



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar

Logi-termikus szimuláció sztenderd tervező rendszerekben

Doktori értekezés

Szerző: Timár András

Témavezető: Dr. Rencz Márta, egyetemi tanár

Elektronikus Eszközök Tanszéke
Budapest, 2013.

Családomnak.

Nyilatkozat önálló munkáról, hivatkozások átvételéről

Alulírott Timár András 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, 2013. november 13.

Nyilatkozat nyilvánosságra hozatalról

Alulírott Timár András 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, 2013. november 13.

Tartalomjegyzék

1. Bevezetés	3
1.1. THERMINATOR projekt	6
2. Logikai és termikus szimulációs eszközök	9
2.1. Logikai szimulátorok	11
2.1.1. Funkcionális hibák, hazardok	12
2.1.2. A QuestaSim logikai szimulátor tulajdonságai	14
2.2. Termikus szimulátorok	14
3. Termikus szimuláció	17
3.1. A termikus modell előállítása és szimulációja	18
3.2. A termikus ekvivalens RC hálózat számítása	21
3.3. HotSpot	24
3.4. Termikus szimuláció bemutatása egy példán	26
4. Teljesítmény és energia modellek	28
4.1. Disszipáció CMOS áramkörökben	29
4.1.1. Szivárgási áram és teljesítmény	29
4.1.2. Dinamikus és belső energia	30
4.1.3. Statikus és dinamikus fogyasztás analízis	31
4.2. A Liberty fájlformátum	32
4.3. Nem lineáris teljesítmény és időzítési modellek (NLPM, NLDM)	32
4.4. Skálázható polinom modellek (SPPM, SPDM)	34
4.5. Összetett áramforrású teljesítmény modell (CCS)	35
4.5.1. A nanométeres tervezés kihívásai	36
4.5.2. A CCS modellezés	37
4.5.3. A CCS teljesítmény modell	38
4.5.4. CCS szivárgási teljesítmény modellezés	38
4.5.5. CCS dinamikus teljesítmény modellezés	38
4.5.6. CCS összefoglalás	39
4.6. Teljesítmény karakterizáció SPICE szimulációkkal	41
5. Logi-termikus szimuláció EDA környezetben	43
5.1. A logikai- és termikus szimulátorok összekapcsolása	44
5.2. A logi-termikus szimuláció folyamata	45
5.2.1. A Verilog PLI interfész	47

5.3.	A szimuláció gyorsítása partíciókra bontással	50
5.3.1.	A particionálás hibája	52
5.4.	A szimuláció gyorsítása skálázással	56
5.4.1.	A módszer leírása	57
6.	Hőmérsékletfüggő időzítések	62
6.1.	A hőmérsékletfüggő időzítések számítása	62
6.2.	A hőmérsékletfüggő időzítési szimulációs eljárás lépései	66
6.3.	Időzítési problémák detektálása	68
6.4.	Demonstrációs áramkör	70
6.4.1.	Teljesítmény karakterizáció	71
6.5.	Teljesítménysűrűség és hőmérsékleti térkép	72
6.6.	Hőmérsékletfüggő frekvencia	77
6.7.	Validálás és verifikáció	78
6.7.1.	SPICE–logi-termikus verifikáció	78
6.7.2.	Saját megoldó algoritmus összehasonlítása SPICE szimulációval	79
6.7.3.	A frekvenciamenet SPICE verifikációja	80
6.7.4.	CellTherm és LogiTherm összehasonlítás	82
6.8.	Ipari méretű és bonyolultságú áramkörök szimulációja	84
6.8.1.	1000 inverteres gyűrűs oszcillátor	84
6.8.2.	STMicroelectronics–POLITO tesztáramkör	85
7.	Nagy felbontású hőmérsékletfüggő időzