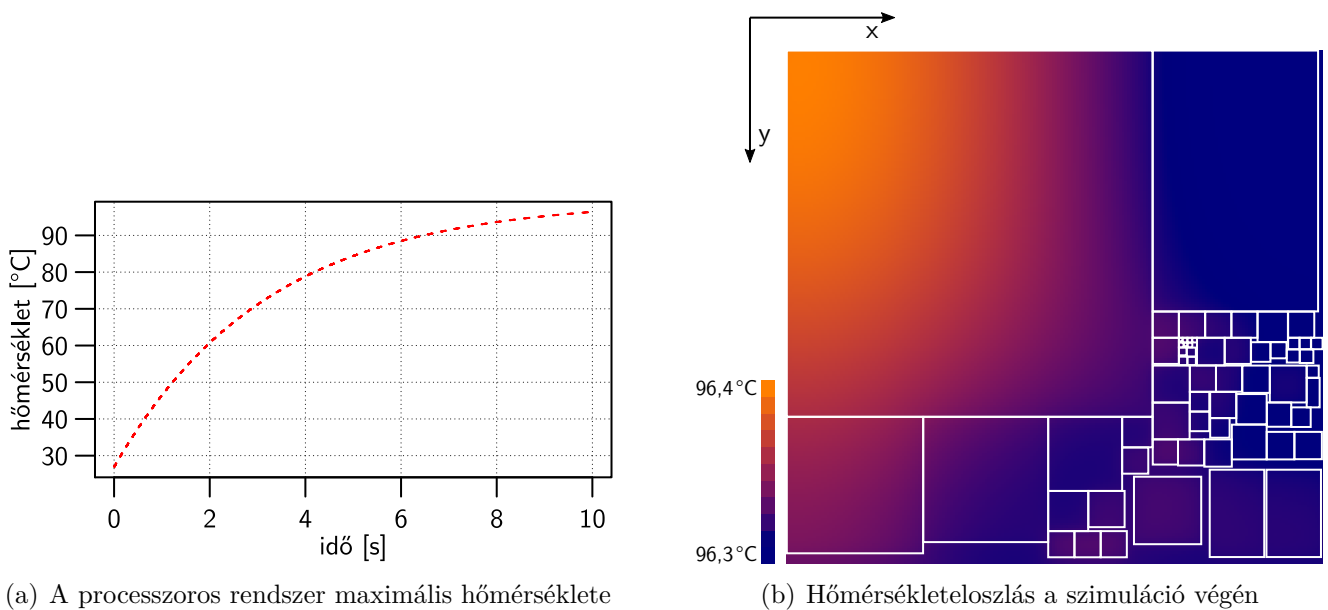
$$W_{PID}(s) = A_{PID} \left( 1 + \frac{1}{sT_I} + sT_D \right). \quad (7.2)$$

A szabályozót úgy terveztem meg, hogy a nyílt hurok átviteli függvényében kiessen a szakasz két legnagyobb időállandója és a fázistartalék 60° legyen. Az így meghatározott paraméterek a 7.3. táblázatban láthatóak.

$A_{PID}$	$T_I$	$T_D$
2	700 ms	142,86 ms

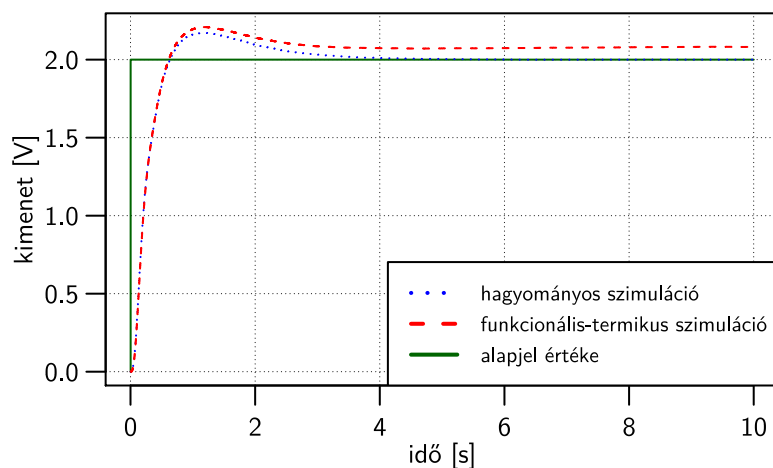
7.3. táblázat. A PID szabályozó paramétereinek

A layout komponensek elrendezése a 40. oldalon az 5.7. ábrán látható. A lapka laterális mérete 2,46 mm × 2,352 mm, a vastagsága 200 μm. A funkcionális-termikus szimulációk során 100 W/m<sup>2</sup> K hőátadási tényezőt alkalmaztam a chip alsó oldalán, a többi oldalon adiabatikus peremfeltételt adtam meg. A megadott hőátadási tényező a lapkánál tízszer nagyobb felületű, természetes konvekció alapuló hűtőborda hatásának felel meg.



7.12. ábra. A lapka maximális hőmérséklete és a hőmérsékleteloszlás a szimuláció végén

A 7.12(a) ábrán látható a  $\beta$  processzormag lapka maximális hőmérséklete a szimuláció során, ami a kezdeti  $26\text{ }^{\circ}\text{C}$ -ról a disszipáció következtében  $100\text{ }^{\circ}\text{C}$ -ig emelkedik. A 7.12(b) ábrán látható a hőmérsékleteloszlás a szimuláció végén. A lapkán kialakuló  $0,1\text{ }^{\circ}\text{C}$ -os hőmérsékletkülönbség nagyobb, mint az  $\alpha$  processzor esetén, azonban ez is kis értéknek számít. A szimuláció során a processzor  $\sim 40\text{ mW}$ -ot disszipál, ami kis disszipációgradienst jelent, így a jelentős hőmérsékletkülönbség sem alakul ki.



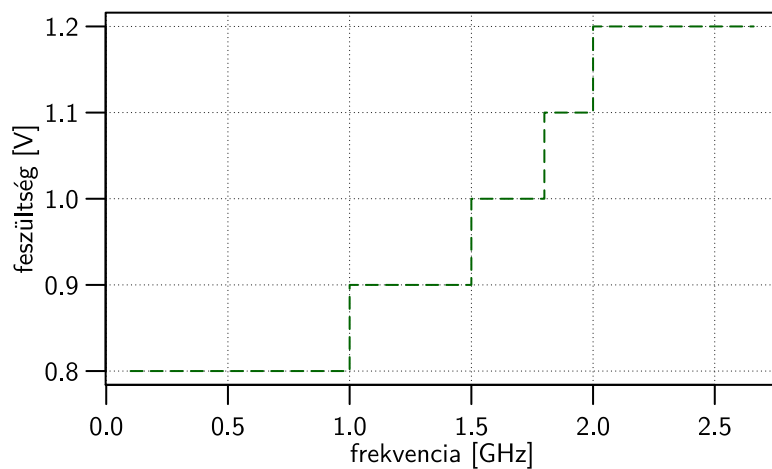
7.13. ábra. A PID szabályozó ugrásválasza

A 7.13. ábrán a szakasz ugrásválasza látható hagyományos szimuláció során a hőmérsékletfüggő paraméterek figyelembe vétele nélkül, valamint funkcionális-termikus szimuláció során. Ez utóbbi módszerrel a PID szabályozók esetén nem megszokott maradó hiba is vizsgálhatóvá vált, amelyet a hőmérsékletre érzékeny kevert jelű áramkörök okoztak.

### 7.3.3. Dinamikus hőmérséklet-menedzsment vizsgálata

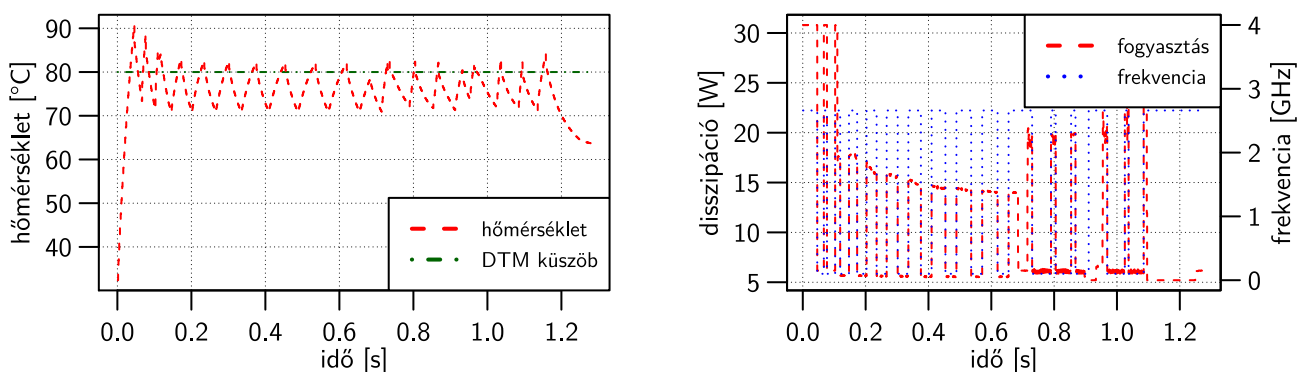
Modern, többmagos processzorok működés közben a terhelés alapján képesek állítani a frekvenciájukat – akár az egyes magokat külön-külön. Így csökkenthető a fogyasztás kisebb terhelés mellett, valamint megakadályozható az áramköri lapka esetleges túlmelegedése.

Egy ilyen funkciójú hőmérséklet-menedzsment algoritmust valósítottam meg a  $\gamma$  processzor felhasználásával. A processzor layoutja a 41. oldalon az 5.8. ábrán látható, a lapka mérete  $17,2\text{ mm} \times 13,2\text{ mm}$ , a vastagsága  $500\text{ }\mu\text{m}$ . A disszipáció becslése a szimulációk során az 5.2.2. fejezetben leírt módon a McPAT modell felhasználásával történt. Funkcionális-termikus szimulációk során  $5500\text{ W/m}^2\text{ K}$  hőátadási együtthatót alkalmaztam a lapka alján (az aktív elemek a lapka tetején helyezkednek el), a többi oldalon adiabatikus peremfeltételt adtam meg. Ez a hőátadási együttható az alkalmazott lapkaméretek esetén körülbelül  $\sim 0,8\text{ K/W}$  hőellenállásnak felel meg, ami  $100\text{ W}$  disszipáció esetén  $80\text{ }^\circ\text{C}$  hőmérsékletemelkedést okozna.



7.14. ábra. Az alkalmazott feszültségek a frekvencia függvényében

Az implementált DTM algoritmus minden mag hőmérsékletét monitorozza, túlmelegedés ( $T > 80\text{ }^\circ\text{C}$ ) esetén pedig visszaveszi az adott mag órajelét  $100\text{ MHz}$ -re, a tápfeszültséget eközben a 7.14. ábrán látható összefüggés<sup>9</sup> szerint választja meg.



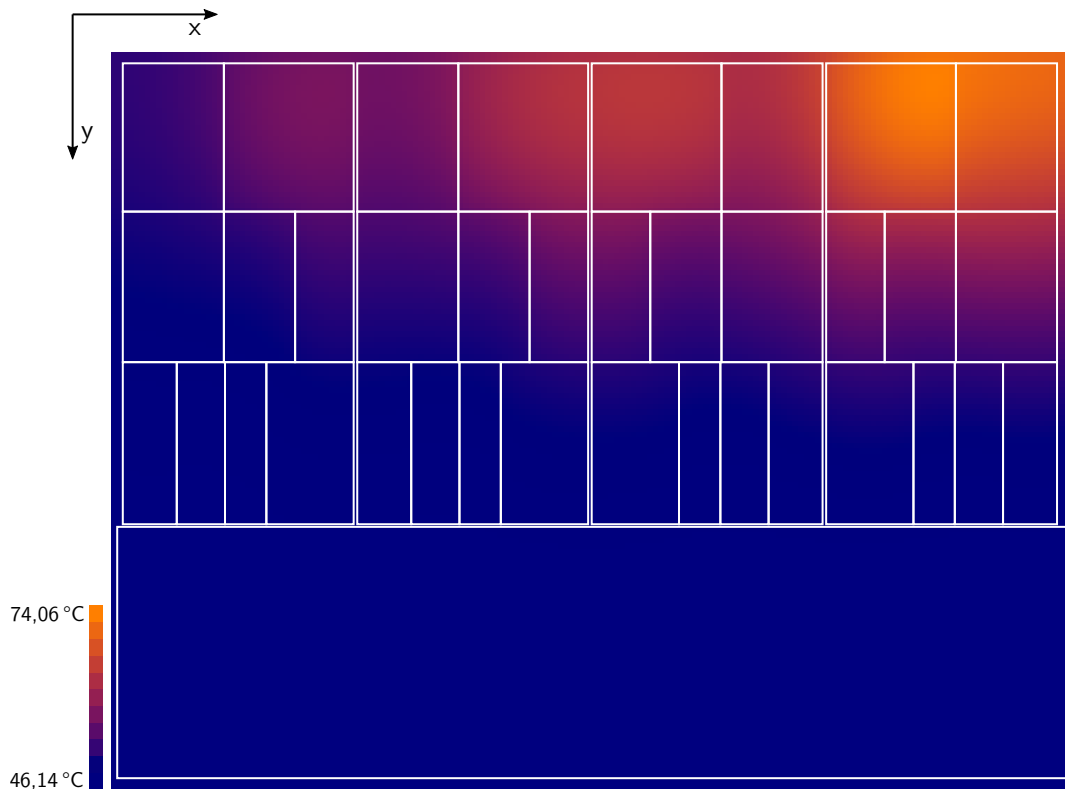
(a) Az első mag hőmérséklete

(b) Az első mag fogyasztása és frekvenciája

7.15. ábra. Dinamikus hőmérséklet-menedzsment (DTM) működés közben

<sup>9</sup> A tápfeszültség és az órajel frekvencia közötti kapcsolatot a Sniper szimulátor készítői határozták meg katalógusadatuk alapján.

A 7.15(a) ábra az első mag hőmérsékletét, a 7.15(b) ábra pedig az első mag frekvenciáját és disszipációját mutatja be szimuláció közben. Megfigyelhető, hogy miután a hőmérséklet meghaladta a 80 °C-os küszöböt, a DTM hatására a mag órajele és emiatt a fogyasztása is lecsökken. A szimuláció végén kialakuló hőmérsékleteloszlás a 7.16. ábrán látható. A lapkán tapasztalt hőmérsékletkülönbség a korábbi példákkal szemben már jelentős, a kapott eredmény kvalitatíve összehasonlítható a 2.6. ábrán látható mérési eredménnyel. Ezzel a szimulátor konfigurációval kifinomult DTM algoritmusokat is ki lehet értékelni, amelyek az egyes magok hőmérsékletét is figyelembe veszik a feladatok ütemezésekor.



7.16. ábra. A lapkán kialakult hőmérséklet-eloszlás a szimuláció végén

Algoritmusszintű modellek funkcionális-termikus szimulációja nagyobb időbeli átfogást tesz lehetővé, mint ami regiszterátviteli szinten vagy a kapuszintra elérhető. Ezáltal összetett processzorok működése is vizsgálható a termikus időállandók nagyságrendjében.

**IV. Tézis.** *Az I. tézisben megfogalmazott absztrakt interfész általános jellegét felhasználva módszert dolgoztam ki digitális, valamint analóg és kevert jelű (mixed-signal) áramkörök funkcionális-termikus együttes verifikációjára. Az új eljárással az áramkör viselkedését befolyásoló hőmérsékletfüggő paraméterek is figyelembe vehetőek a verifikáció során, amelyhez az áramkör termikus környezetének modelljén túl a hagyományos funkcionális verifikáció során alkalmazott eszközök használhatóak [S6, S9].*

## 7.4. A bővített verifikációs módszer összefoglalása

### A szimulációs eredmények jelentősége

A funkcionális-termikus verifikáció lehetőségeit három példarendszeren keresztül mutattam be. Az  $\alpha$  processzorban a logikai kapuk hőmérsékletfüggését modelleztem. Korábbi logi-termikus szimulációs elven működő szimulátorokkal szemben az új módszerben lehetőség nyílik kevert elvonatkoztatású modell vizsgálatára, amely jelentősen csökkenti a szimuláció futásidejét. A szimuláció közben a növekedő hőmérséklet hatására a kapuk késleltetése is nőtt, amit a referenciának tekintett kapusintű szimulációhoz hasonlítottam.

A  $\beta$  kevert jelű rendszerben PID szabályozót valósítottam meg. A szabályozókör pontatlanságát a hőmérsékletfüggő ADC és DAC modulok okozzák, amelyek miatt PID szabályozóknál nem megszokott maradó hiba is fellépett.

A  $\gamma$  processzorral bemutattam, hogyan használható a szimulációs módszer a hőmérsékletmenedzsment algoritmusok, és a processzor hűtési megoldásainak kiértékeléséhez. Az alkalmazott hűtés nem bizonyult elegendőnek ahhoz, hogy a disszipáló hőt elvezesse, így a DTM visszavette a túlmelegedő mag órajelét.

Az eredmények alapján a bővített verifikációs módszer olyan modellek verifikációjára használható, amelyeknél a hőmérséklet erősen befolyásolja az áramkör működését.

### A keretrendszer előnyei

A megvalósított keretrendszer alkalmas különböző elvonatkoztatású modellek funkcionális és termikus viselkedésének együttes vizsgálatára. Az alkalmazott módszer (4. fejezet) általános jellegű, így különböző modellező nyelveken implementált modelleket is fogadni tud, a későbbiekben pedig a keretrendszer tovább bővíthető például Verilog-AMS<sup>10</sup>, VHDL-AMS [116] és SystemVerilog [117] támogatásával. A 2.4.2. fejezetben bemutatott együttes szimulációs megoldások csak egy modellező nyelvet támogatnak, vagy fix modell könyvtárat használnak.

### A keretrendszer korlátai

A disszertációban bemutatott módszer, és az azon alapuló keretrendszer nem oldja meg az egyes áramköri tulajdonságok – például a disszipáció vagy layout – modellezésének nehézségeit. A szakirodalom részletesen foglalkozik a digitális integrált áramkörök fogyasztásának modellezésével illetve a területigény becslésével, amely eredmények felhasználhatóak az általam megvalósított keretrendszerben is.

<sup>10</sup> Az IEEE-nek nincs szabványa a Verilog analóg és kevert jelű kiegészítésére, de az Accellera által bővített nyelvet [118] a Cadence® IES részben támogatja.

A funkcionális-termikus szimulációs keretrendszer használata során figyelni kell az alkalmazott disszipációmodell pontosságára, mert annak közvetlen hatása van a termikus szimuláció eredményére. A funkcionális modell granularitása ugyancsak hatással van az eredményekre. Durva felbontású funkcionális modell kisszámú disszipációforrást eredményez, ami csökkenti a termikus szimuláció részletességét is, azonban a szimuláció számításigénye is csökken.

### A keretrendszer validációjának kérdése

Az értekezésben bemutatott funkcionális-termikus szimulációs módszer és az azt alkalmazó keretrendszer egy lépésben csak másik együttes szimulációs módszerrel validálható. Nincs tudomásom más olyan általános módszerről, amely tetszőleges elvonatkoztatási szintet és több modellező nyelvet is támogat. Emiatt a már létező megoldásokkal az új eljárást csak részben lehet validálni. Az új módszerben a kapusintű együttes szimuláció megegyezik a logi-termikus szimulációs eljárással, amelyet mérésekkel is validáltak [73].

A keretrendszerben alkalmazott modellek és szimulátorok egyesével validálhatóak más, hasonló megoldások alapján. A HotSpot, 3D-ICE és SUNRED termikus motorokat validálták a készítői, a disszertációban bemutatott SloTh szimulátort az utóbbi kettő szimulátorral vetettem össze. Az RTL modellek esetén alkalmazott disszipációmodellt a kapusinten számolt fogyasztással karakterizáltam, és a pontosságát is ahhoz hasonlítottam. A standard cellák fogyasztását a félvezetőgyár – jelen esetben az AMS – szolgáltatja. A gyártó a PDK-ban biztosított tranzisztor és standard cella modelleket tesztáramkörökön végzett mérések alapján határozza meg. A  $\gamma$  processzor esetén alkalmazott McPAT disszipációs modellt a Sniper szimulátor készítői valós processzoron végzett mérések alapján kalibrálták és validálták [13].

A hőmérsékletfüggő paraméterek modellezése szabványok hiányában a szimulációs módszer felhasználójára van bízva. A 7.3.1. fejezetben felhasznált hőmérsékletfüggő késleltetéseket Timár András korábbi munkája alapján határoztam meg [8]. A 7.3.2. fejezetben alkalmazott ADC és DAC modulok hőmérsékletfüggő működése katalógusadatokon alapulnak, az értékük azonban nem valóságosak. A 7.3.3. fejezetben bemutatott hőmérsékletmenedzsment algoritmust létező megoldások egyszerűsítésével implementáltam.



## 8. fejezet

# Összefoglalás

Kutatásom során integrált áramkörök funkcionális és termikus működésének együttes szimulációjával foglalkoztam. Célom egy olyan módszer kidolgozása és megvalósítása volt, amely az eddigi megoldásokkal szemben az IC-k tervezési folyamata során alkalmazott elvonatkoztatási szintek mindegyikén használható az áramköri funkció és a termikus viselkedés együttes vizsgálatára. Ennek érdekében absztrakt interfészt dolgoztam ki, amely a célkitűzésnek megfelelően több elvonatkoztatási szinthez és modellező nyelvhez illeszthető. Bemutattam az interfész tervezési folyamatba illesztésének módját, amihez felhasználtam a szakirodalomban alkalmazott korábbi eredményeket és hagyományos EDA eszközöket.

A disszertáció második felében egy új termikus motort mutattam be, amely képes kihasználni a modern, többmagos CPU és GPU architektúrákat a gyors szimuláció érdekében. Az új szimulátorban megvalósított adaptív technika ugyanakkor csökkenti a relaxált együttes szimulációs módszer hátrányát a szimultán iterációs technikához képest. A termikus motort validáltam a szakirodalomban bemutatott, más 3D IC elrendezéseket támogató termikus szimulátorokkal, továbbá megvizsgáltam a termikus felbontás, időlépték és CPU magok számának hatását futásidő szempontjából. Kifejlesztettem egy keretrendszert, amely realizálja az értékezésben megfogalmazott absztrakt interfészt, amihez több – az integrált áramkörök fejlesztésénél használatos – hardverleíró nyelvet és az ismertetett termikus szimulátorokat is illesztettem. A keretrendszer felhasználásával három különféle mintapéldán keresztül – mindhárom processzort tartalmazó komplex rendszer – bemutattam a funkcionális-termikus együttes szimuláció lehetőségeit különböző elvonatkoztatási szinteken.

### Az új szimulációs módszer jelentősége

A kutatási eredményeken alapuló funkcionális-termikus szimulációs módszerrel a tervezési folyamat tetszőleges fázisában vizsgálható az integrált áramköri modell működésének hőmérséklet-függése. A mechanikai terveken dolgozó gépészmérnökök és a modul funkcionalitását implementáló villamosmérnökök munkája összefügg, a két tartomány egymásra hatását érdemes egy szimulációs környezetben vizsgálni. Ez megkönnyítheti a funkcionális modellen és a termikus megoldáson dolgozó mérnökök munkájának összehangolását.

Az új verifikációs módszerrel olyan tervezési döntések támaszthatóak alá, amelynek hatása van az áramkör termikus és funkcionális viselkedésére. Modern digitális integrált áramkörök esetén erre például a DTM és DVFS algoritmusok, és egyéb, a lapka hőmérsékletének mérésén alapuló, az áramkör működését befolyásoló szabályozó algoritmusok implementálásakor van szükség. Hagyományos tervezési folyamatban erre csak az áramköri szintű elektro-termikus szimulációval nyílik lehetőség.

## Nyitott kérdések

A szimuláció pontossága az alkalmazott disszipációmodellek, layout és termikus motor függvényében változhat. A bemutatott együttes szimulációs módszer bemenetként támaszkodik a disszipációmodellre és layout komponensek fizikai elrendezésére. A disszipáció modellezése a szakirodalomban részletesen tárgyalt terület, az IEEE szabványosított módszerek bemutatásán dolgozik [28, 29]. Az I. tézisben megfogalmazott absztrakt interfész, és az arra épülő keretrendszer lehetővé teszi tetszőleges disszipációmodell alkalmazását, amennyiben ezek illeszthetőek a tervezés során használt funkcionális szimulátorhoz.

A szimulációs keretrendszer jelenlegi implementációjában használt termikus szimulátorok akadémiai fejlesztések. Ipari megoldás – például Mentor® FloTHERM® – abban az esetben alkalmazható, ha a felhasználó program megfelelő API-t biztosít a termikus motorjához. A kutatómunkának nem volt célja, hogy a szimulációs keretrendszer kész termék legyen, így kereskedelmi termikus szimulátor illesztését rövid kutatómunkát követően nem valósítottam meg.

# Köszönetnyilvánítás

Köszönettel tartozom témavezetőmnek, Poppe Andrásnak a három éves támogató munkájáért, konzultációkért, cikkek lektorálásáért és a kapott lehetőségekért, mégha bizonyos esetekben nem is lelkesedtem értük.

Köszönöm Horváth Péternek a szakmai útmutatást és tanácsokat, a publikációim folyamatos és fáradhatatlan véleményezését. Egyúttal megköszönöm Czirkos Zoltánnak a keretrendszer implementálásában nyújtott folyamatos szakmai segítséget, Timár Andrásnak a CellTherm és egyéb forráskódok biztosítását, és Nagy Gergelynek az eredeti LogiTherm forráskódját.

Köszönöm Németh Márton és Takács Gábor segítségét a termikus szimulációhoz szükséges elméleti háttér elsajátításához, Bognár György, Gärtner Péter és Mann Zoltán kezdeti iránymutatását, amely végül a doktori kutatás irányába terelt, és végül a Budapesti Műszaki és Gazdaságtudományi Egyetem Elektronikus Eszközök Tanszékének valamennyi munkatársának, hogy munkámban segítettek.

Köszönöm családomnak és barátaimnak az évek során nyújtott folyamatos segítséget, támogatást és türelmet, amivel átsegítettek a hullámvölgyeken.

# A függelék

## A konduktív hővezetés differenciálegyenlete

A 2.2. fejezetben alkalmazott konduktív hővezetést leíró (2.4) differenciálegyenlet egy térfogatelemre felírható energiamérleg egyszerűsítésével határozható meg. Az alábbi egyenletben a  $V$  térfogatú elemre felírt energiamegmaradás törvénye látható,  $c_v$  az anyag térfogategységre eső hőkapacitása,  $\lambda$  annak hővezetése,  $T$  a hőmérséklete,  $T_a$  a környezet hőmérséklete,  $p$  a disszipációs-sűrűség és  $t$  az idő.

$$\int_V c_v T dV = \int_t \oint_A \lambda \cdot \text{grad}(T) dA dt + \int_t \oint_A c_v \bar{v} T dA dt + \int_t \oint_A \sigma (T - T_a)^4 dA dt + \int_t \int_V p dV dt \quad (\text{A.1})$$

A fenti egyenlet mindkét oldalát idő szerint deriválva:

$$\frac{d}{dt} \left( \int_V c_v T dV \right) = \oint_A \lambda \cdot \text{grad}(T) dA + \oint_A c_v \bar{v} T dA + \oint_A \sigma (T - T_a)^4 dA + \int_V p dV. \quad (\text{A.2})$$

Jobb oldalon Gauss-tételt alkalmazva:

$$\frac{d}{dt} \left( \int_V c_v T dV \right) = \int_V \text{div}(\lambda \cdot \text{grad}(T)) dV + \int_V \text{div}(c_v \bar{v} T) dV + \int_V \text{div}(\sigma (T - T_a)^4) dV + \int_V p dV. \quad (\text{A.3})$$

Bal oldalon alkalmazva a Leibniz-tételt:

$$\int_V c_v \frac{\partial T}{\partial t} dV = \int_V \text{div}(\lambda \cdot \text{grad}(T)) dV + \int_V \text{div}(c_v \bar{v} T) dV + \int_V \text{div}(\sigma (T - T_a)^4) dV + \int_V p dV. \quad (\text{A.4})$$

Térfogat szerint deriválva:

$$c_v \frac{\partial T}{\partial t} = \text{div}(\lambda \cdot \text{grad}(T)) + \text{div}(c_v \bar{v} T) + \text{div}(\sigma (T - T_a)^4) + p. \quad (\text{A.5})$$

Homogén anyagot feltételezve:

$$c_v \frac{\partial T}{\partial t} = \lambda \cdot \text{div}(\text{grad}(T)) + \text{div}(c_v \bar{v} T) + \text{div}(\sigma (T - T_a)^4) + p. \quad (\text{A.6})$$

Láncszabályt alkalmazva a konvektív tagra:

$$c_v \frac{\partial T}{\partial t} = \lambda \cdot \text{div}(\text{grad}(T)) + c_v T \cdot \text{div}(\bar{v}) + c_v \bar{v} \cdot \text{grad}(T) + \text{div}(\sigma (T - T_a)^4) + p. \quad (\text{A.7})$$

A gyakorlatban használt hűtőszközökben feltételezhető a tömeg összenyomhatatlansága a kis nyomáskülönbségek alkalmazása miatt:

$$c_v \frac{\partial T}{\partial t} = \lambda \cdot \operatorname{div}(\operatorname{grad}(T)) + c_v \bar{v} \cdot \operatorname{grad}(T) + \operatorname{div}(\sigma(T - T_a)^4) + p. \quad (\text{A.8})$$

Termikus szimuláció során a fenti differenciál-egyenlet megoldása valamilyen numerikus módszerrel határozható meg. A hőmérsékleti sugárzás a szilícium alapú integrált áramkörök üzemi hőmérsékleti tartományában elhanyagolható, lapka méretű szimulációs tartomány esetén a konvektív hőtranszportot jellemzően nem szükséges modellezni:

$$c_v \frac{\partial T}{\partial t} = \lambda \cdot \operatorname{div}(\operatorname{grad}(T)) + p. \quad (\text{A.9})$$

# B függelék

## Numerikus módszerek ODE-k megoldásához

Az értekezésben említett termikus motorok közül a HotSpot a negyedrendű Runge-Kutta módszert, a 3D-ICE és a SUNRED a hátralépő Euler módszert használja a differenciálegyenlet megoldásához. A SloTh termikus szimulátorban az előrelépő és hátralépő Euler, valamint a trapéz módszeren alapuló numerikus eljárásokat valósítottam meg a

$$\frac{dy}{dt} = f(t, y) \quad (\text{B.1})$$

alakú közönséges differenciálegyenletrendszer megoldásához. Ebben a függelékben bemutatom, hogy az egyes módszerek hogyan alkalmazhatóak, milyen lesz a kapott egyenletrendszer.

### Előrelépő Euler módszer

Előrelépő módszer esetén  $y_{(n+1)}$  a már ismert  $f(t_{(n)}, y_{(n)})$  alapján számítható:

$$\begin{aligned} \frac{y_{(n+1)} - y_{(n)}}{\Delta t} &= f(t_{(n)}, y_{(n)}), \\ y_{(n+1)} &= \Delta t \cdot f(t_{(n)}, y_{(n)}) + y_{(n)}. \end{aligned} \quad (\text{B.2})$$

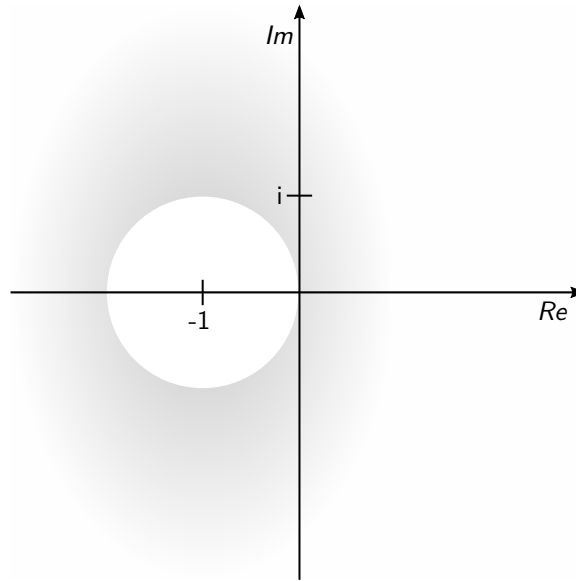
A fenti összefüggést alkalmazva a (6.16) egyenletrendszerre egy lépésben meghatározható az új hőmérsékletvektor:

$$\begin{aligned} \frac{\bar{T}_{(n+1)} - \bar{T}_{(n)}}{\Delta t} &= \mathbf{C}_{\text{th}}^{-1} (\mathbf{G}\bar{T}_{(n)} + \bar{P}) \\ \bar{T}_{(n+1)} &= \mathbf{C}_{\text{th}}^{-1} (\mathbf{G}\bar{T}_{(n)} + \bar{P}) \Delta t + \bar{T}_{(n)}. \end{aligned} \quad (\text{B.3})$$

A fenti egyenlet alapján az  $n + 1$ -edik hőmérsékletértékek kiszámításához elégséges két mátrixvektor szorzást és vektor-vektor összeadást elvégezni. Az előrelépő Euler módszer tehát kis számításigénnyel rendelkezik, azonban a numerikus stabilitás érdekében az időléptéket ( $\Delta t$ ) nagyon kicsire kell megválasztani. A stabilitás feltétele, hogy a (6.16) egyenletben a  $\mathbf{C}_{\text{th}}^{-1}\mathbf{G}$  mátrix összes sajátértékére ( $\lambda_i$ ) és az időléptékre ( $\Delta t$ ) teljesüljön az alábbi feltétel [119]:

$$|1 + \lambda_i \Delta t| \leq 1. \quad (\text{B.4})$$

Tehát  $\Delta t$ -t úgy kell megválasztani, hogy a fenti egyenlőtlenség a legkisebb sajátérték esetén is teljesüljön. A  $\lambda_i \Delta t$  szorzat megengedett értékei a B.1. ábrán fehérrel jelölt körlapon találhatóak. Az előrelépő Euler módszer kumulatív hibája elsőrendű, vagyis a sorozatos számítások utáni hiba az időléptékkal arányos [119].



B.1. ábra. Előrelépő Euler stabilitási régiója a komplex számsíkon ábrázolt fehér körlap [119]

### Hátralépő Euler módszer

Hátralépő Euler módszer implicit,  $y_{(n+1)}$  kiszámításához a  $f(t_{(n+1)}, y_{(n+1)})$  függvény ismerete szükséges:

$$\begin{aligned} \frac{y_{(n+1)} - y_{(n)}}{\Delta t} &= f(t_{(n+1)}, y_{(n+1)}), \\ y_{(n+1)} &= \Delta t \cdot f(t_{(n+1)}, y_{(n+1)}) + y_{(n)}. \end{aligned} \quad (\text{B.5})$$

A fenti összefüggést alkalmazva a (6.16) egyenletrendszerre adódik a megoldandó egyenletrendszer:

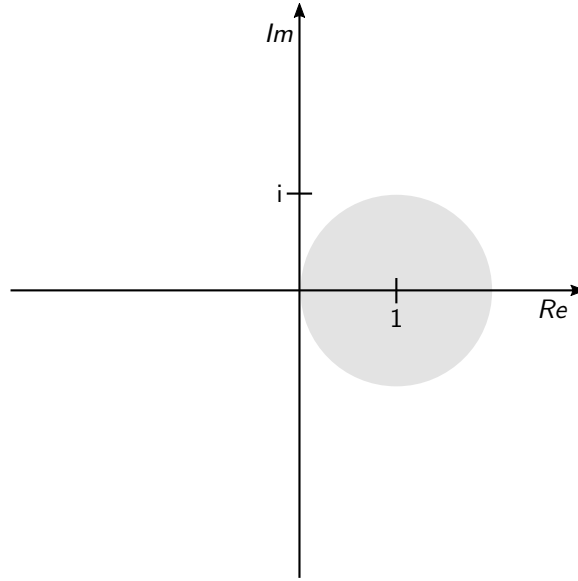
$$\begin{aligned} \frac{\bar{T}_{(n+1)} - \bar{T}_{(n)}}{\Delta t} &= \mathbf{C}_{\text{th}}^{-1} (\mathbf{G}\bar{T}_{(n+1)} + \bar{P}), \\ \left(-\mathbf{G} + \frac{\mathbf{C}_{\text{th}}}{\Delta t}\right) \bar{T}_{(n+1)} &= \bar{P} + \frac{\mathbf{C}_{\text{th}}}{\Delta t} \bar{T}_{(n)}. \end{aligned} \quad (\text{B.6})$$

Az alábbi egyenlet írja le a hátralépő Euler módszer stabilitási feltételét [119] (B.2. ábra):

$$\frac{1}{|1 - \lambda_i \Delta t|} \leq 1, \quad (\text{B.7})$$

ahol  $\lambda_i$  a (6.16) lineáris egyenletrendszerben szereplő  $\mathbf{C}_{\text{th}}^{-1}\mathbf{G}$  mátrix sajátértéke.

Az hátralépő Euler módszer kumulatív hibája elsőrendű, vagyis a sorozatos számítások utáni hiba az időléptékkel arányos [119].



B.2. ábra. Hátralépő Euler stabilitási régiója a szürke körlap kivételével a teljes komplex számsík

### Trapéz módszer

A trapéz módszer az előrelépő és hátralépő Euler kombinációjának tekinthető,  $y_{(n+1)}$  kiszámításához szükséges  $f(t_{(n)}, y_{(n)})$  és  $f(t_{(n+1)}, y_{(n+1)})$  is:

$$\begin{aligned} \frac{y_{(n+1)} - y_{(n)}}{\Delta t} &= \frac{1}{2}f(t_{(n)}, y_{(n)}) + \frac{1}{2}f(t_{(n+1)}, y_{(n+1)}), \\ y_{(n+1)} &= \Delta t \cdot \frac{1}{2}f(t_{(n)}, y_{(n)}) + \Delta t \cdot \frac{1}{2}f(t_{(n+1)}, y_{(n+1)}) + y_{(n)}. \end{aligned} \quad (\text{B.8})$$

A fenti összefüggést alkalmazva a (6.16) egyenletrendszerre adódik a megoldandó egyenletrendszer:

$$\begin{aligned} \frac{\bar{T}_{(n+1)} - \bar{T}_{(n)}}{\Delta t} &= \frac{\mathbf{C}_{\text{th}}^{-1}}{2} (\mathbf{G}\bar{T}_{(n)} + \bar{P} + \mathbf{G}\bar{T}_{(n+1)} + \bar{P}), \\ \left(-\mathbf{G} + \frac{2\mathbf{C}_{\text{th}}}{\Delta t}\right) \bar{T}_{(n+1)} &= \left(\mathbf{G} + \frac{2\mathbf{C}_{\text{th}}}{\Delta t}\right) \bar{T}_{(n)} + 2\bar{P}. \end{aligned} \quad (\text{B.9})$$

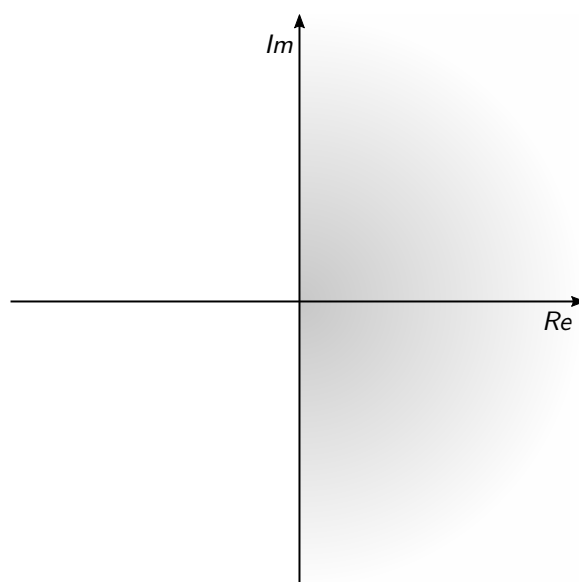
A következő összefüggés a trapéz szabályon alapuló módszer stabilitási feltétele [120] (B.3. ábra):

$$\left| \frac{1 + \frac{1}{2}\lambda_i \Delta t}{1 - \frac{1}{2}\lambda_i \Delta t} \right| \leq 1, \quad (\text{B.10})$$

ahol  $\lambda_i$  a (6.16) lineáris egyenletrendszerben szereplő  $\mathbf{C}_{\text{th}}^{-1}\mathbf{G}$  mátrix sajátértéke.

A trapéz módszer kumulatív hibája másodrendű, vagyis a sorozatos számítások utáni hiba az időlépték négyzetével arányos [120].





B.3. ábra. Trapéz módszer stabilitási régiója a komplex számsík bal oldala

# C függelék

## További szimulációs eredmények

Az Elektronikus Eszközök Tanszékének IC tervező laborjaiban az Europractice [121] által licenszelt AMS 0,35  $\mu\text{m}$  CMOS technológia van telepítve. A 25. oldalon a 3.1. fejezetben bemutatott processzorok összetettsége nem indokolja a rendelkezésre álló PDK-nál fejlettebb gyártástechnológia alkalmazását. Modernebb technológia vizsgálatához jelentősen összetettebb példaáramkörökre lenne szükség, ilyenek azonban sajnos nem álltak rendelkezésre, kifejlesztésük pedig nem volt lehetséges a kutatás keretein belül.

A fenti okok miatt statikus disszipáció hatását az AMS 0,35  $\mu\text{m}$  CMOS technológia módosított változatán mutatom be. A PDK-ban szivárgásból adódó fogyasztás három sarokpontban<sup>1</sup> áll rendelkezésre. A sarokpontok között nem csak a hőmérséklet, hanem a tápfeszültség is változik, így az adatok alapján nem lehet kizárólag hőmérsékletfüggést modellezni. Ezért sorozatos SPICE szimulációval különböző hőmérsékletek mellett meghatároztam a processzormodellekben felhasznált logikai kapuk szivárgási áramát. A NAND20 cella szivárgási árama a 37. oldalon az 5.5. ábrán látható. A kapott eredményeket felskáláztam, hogy szobahőmérsékleten a szivárgásból származó disszipáció a dinamikus fogyasztás megközelítőleg fele legyen.

A módosított statikus disszipáció hatásának vizsgálatához újra lefuttattam a 68. oldalon a 7.3. fejezetben olvasható szimulációkat.

### C.1. $\alpha$ processzor

A statikus fogyasztáson kívül az  $\alpha$  processzor összes paraméterét változatlanul hagytam. Ebben a fejezetben bemutatom a processzor RTL és kapuszintű megvalósításának hőmérsékletét és fogyasztását a módosított AMS 0,35  $\mu\text{m}$  CMOS technológián.

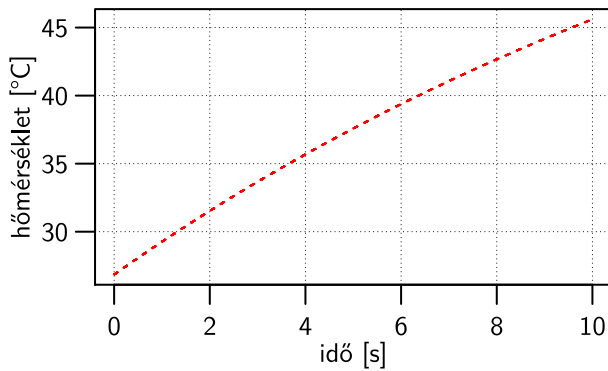
#### Kapuszintű modell

A C.1(a) ábrán látható az  $\alpha$  processzor kapuszintű modelljének maximális hőmérséklete a szimuláció során. A lapka legnagyobb hőmérséklete  $\sim 45,6^\circ\text{C}$ , ami nagyobb a 68. oldalon a 7.3.1. fejezetben látható értéknél. Ezt az eltérést a megnövekedett disszipáció okozza. A C.1(b) ábra mutatja be a processzor statikus és dinamikus fogyasztását az idő függvényében. A dinamikus fogyasztás a szimuláció során átlagosan 0,317 mW, a statikus fogyasztás a szimuláció elején 0,163 mW ami a szimuláció végére  $\sim 0,179$  mW-ra emelkedik.

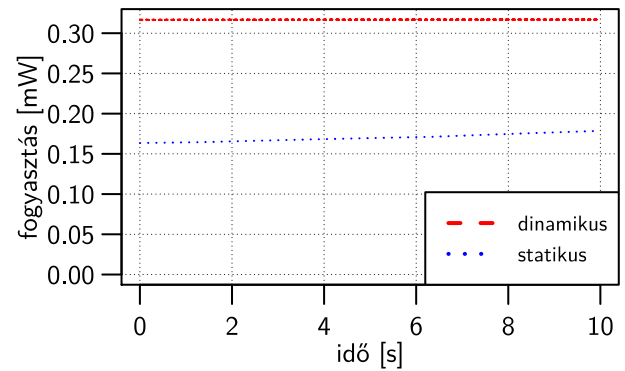
A C.1(c) ábrán látható az  $\alpha$  processzor kapszintű modell hőmérsékleteloszlása a szimuláció végén. A maximális hőmérséklet nagyobb a 69. oldalon a 7.6(a) ábrán jelzett értéknél, mivel a

---

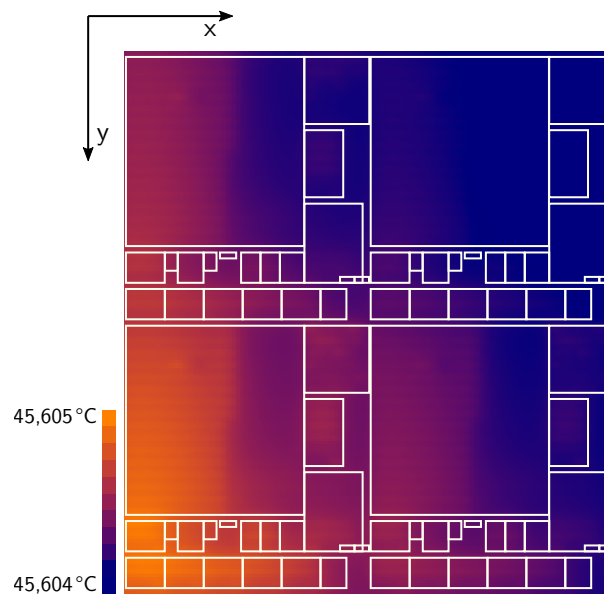
<sup>1</sup> A három pont jellemzően a legkedvezőbb, a tipikus, és a legrosszabb értékeket fedi.



(a) A processzor maximális hőmérséklete



(b) A processzor fogyasztása



(c) Hőmérsékleteloszlás a szimuláció végén

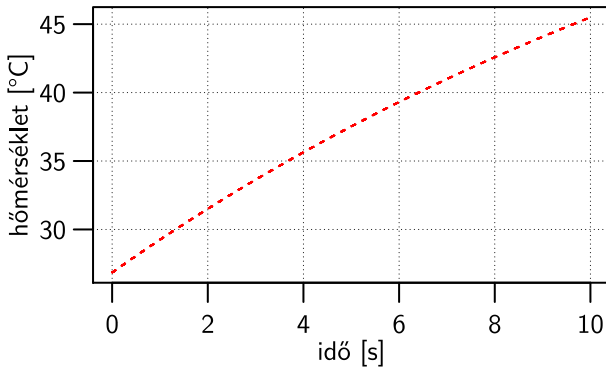
C.1. ábra. Az  $\alpha$  processzor kapusintű modelljének hőmérséklete és fogyasztása

processzor többet fogyaszt. A lapkán kialakult hőmérsékletkülönbség a korábbi esetekhez hasonlóan gyakorlatilag elhanyagolható.

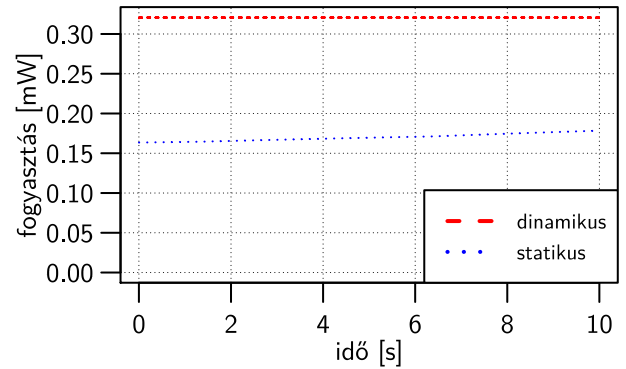
## RTL modell

A C.2(a) ábrán látható az  $\alpha$  processzor RTL modelljének maximális hőmérséklete a szimuláció során. A lapka legnagyobb hőmérséklete  $\sim 45,5^\circ\text{C}$ , ami nagyobb a 68. oldalon a 7.3.1. fejezetben látható értéknél. Ezt az eltérést a megnövekedett disszipáció okozza. A C.2(b) ábra mutatja be a processzor statikus és dinamikus fogyasztását az idő függvényében. A dinamikus fogyasztás a szimuláció során átlagosan  $0,32\text{ mW}$ , a statikus fogyasztás a szimuláció elején  $0,163\text{ mW}$  ami a szimuláció végére  $\sim 0,178\text{ mW}$ -ra emelkedik.

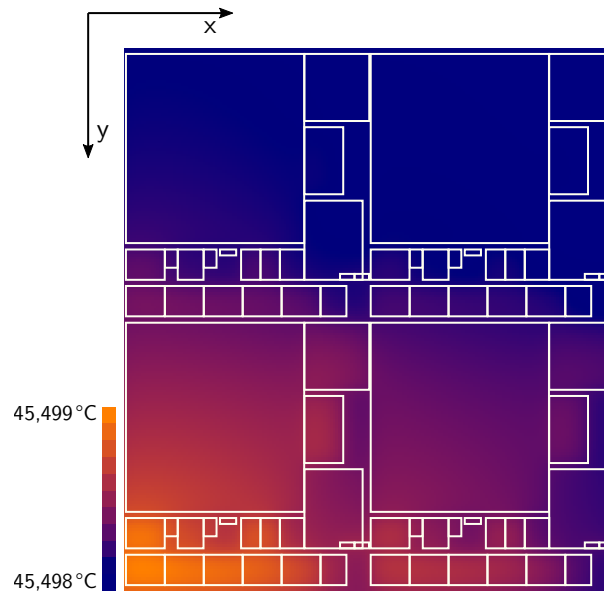
A C.2(c) ábrán látható az  $\alpha$  processzor RTL implementációjának hőmérsékleteloszlása a szimuláció végén. A maximális hőmérséklet nagyobb a 69. oldalon a 7.6(b) ábrán jelzett értéknél, mivel a processzor többet fogyaszt. A lapkán kialakult hőmérsékletkülönbség a korábbi esetekhez hasonlóan gyakorlatilag elhanyagolható.



(a) A processzor maximális hőmérséklete



(b) A processzor fogyasztása



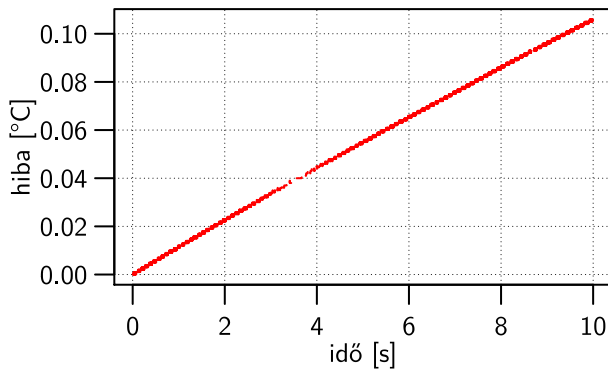
(c) Hőmérsékleteloszlás a szimuláció végén

C.2. ábra. Az  $\alpha$  processzor RTL implementációjának hőmérséklete és fogyasztása

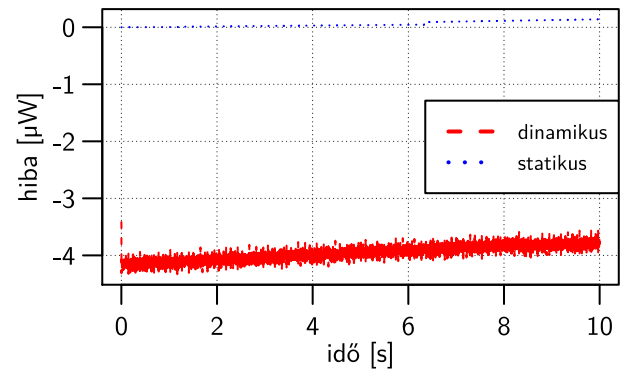
### Az eredmények összehasonlítása

A kapusintű és RTL modellek között tapasztalt különbséget az RTL disszipációmodell hibája és a layoutok eltérő felbontása okozza. A 68. oldalon a 7.3.1. fejezetben látható eredményekhez képest ezeknél az eredményeknél a kapusintű modell hőmérséklete nagyobb az RTL-hez képest a kisebb fogyasztás ellenére. Ezt az okozza, hogy az RTL disszipációmodell relatív hibája a megnövekedett fogyasztás miatt csökken, a kapusintű modell környezet felé tapasztalt eredő hőellenállása pedig nagyobb a layoutot alkotó kisméretű alkatrészek miatt.

A C.3(a) ábrán látható az RTL modell hőmérsékletének, a C.3(b) ábrán pedig a statikus és dinamikus fogyasztásának hibája látható a kapusintű modellhez képest. A statikus disszipáció eltérését a hőmérséklet hibája okozza.



(a) A hőmérsékletkülönbség



(b) A fogyasztások közötti különbség

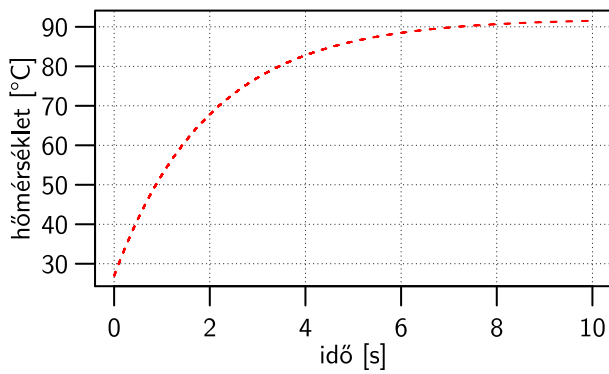
C.3. ábra. Az  $\alpha$  processzor RTL modelljének abszolút hibája a kapusintű modellhez képest

## C.2. $\beta$ processzor

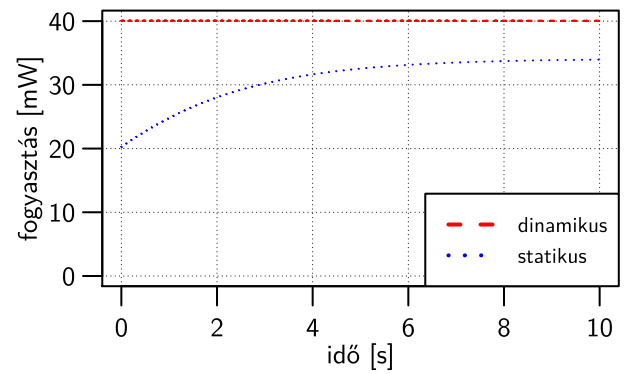
A  $\beta$  processzor esetén a megnövelt statikus disszipáció a lapka hőmérsékletének jelentős növekedését okozta volna, ezért a peremfeltételt  $200 \text{ W/m}^2 \text{ K}$ -re módosítottam, a többi paramétert változatlanul hagytam.

A C.4(a) ábrán látható a  $\beta$  processzor maximális hőmérséklete a szimuláció során. Látható, hogy a hőmérséklet felfutása gyorsabb a 73. oldalon a 7.12(a) ábrán látható eredménynél. A módosított peremfeltétel miatt a rendszerre jellemző legnagyobb időállandó kisebb lett, így a hőmérséklet hamarabb eléri az állandósult állapotot. A C.4(b) ábra mutatja be a processzor statikus és dinamikus fogyasztását az idő függvényében. A várakozásnak megfelelően a statikus fogyasztás a hőmérséklet emelkedésével nőtt.

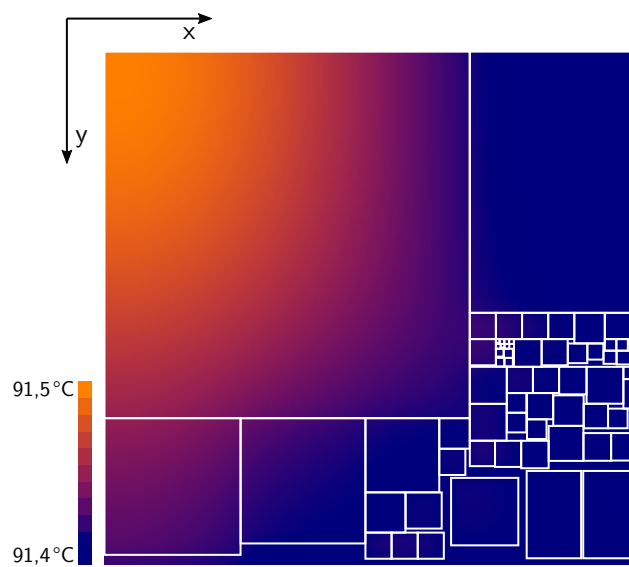
A C.4(c) ábrán a lapkán hőmérsékleteloszlása látható a szimuláció végén. A maximális hőmérséklet a 73. oldalon a 7.12(b) ábrán látható értéknél kevesebb, mert a fogyasztás kisebb mértékben nőtt a peremfeltételhez képest.



(a) A processzor maximális hőmérséklete



(b) A processzor fogyasztása



(c) Hőmérsékleteloszlás a szimuláció végén

C.4. ábra. A  $\beta$  processzor hőmérséklete és fogyasztása

# D függelék

## Rövidítések jegyzéke

ADC	Analog Digital Converter
ALU	Arithmetic Logic Unit
AMS	Analog Mixed Signal/AustriaMicroSystems
API	Application Programming Interface
ASIC	Application Specific Integrated Circuit
ATLAS	Automatically Tuned Linear Algebra Software
ATMI	Analytical Model of Temperature in Microprocessors
BHT	Branch History Table
BP	Branch Predictor
OpenBLAS	Open Basic Linear Algebra Subprograms
BSIM	Berkeley Short-Channel IGFET Model
CAD	Computer Aided Design
Cadence <sup>®</sup> IES	Cadence <sup>®</sup> Incisive Enterprise Simulator
CACTI	Cache Access and Cycle Time Model
CMOS	Complementary Metal Oxide Semiconductor
CSV	Comma Separated Value
DAC	Digital Analog Converter
DEF	Design Exchange Format
DRC	Design Rule Check
DSP	Digital Signal Processor
DTM	Dynamic Thermal Management
DVFS	Dynamic Voltage Frequency Scaling
EDA	Electronic Design Automation
ESA	European Space Agency
FDM	Finite Difference Method
FEM	Finite Element Method
FFT	Fast Fourier Transform
finFET	fin Field Effect Transistor
FSM	Finite State Machine
FVM	Finite Volume Method
GPU	Graphics Processor Unit
HDL	Hardware Description Language
HKMG	High $\kappa$ Metal Gate
IC	Integrated Circuit

---

ID	Instruction Decode
IEEE	Institute of Electrical and Electronics Engineers
IFU	Instruction Fetch Unit
IGFET	Insulated Gate Field Effect Transistor
Intel® MKL	Intel® Math Kernel Library
IP	Intellectual Property
IS	Instruction Scheduler
L1-D	L1 Data cache
LAPACK	Linear Algebra Package
LEF	Library Exchange Format
LUT	Look Up Table
LVS	Layout Vs. Schematic
McPAT	Multicore Power, Area, and Timing model
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
NID	Network Identification by Deconvolution
NYHL	Nyomtatott Huzalozású Lemez
ODE	Ordinary Differential Equation
PARSEC	Princeton Application Repository for Shared-Memory Computers
PDK	Process Design Kit
PID	Proportional Integral Derivative
PLATO	Planetary Transits and Oscillations of stars
RTL	Register Transfer Level
SISSI	Simulator for Integrated Structures by Simultaneous Iteration
SiP	System in Package
SMT	Simultaneous Multithreading
SoC	System on Chip
SOI	Silicon on Insulator
SPEC	Standard Performance Evaluation Corporation
SPICE	Simulation Program with Integrated Circuit Emphasis
SST	Structural Simulation Toolkit
STA	Static Timing Analysis
SUNRED	Successive Network Reduction
SuperLU	Supernodal LU
SVM	Silicon Valley Microelectronics
TDP	Thermal Design Power
TIM	Thermal Interface Material
TIMA	Techniques de l'Informatique et de la Microélectronique pour l'Architecture des systèmes intégrés
TLM	Transaction Level Modeling
UML	Unified Modeling Language
UPF	Unified Power Format
VLSI	Very Large Scale Integration
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
VHPI	VHDL Procedural Interface
VPI	Verilog Procedural Interface
3D-ICE	3D Interlayer Cooling Emulator

---



# Saját közlemények

- [S1] Gy. Bognár, G. Takács, L. Pohl, *L. Jani*, A. Timár, P. Horváth, M. Németh, A. Poppe, and P.G. Szabó. Integrating Chip-level Microfluidics Cooling into System Level Design of Digital Circuits. In *IEEE Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM)*, pages 77–87, March 2017. DOI:10.1109/SEMI-THERM.2017.7896912.
- [S2] M. Németh, G. Takács, *L. Jani*, and A. Poppe. Compact modeling approach for microchannel cooling and its validation. *Microsystem Technologies*, 23:1–13, 2017. DOI:10.1007/s00542-017-3330-z.
- [S3] M. Németh, *L. Jani*, and A. Poppe. Compact modeling approach for microchannel cooling aimed at high-level thermal analysis of 3D packaged ICs. In *Symposium on Design, Test, Integration and Packaging of MEMS/MOEMS (DTIP)*, pages 182–187, May 2016. DOI:10.1109/DTIP.2016.7514865.
- [S4] *L. Jani* and A. Poppe. Multilevel logic and thermal co-simulation. *Microelectronics Reliability*, 67:46–53, 2016. DOI:10.1016/j.microrel.2016.08.019.
- [S5] *L. Jani* and A. Poppe. Extension of SystemC with logi-thermal simulation capabilities. In *International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*, pages 1–4, Sept 2015. DOI:10.1109/THERMINIC.2015.7389604.
- [S6] *L. Jani* and A. Poppe. Framework for thermal-aware verification of digital and mixed signal systems. *Microelectronics Reliability*, 68:1–12, 2017. DOI:10.1016/j.microrel.2017.03.023.
- [S7] *L. Jani* and A. Poppe. Improved method for logi-thermal simulation with temperature dependent signal delay. In *International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*, pages 302–306, Sept 2016. DOI:10.1109/THERMINIC.2016.7749071.
- [S8] *L. Jani* and A. Poppe. Adaptive co-simulation of functional-thermal behaviour of integrated circuits. In *International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*, pages 1–8, Sept 2017. DOI:10.1109/THERMINIC.2017.8233823.
- [S9] *L. Jani* and A. Poppe. Extending a Multi-Level Logi-Thermal Simulation Framework to a Mixed Signal Thermal Aware Simulation Environment Using SystemC-AMS. In *The Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITHERM)*, pages 307–312, June 2017. DOI:10.1109/ITHERM.2017.7992486.

# Tézishez szorosan nem kapcsolódó közlemények

- [N1] *L. Jani* and *Z. Á. Mann*. Cache optimization for CPU-GPU heterogeneous processors. *American Journal of Algorithms and Computing*, 2(1):18–31, 2015. DOI:10.7726/ajac.2013.1001.
- [N2] *L. Jani* and *Z. Á. Mann*. Megosztott cache heterogén multiprocesszoros rendszerekben (1. rész). *ELEKTRONET*, XXIV(8):28–30, 2015.
- [N3] *L. Jani* and *Z. Á. Mann*. Megosztott cache heterogén multiprocesszoros rendszerekben (2. rész). *ELEKTRONET*, XXV(1):30–32, 2016.

# Irodalomjegyzék

- [1] G. E. Moore. Cramming More Components onto Integrated Circuits. *Electronics*, 38(8):114–117, 1965. DOI:10.1109/jproc.1998.658762.
- [2] A. Danowitz and K. Kelley and J. Mao and J. P. Stevenson and M. Horowitz. CPU DB: Recording Microprocessor History. *Communications of the ACM*, 55(4):55–63, 2012. DOI:10.1145/2133806.2133822.
- [3] R. H. Dennard, F. H. Gaensslen, V. L. Rideout, E. Bassous, and A. R. LeBlanc. Design of ion-implanted MOSFET's with very small physical dimensions. *IEEE Journal of Solid-State Circuits*, 9(5):256–268, 1974. DOI:10.1109/jssc.1974.1050511.
- [4] J. L. Hennessy and D. A. Patterson. Trends In Power and Energy in Integrated Circuits. In *Computer Architecture, Fifth Edition: A Quantitative Approach*, chapter 1.5, pages 21–26. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 5th edition, 2011.
- [5] D. D. Gajski and R. H. Kuhn. Guest Editors' Introduction: New VLSI Tools. *IEEE Computer*, 16(12):11–14, Dec 1983. DOI:10.1109/MC.1983.1654264.
- [6] H. Kaeslin. Modelling Hardware with VHDL. In *Digital Integrated Circuit Design From VLSI Architectures to CMOS fabrication*, chapter 4, pages 175–285. Cambridge University Press, New York, USA, 1st edition, 2008.
- [7] P. Horváth. *Rendszerszintű szintézis módszer automatizált RTL tervezéshez*. PhD thesis, Budapesti Műszaki és Gazdaságtudományi Egyetem, 2016.
- [8] A. Timár. *Logi-termikus szimuláció sztenderd tervező rendszerekben*. PhD thesis, Budapesti Műszaki és Gazdaságtudományi Egyetem, 2013.
- [9] S. J. E. Wilton and N. P. Jouppi. CACTI: an enhanced cache access and cycle time model. *IEEE Journal of Solid-State Circuits*, 31(5):677–688, May 1996. DOI:10.1109/4.509850.
- [10] V. Tiwari D. Brooks and M. Martonosi. Wattch: a framework for architectural-level power analysis and optimizations. In *International Symposium on Computer Architecture (ISCA)*, pages 83–94, May 2000. DOI:10.1145/339647.339657.
- [11] S. Li et al. McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 469–480, May 2009.
- [12] S. S. Kumar, A. Zjajo, and R. v. Leuken. Ctherm: An Integrated Framework for Thermal-Functional Co-simulation of Systems-on-Chip. In *23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, pages 674–681, March 2015. DOI:10.1109/PDP.2015.56.

- 
- [13] W. Heirman, S. Sarkar, T. E. Carlson, I. Hur, and L. Eeckhout. Power-aware multi-core simulation for early design stage hardware/software co-optimization. In *International Conference on Parallel Architectures and Compilation Techniques (PACT)*, pages 3–12, Sept 2012. DOI:10.1145/2370816.2370820.
- [14] T. Nowatzki, J. Menon, C.H. Ho, and K. Sankaralingam. gem5, gpgpusim, mcpat, gpuwattch, "your favorite simulator here" considered harmful, 2014. URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.699.3967>.
- [15] E. Macii and M. Poncino. Power Macro-Models for High-Level Power Estimation. In C. Piguet, editor, *Low Power CMOS circuits: Technology, Logic Design and CAD tools*, chapter 19, pages 334–351. Taylor & Francis Group, Boca Raton, FL, USA, 1st edition, 2005.
- [16] G.B. Vece, M. Conti, and S. Orcioni. Transaction-level Power Analysis of VLSI Digital Systems. *Integration, the VLSI Journal*, 50(C):116–126, jun 2015. DOI:10.1016/j.vlsi.2015.02.003.
- [17] S. Ahuja, D. A. Mathaikutty, S. Shukla, and A. Dingankar. Assertion-Based Modal Power Estimation. In *International Workshop on Microprocessor Test and Verification*, pages 3–7, Dec 2007. DOI:10.1109/MTV.2007.10.
- [18] S. Ahuja, D. A. Mathaikutty, G. Singh, J. Stetzer, S. K. Shukla, and A. Dingankar. Power estimation methodology for a high-level synthesis framework. In *International Symposium on Quality Electronic Design*, pages 541–546, March 2009. DOI:10.1109/ISQED.2009.4810352.
- [19] A. Bogliolo, L. Benini, and G. De Micheli. Regression-based RTL Power Modeling. *ACM Transactions on Design Automation of Electronic Systems*, 5(3):337–372, Jul 2000. DOI:10.1145/348019.348081.
- [20] B. C. Schafer and K. Wakabayashi. Precision tunable RTL macro-modelling cycle-accurate power estimation. *IET Computers Digital Techniques*, 5(2):95–103, March 2011. DOI:10.1049/iet-cdt.2010.0044.
- [21] W. Q. Y. Cao, Y. Yan, and X. Gao. Neural Network Macromodel for High-Level Power Estimation of CMOS Circuits. In *International Conference on Neural Networks and Brain*, volume 2, pages 1009–1014, Oct 2005. DOI:10.1109/ICNNB.2005.1614789.
- [22] S. Gupta and F. N. Najm. Power modeling for high-level power estimation. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 8(1):18–29, Feb 2000. DOI:10.1109/92.820758.
- [23] A. Timár and M. Rencz. Real-time heating and power characterization of cells in standard cell designs. *Microelectronics Journal*, 44(11):977 – 985, 2013. DOI:10.1016/j.mejo.2012.03.002.
- [24] J. J. Ebers and J. L. Moll. Large-Signal Behavior of Junction Transistors. *Proceedings of the IRE*, 42(12):1761–1772, Dec 1954. DOI:10.1109/JRPROC.1954.274797.
- [25] H. K. Gummel and H. C. Poon. An Integral Charge Control Model of Bipolar Transistors. *Bell System Technical Journal*, 49(5):827–852, 1970. DOI:10.1002/j.1538-7305.1970.tb01803.x.
- [26] B. J. Sheu, D. L. Scharfetter, P. K. Ko, and M. C. Jeng. BSIM: Berkeley short-channel IGFET model for MOS transistors. *IEEE Journal of Solid-State Circuits*, 22(4):558–566, Aug 1987. DOI:10.1109/JSSC.1987.1052773.
-

- [27] IEEE Standard for Design and Verification of Low-Power, Energy-Aware Electronic Systems. *IEEE Std 1801-2015*, pages 1–515, March 2016. DOI:10.1109/IEEESTD.2016.7445797.
- [28] IEEE Project P2415 - Standard for Unified Hardware Abstraction and Layer for Energy Proportional Electronic Systems, 2017. URL: <https://standards.ieee.org/develop/project/2415.html>.
- [29] IEEE Project P2416 - Standard for Power Modeling to Enable System Level Analysis, 2017. URL: <https://standards.ieee.org/develop/project/2416.html>.
- [30] A. Bar-Cohen, A. A. Watwe, and R. S. Prasher. Heat Transfer in Electronic Equipment. In A. Bejan and A. D. Kraus, editors, *Heat Transfer Handbook*, chapter 13, pages 947–1028. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2003.
- [31] L. Pohl. *Speciális félvezetőeszközök szimulációja szukcesszív hálózatredukciós módszerrel*. PhD thesis, Budapesti Műszaki és Gazdaságtudományi Egyetem, 2012.
- [32] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy. Compact thermal modeling for temperature-aware design. In *Design Automation Conference (DAC)*, pages 878–883, July 2004. DOI:10.1145/996566.996800.
- [33] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan. HotSpot: A compact thermal modeling methodology for early-stage VLSI design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 14(5):501–513, 2006. DOI:10.1109/TVLSI.2006.876103.
- [34] M. R. Stan, K. Skadron, M. Barcella, W. Huang, K. Sankaranarayanan, and S. Velusamy. Hot-spot: A dynamic compact thermal model at the processor-architecture level. *Microelectronics Journal*, 34(12):1153–1165, 2003. DOI:10.1016/S0026–2692(03)00206–4.
- [35] HotSpot forráskód, 2017. URL: <http://lava.cs.virginia.edu/HotSpot/>.
- [36] M.-Y. Hsieh. A scalable simulation framework for evaluating thermal management techniques and the lifetime reliability of multithreaded multicore systems. In *International Green Computing Conference and Workshops*, pages 1–6, July 2011. DOI:10.1109/IGCC.2011.6008574.
- [37] Zs. Benedek, E. Kollár, B. Courtois, G. Farkas, S. Mir, A. Poppe, M. Rencz, V. Székely, and K. Torki. A Scalable Multi-functional Thermal Test Chip Family: Design and Evaluation. *Journal of Electronic packaging*, 123(4):323–330, 2001. DOI:10.1115/1.1389846.
- [38] K. Skadron, M. R. Stan, Wei Huang, K. Sankaranarayanan, Zhijian Lu, and J. Lach. A computer-architecture approach to thermal management in computer systems: opportunities and challenges. In *International Conference on Thermal and Mechanical Simulation and Experiments in Microelectronics and Microsystems (EuroSimE)*, pages 415–422, May 2004. DOI:10.1109/ESIME.2004.1304072.
- [39] A. Sridhar, A. Vincenzi, M. Ruggiero, T. Brunschwiler, and D. Atienza. Compact transient thermal model for 3D ICs with liquid cooling via enhanced heat transfer cavity geometries. In *International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*, pages 1–6, Oct 2010.
-

- 
- [40] A. Sridhar, A. Vincenzi, M. Ruggiero, T. Brunswiler, and D. Atienza. 3D-ICE: Fast compact transient thermal modeling for 3D ICs with inter-tier liquid cooling. In *International Conference on Computer-Aided Design (ICCAD)*, pages 463–470, Nov 2010.
- [41] A. Sridhar, A. Vincenzi, D. Atienza, and T. Brunswiler. 3D-ICE: A compact thermal model for early-stage design of liquid-cooled ICs. *IEEE Transactions on Computers*, 63(10):2576–2589, 2014. DOI:10.1109/TC.2013.127.
- [42] 3D-ICE forráskód, 2017. URL: <http://es1.epfl.ch/3D-ICE>.
- [43] V. Székely. SUNRED a new thermal simulator and typical applications. In *International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*, pages 21–23, Sept 1997.
- [44] Zs. Kohári, V. Székely, M. Rencz, A. Páhi, V. Dudek, and B. Höfflinger. Studies on the heat removal features of stacked SOI structures with a dedicated field solver program (SUNRED). *Microelectronics Reliability*, 38(12):1881–1891, 1998. DOI:10.1109/ESSDERC.1997.194474.
- [45] A. Páhi and V. Székely and M. Rosenthal and M. Rencz. 3D extension of the SUNRED field solver. In Anon, editor, *International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*, pages 185–190, Sept 1998.
- [46] L. Pohl. Multithreading and Strassen’s algorithms in SUNRED field solver. In *International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*, pages 137–141, Sept 2008. DOI:10.1109/THERMINIC.2008.4669895.
- [47] A. Poppe, L. Pohl, E. Kollár, Z. Kohári, H. Lifka, and C. Tanase. Methodology for thermal and electrical characterization of large area OLEDs. In *IEEE Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM)*, pages 38–44, March 2009. DOI:10.1109/STHERM.2009.4810740.
- [48] Zs. Kohári, L. Pohl, and A. Poppe. How thermal environment affects OLEDs’ operational characteristics. In *IEEE Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM)*, pages 331–336, March 2012. DOI:10.1109/STHERM.2012.6188869.
- [49] A. Poppe, G. Farkas, M. Rencz, Zs. Benedek, L. Pohl, V. Székely, K. Torki, S. Mir, and B. Courtois. Design issues of a multi-functional intelligent thermal test die. In *IEEE Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM)*, pages 50–57, 2001. DOI:10.1109/STHERM.2001.915144.
- [50] D. Fetis and P. Michaud. An evaluation of HotSpot-3.0 block-based temperature model, 2006. URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.78.336>.
- [51] W. Huang, K. Sankaranarayanan, R. J. Rib, M. R. Stan, and K. Skadron. An Improved Block-Based Thermal Model in HotSpot 4.0 with Granularity Considerations, 2007. URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.80.3864>.
- [52] L. Codecasa. Nonlinear dynamic compact thermal models by structure-preserving projection. *Microelectronics Journal*, 45(12):1764–1769, 2014. DOI:10.1016/j.mejo.2014.06.001.
-

- 
- [53] L. Codecasa, V. d'Alessandro, A. Magnani, and N. Rinaldi. Structure-preserving approach to multi-port dynamic compact models of nonlinear heat conduction. *Microelectronics Journal*, 46(12A):1129–1137, 2015. DOI:10.1016/j.mejo.2015.09.021.
- [54] L. Codecasa, V. d'Alessandro, A. Magnani, N. Rinaldi, and P. J. Zampardi. FAst Novel Thermal Analysis Simulation Tool for Integrated Circuits (FANTASTIC). In *International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*, pages 1–6, Sept 2014. DOI:10.1109/THERMINIC.2014.6972507.
- [55] M. Németh. *Mikrorendszerek transzportfolyamatainak redukált rendű modelljei*. PhD thesis, Budapesti Műszaki és Gazdaságtudományi Egyetem, 2017.
- [56] V. Székely. THERMODEL: a tool for compact dynamic thermal model generation. *Microelectronics Journal*, 29(4):257 – 267, 1998. Thermal Investigations of ICs and Microstructures II. DOI:10.1016/S0026–2692(97)00065–7.
- [57] V. Székely, M. Rencz, A. Poppe, and B. Courtois. THERMODEL: A Tool for Thermal Model Generation, and Application for MEMS. *Analog Integrated Circuits and Signal Processing*, 29(1):49–59, Oct 2001. DOI:10.1023/A:1011226213197.
- [58] V. Székely. Identification of RC networks by deconvolution: chances and limits. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 45(3):244–258, Mar 1998. DOI:10.1109/81.662698.
- [59] M. Rencz, V. Székely, A. Morelli, and C. Villa. Determining Partial Thermal Resistances With Transient Measurements, and Using the Method to Detect die Attach Discontinuities. In *IEEE Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM)*, pages 15–20, March 2002. DOI:10.1109/STHERM.2002.991340.
- [60] A. Poppe. Simulation of LED Based Luminaires by Using Multi-Domain Compact Models of LEDs and Compact Thermal Models of their Thermal Environment. *Microelectronics Reliability*, 72(5):65–74, 2017. DOI:10.1016/j.microrel.2017.03.039.
- [61] Y. Halevi. Can any reduced order model be obtained via projection? In *Proceedings of the 2004 American Control Conference*, volume 1, pages 113–118 vol.1, June 2004. DOI:10.23919/ACC.2004.1383589.
- [62] F. J. Mesa-Martinez, J. Nayfach-Battilana, and J. Renau. Power Model Validation Through Thermal Measurements. In *International Symposium on Computer Architecture (ISCA)*, pages 302–311, New York, NY, USA, 2007. ACM. DOI:10.1145/1250662.1250700.
- [63] K. Dev, A. N. Nowroz, and S. Reda. Power mapping and modeling of multi-core processors. In *International Symposium on Low Power Electronics and Design (ISLPED)*, pages 39–44, Sept 2013. DOI:10.1109/ISLPED.2013.6629264.
- [64] W. Huang, K. Skadron, S. Gurumurthi, R. J. Ribando, and M. R. Stan. Differentiating the roles of IR measurement and simulation for power and temperature-aware design. In *International Symposium on Performance Analysis of Systems and Software*, pages 1–10, April 2009. DOI:10.1109/ISPASS.2009.4919633.
-

- [65] K. Maize, A. Ziabari, W. D. French, P. Lindorfer, B. OConnell, and A. Shakouri. Thermoreflectance CCD Imaging of Self-Heating in Power MOSFET Arrays. *IEEE Transactions on Electron Devices*, 61(9):3047–3053, Sept 2014. DOI:10.1109/TED.2014.2332466.
- [66] Microsanj™, 2018. URL: <https://www.microsanj.com/>.
- [67] P. L. Komarov, M. G. Burzo, and P. E. Raad. CCD Thermoreflectance Thermography System: Methodology and Experimental Validation. In *International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*, pages 1–6, Sept 2006.
- [68] M. G. Burzo, P. L. Komarov, and P. E. Raad. Noncontact transient temperature mapping of active electronic devices using the thermoreflectance method. *IEEE Transactions on Components and Packaging Technologies*, 28(4):637–643, Dec 2005. DOI:10.1109/TCAPT.2005.859738.
- [69] S. Grauby, S. Dilhaire, S. Jorez, and W. Claeys. Temperature variation mapping of a microelectromechanical system by thermoreflectance imaging. *IEEE Electron Device Letters*, 26(2):78–80, Feb 2005. DOI:10.1109/LED.2004.841468.
- [70] Zs. Kohári, A. Csendes, V. Székely, and M. Rencz. Thermal investigation of monolithic structures. *Microelectronics Journal*, 28(3):317–325, 1997. DOI:10.1016/S0026–2692(96)00035–3.
- [71] A. Csendes, V. Székely, and M. Rencz. Thermal mapping with liquid crystal method. *Microelectronic Engineering*, 31(1-4):281–290, 1996. DOI:10.1016/0167–9317(95)00350–9.
- [72] A Csendes. *IC csípek és mikrostruktúrák termikus vizsgálata*. PhD thesis, Budapesti Műszaki és Gazdaságtudományi Egyetem, 1997.
- [73] V. Székely, A. Poppe, A. Páhi, A. Csendes, G. Hajas, and M. Rencz. Electro-thermal and logi-thermal simulation of VLSI designs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 5(3):258–269, Sept 1997. DOI:10.1109/92.609868.
- [74] V. Székely and K. Tarnay. Accurate algorithm for temperature calculation of devices in nonlinear-circuit-analysis programs. *Electronics Letters*, 8(19):470–472, September 1972. DOI:10.1049/e1:19720338.
- [75] V. Székely. Accurate calculation of device heat dynamics: a special feature of the TRANS-TRAN circuit-analysis program. *Electronics Letters*, 9(6):132–134, March 1973. DOI:10.1049/e1:19730098.
- [76] V. Székely and K. Tarnay. Thermal Coupling Phenomena in IC’s: Models for Analysis and Synthesis for Circuits based on Thermal Coupling. In *European Solid State Circuits Conference (ESSCIRC)*, pages 54–55, Sept 1976.
- [77] V. Székely, A. Poppe, M. Rencz, A. Csendes, and A. Páhi. Self-consistent electro-thermal simulation: fundamentals and practice. *Microelectronics Journal*, 28:247–262, 1997.
- [78] V. Székely, A. Poppe, M. Rencz, A. Csendes, and A. Páhi. Electro-thermal simulation: a realization by simultaneous iteration. *Microelectronics Journal*, 28(3):247 – 262, 1997. DOI:10.1016/S0026–2692(96)00029–8.
-



- [79] K. Németh. On the analysis of nonlinear resistive networks considering the effect of temperature. *IEEE Journal of Solid-State Circuits*, 11(4):550–552, Aug 1976. DOI:10.1109/JSSC.1976.1050774.
- [80] K. Fukahori and P. R. Gray. Computer simulation of integrated circuits in the presence of electrothermal interaction. *IEEE Journal of Solid-State Circuits*, 11(6):834–846, Dec 1976. DOI:10.1109/JSSC.1976.1050825.
- [81] Mentor® A Siemens Business ELDO, 2017. URL: [https://www.mentor.com/products/ic\\_nanometer\\_design/analog-mixed-signal-verification/eldo-platform](https://www.mentor.com/products/ic_nanometer_design/analog-mixed-signal-verification/eldo-platform).
- [82] P. Raynaud. Single kernel electro-thermal IC simulator. In *International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*, pages 356–358, Sept 2013. DOI:10.1109/THERMINIC.2013.6675231.
- [83] © Keysight Technologies HeatWave, 2017. URL: <http://www.keysight.com/en/pc-2447833/heatwave-electro-thermal-analysis-software>.
- [84] A. Timár, Gy. Bognár, A. Poppe, and M. Rencz. Electro-thermal co-simulation of ICs with runtime back-annotation capability. In *International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*, pages 1–6, Oct 2010.
- [85] A. Timár and M. Rencz. Temperature dependent timing in standard cell designs. *Microelectronics Journal*, 45(5):521 – 529, 2014. DOI:10.1016/j.mejo.2013.08.016.
- [86] G. Nagy and A. Poppe. A novel simulation environment enabling multilevel power estimation of digital systems. In *International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*, pages 1–4, Sept 2011.
- [87] G. Nagy and A. Poppe. Simulation framework for multilevel power estimation and timing analysis of digital systems allowing the consideration of thermal effects. In *Latin American Test Workshop (LATW)*, pages 1–5, April 2012. DOI:10.1109/LATW.2012.6261250.
- [88] G. Nagy, P. Horváth, L. Pohl, and A. Poppe. Advancing the thermal stability of 3D ICs using logi-thermal simulation. *Microelectronics Journal*, 46(12, Part A):1114 – 1120, 2015. DOI:10.1016/j.mejo.2015.06.025.
- [89] T. Bouhadiba, M. Moy, F. Maraninchi, J. Cornet, L. Maillet-Contoz, and I. Materic. Co-simulation of Functional SystemC TLM Models with Power/Thermal Solvers. In *IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum*, pages 2176–2181, May 2013. DOI:10.1109/IPDPSW.2013.206.
- [90] T. Bouhadiba, M. Moy, and F. Maraninchi. System-level modeling of energy in TLM for early validation of power and thermal management. In *Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1609–1614, March 2013. DOI:10.7873/DATE.2013.327.
- [91] P. Michaud, Y. Sazeides, A. Sez nec, T. Constantinou, and D. Fetis. An Analytical Model of Temperature in Microprocessors. Research Report RR-5744, INRIA, 2005. URL: <https://hal.inria.fr/inria-00070275>.
-

- [92] S. Vinco, Y. Chen, F. Fummi, E. Macii, and M. Poncino. A Layered Methodology for the Simulation of Extra-Functional Properties in Smart Systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(10):1702–1715, Oct 2017. DOI:10.1109/TCAD.2017.2650980.
- [93] SoC-Lib: an open platform for virtual prototyping of multi-processors system on chip, 2017. URL: <http://www.soclib.fr>.
- [94] T. E. Carlson, W. Heirman, S. Eyerman, I. Hur, and L. Eeckhout. An Evaluation of High-Level Mechanistic Core Models. *ACM Transactions on Architecture and Code Optimization (TACO)*, 2014. DOI:10.1145/2629677.
- [95] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta. The SPLASH-2 Programs: Characterization and Methodological Considerations. *SIGARCH Computer Architecture News*, 23(2):24–36, May 1995. DOI:10.1145/225830.223990.
- [96] J. L. Henning. SPEC CPU2006 Benchmark Descriptions. *SIGARCH Computer Architecture News*, 34(4):1–17, Sep 2006. DOI:10.1145/1186736.1186737.
- [97] C. Bienia, S. Kumar, J. P. Singh, and K. Li. The PARSEC Benchmark Suite: Characterization and Architectural Implications. In *17th International Conference on Parallel Architectures and Compilation Techniques (PACT)*, PACT '08, pages 72–81, New York, NY, USA, 2008. ACM. DOI:10.1145/1454115.1454128.
- [98] PARALUTION Labs. PARALUTION v1.1.0, 2016. URL: <http://www.paralution.com/>.
- [99] B. Murmann. ADC Performance Survey 1997-2017, 2017. URL: <http://web.stanford.edu/~murmman/adcsurvey.html>.
- [100] D.A. Patterson and J.L. Hennessy. Real Stuff: the AMD Opteron X4 (Barcelona) and Intel Nehalem Memory Hierarchies. In *Computer Organization and Design*, chapter 5.10, pages 539–543. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 4th edition, 2012.
- [101] Automatically Tuned Linear Algebra Software, 2017. URL: <http://math-atlas.sourceforge.net/>.
- [102] Linear Algebra PACKage, 2017. URL: <http://www.netlib.org/lapack/>.
- [103] Supernodal LU, 2017. URL: <http://crd.lbl.gov/~xiaoye/SuperLU/>.
- [104] K. Rupp, P. Tillet, F. Rudolf, J. Weinbub, A. Morhammer, T. Grasser, A. Jüngel, and S. Selberherr. ViennaCL—Linear Algebra Library for Multi- and Many-Core Architectures. *SIAM Journal on Scientific Computing*, 38(5):S412–S439, 2016. DOI:10.1137/15M1026419.
- [105] Spectra, 2017. URL: <https://spectralib.org/>.
- [106] Spectra, 2017. URL: <http://eigen.tuxfamily.org>.
- [107] Intel® Math Kernel Library, 2017. URL: <https://software.intel.com/en-us/mkl>.
- [108] OpenBLAS, 2017. URL: <http://www.openblas.net/>.
-

- [109] V. Székely and A. Poppe. Nemlineáris tranziens szimuláció. In *Áramkörszimuláció a PC-n*, chapter 2.4, pages 62–75. ComputerBooks Kiadói Kft., Budapest, HU, 5th edition, 1995.
- [110] IEEE Standard for Standard SystemC Language Reference Manual. *IEEE Std 1666-2011*, pages 1–638, Jan 2012. DOI:10.1109/IEEESTD.2012.6134619.
- [111] IEEE Standard for Standard SystemC(R) Analog/Mixed-Signal Extensions Language Reference Manual. *IEEE Std 1666.1-2016*, pages 1–236, April 2016. DOI:10.1109/IEEESTD.2016.7448795.
- [112] Accellera. SystemC és SystemC-AMS forráskód, 2017. URL: <http://accelera.org/downloads/standards/systemc>.
- [113] IEEE Standard for Verilog Hardware Description Language. *IEEE Std 1364-2005*, pages 1–560, April 2006. DOI:10.1109/IEEESTD.2006.99495.
- [114] IEEE Standard VHDL Language Reference Manual. *IEEE Std 1076-2008*, pages c1–626, Jan 2009. DOI:10.1109/IEEESTD.2009.4772740.
- [115] J. Ou, P. Brunmayr, F. Muhammad, J. Haase, and C. Grimm. TU Vienna Systeme AMS Communications Library Documentation, 2016. URL: [www.systemc-ams.org/download/TUV\\_AMS\\_Library.tar.gz](http://www.systemc-ams.org/download/TUV_AMS_Library.tar.gz).
- [116] IEEE Standard VHDL Analog and Mixed-Signal Extensions. *IEEE Std 1076.1-2007*, pages 1–328, Nov 2007. DOI:10.1109/IEEESTD.2007.4384309.
- [117] IEEE Standard for SystemVerilog–Unified Hardware Design, Specification, and Verification Language. *IEEE Std 1800-2012*, pages 1–1315, Feb 2013. DOI:10.1109/IEEESTD.2013.6469140.
- [118] Accellera. Verilog-AMS Language Reference Manual, 2017. URL: <http://www.accelera.org/images/downloads/standards/v-ams/VAMS-LRM-2-4.pdf>.
- [119] J. C. Butcher. Numerical Differential Equation Methods. In *Numerical Methods for Ordinary Differential Equations*, chapter 2, pages 51–135. John Wiley & Sons Ltd, West Sussex, England, 2nd edition, 2003.
- [120] E. Süli and D.F. Mayers. Initial value problems for ODEs. In *An Introduction to Numerical Analysis*, chapter 12, pages 310–360. Cambridge University Press, New York, USA, 1st edition, 2003.
- [121] Europractice, 2019. URL: <http://www.europractice-ic.com/>.
-

# Ábrák jegyzéke

1.1. A processzorok egyes paramétereinek változása 1970 és 2014 között [2] . . . . .	3
2.1. Modernizált Gajski-Kuhn-féle Y-diagram [5] . . . . .	7
2.2. Standard cellás integrált áramkörök tervezési folyamata . . . . .	9
2.3. Disszipáció makromodellezés lépései [15] . . . . .	12
2.4. Makroméretű anyagdarab hőellenállása és hőkapacitása . . . . .	17
2.5. Egy egyszerű struktúra termikus kompakt modellje . . . . .	17
2.6. Modern négymagos processzor felületén kialakuló hőmérsékleti viszonyok működés közben [63] . . . . .	19
2.7. Szimulátorcsatolás . . . . .	22
4.1. Termikus viselkedés modellezésével bővített GK diagram . . . . .	28
4.2. Az interfész által megvalósított szimulátorcsatolás . . . . .	30
5.1. Funkcionális-termikus szimulációval bővített tervezési folyamat . . . . .	33
5.2. A layout reprezentáció előállítása különböző elvonatkoztatási szinteken . . . . .	35
5.3. A disszipációmokkellek előállítása különböző elvonatkoztatási szinteken . . . . .	36
5.4. NAND20 cella dinamikus fogyasztása a kimenet 1-0 váltásakor . . . . .	37
5.5. NAND20 cella szivárgási árama . . . . .	37
5.6. Az $\alpha$ processzor layoutja RTL és logikai szinteken . . . . .	39
5.7. A $\beta$ processzor előzetes layoutja RTL szinten . . . . .	40
5.8. A $\gamma$ processzor layoutja algoritmus/architektúra szintjén . . . . .	41
5.9. Az $\alpha$ processzor fogyasztásának becslése és annak hibája órajelenként . . . . .	42
5.10. Az $\alpha$ processzor fogyasztásának becslése és annak hibája 1 $\mu$ s-onként kiértékelve . . . . .	42
5.11. A $\beta$ processzor fogyasztásának becslése és annak hibája órajelenként . . . . .	43
5.12. A $\beta$ processzor fogyasztásának becslése és annak hibája 1 $\mu$ s-onként kiértékelve . . . . .	43
5.13. A $\gamma$ processzor hivatalos TDP értéke és McPAT modellel becsült fogyasztása . . . . .	44
6.1. Szimulációs tartomány diszkrétizálása véges térfogatok módszerével . . . . .	46
6.2. Véges térfogatok módszerével felosztott 2 dimenziós tartomány . . . . .	48
6.3. Dirichlet peremfeltétel . . . . .	48
6.4. Neumann peremfeltétel . . . . .	49
6.5. Robin peremfeltétel . . . . .	50
6.6. Háromdimenziós koordinátarendszer transzformálása egydimenzióssá . . . . .	51
6.7. $\mathbf{G}$ mátrix nem nulla elemei . . . . .	51
6.8. Előrelépő Euler módszer stabilitásának határhelyzete . . . . .	53
6.9. Numerikus módszerek pontossága különböző időlépés mellett . . . . .	54
6.10. A hátralépő Euler és a trapéz módszer hibája az alkalmazott időlépés függvényében . . . . .	55

6.11. Numerikus módszerek sebessége különböző felbontások mellett . . . . .	55
6.12. Az új termikus motor validációjához használt elrendezés . . . . .	56
6.13. Az új termikus motor validációja . . . . .	57
6.14. Termikus motorok futási ideje a térbeli felbontás függvényében . . . . .	58
6.15. Az időlépés hatása a szimuláció sebességére . . . . .	58
6.16. Termikus motorok futásideje és a gyorsulás CPU magok számának függvényében . . . . .	59
6.17. A validáció során alkalmazott elrendezés időállandó-spektruma . . . . .	60
6.18. Az adaptív időlépést meghatározó algoritmus . . . . .	61
6.19. A maximum hőmérsékletek hibája 1 $\mu$ s időlépéshez viszonyítva . . . . .	61
6.20. Az időlépés változása a szimuláció során . . . . .	62
7.1. Funkcionális verifikáció a termikus viszonyok figyelembevételével . . . . .	65
7.2. A keretrendszerbe beépített és illesztett szimulátorok . . . . .	65
7.3. A logikai/funkcionális szimulátorok illesztéséhez használt típusok UML osztálydiagramja . . . . .	66
7.4. A termikus szimulátorok illesztéséhez használt típusok UML osztálydiagramja . . . . .	67
7.5. NAND20 kapu hőmérsékletfüggő késleltetése . . . . .	68
7.6. A szimuláció végén kialakult hőmérsékleteloszlások . . . . .	69
7.7. A kapuszintű és RTL modell hőmérséklet eloszlása közötti abszolút hiba . . . . .	70
7.8. A processzor kritikus – legnagyobb késleltetésű – jelútja . . . . .	70
7.9. Az ALU hőmérséklete és a kritikus út késleltetése a kapuszintű és a kevert modell esetén . . . . .	71
7.10. A processzoros rendszer által megvalósított szabályozási rendszer . . . . .	71
7.11. Az ADC és DAC hibája a hőmérséklet függvényében . . . . .	72
7.12. A lapka maximális hőmérséklete és a hőmérsékleteloszlás a szimuláció végén . . . . .	73
7.13. A PID szabályozó ugrásválasza . . . . .	73
7.14. Az alkalmazott feszültségek a frekvencia függvényében . . . . .	74
7.15. Dinamikus hőmérséklet-menedzsment (DTM) működés közben . . . . .	74
7.16. A lapkán kialakult hőmérséklet-eloszlás a szimuláció végén . . . . .	75
B.1. Előrelépő Euler stabilitási régiója a komplex számsíkon ábrázolt fehér körlap [119] . . . . .	84
B.2. Hátralépő Euler stabilitási régiója a szürke körlap kivételével a teljes komplex számsík . . . . .	85
B.3. Trapéz módszer stabilitási régiója a komplex számsík bal oldala . . . . .	86
C.1. Az $\alpha$ processzor kapuszintű modelljének hőmérséklete és fogyasztása . . . . .	88
C.2. Az $\alpha$ processzor RTL implementációjának hőmérséklete és fogyasztása . . . . .	89
C.3. Az $\alpha$ processzor RTL modelljének abszolút hibája a kapuszintű modellhez képest . . . . .	90
C.4. A $\beta$ processzor hőmérséklete és fogyasztása . . . . .	91

# Táblázatok jegyzéke

2.1. Elektromos és termikus mennyiségek analógiája . . . . .	17
4.1. Különböző absztrakciós szinteken alkalmazott együttes szimulációs módszerek megvalósítása a bemutatott interfésszel . . . . .	31
5.1. Funkcionális modellt felépítő nyelvi elemek ismert HDL-ekben . . . . .	32
5.2. Az $\alpha$ processzor layout komponenseinek mérete . . . . .	39
5.3. A $\beta$ processzor layout komponenseinek mérete . . . . .	40
5.4. A $\gamma$ processzor layout komponenseinek mérete . . . . .	41
6.1. Futási idők és a számított hiba összehasonlítása . . . . .	62
7.1. Az ADC és DAC hőmérsékletfüggő paraméterei . . . . .	71
7.2. A szabályozott szakasz paraméterei . . . . .	72
7.3. A PID szabályozó paraméterei . . . . .	72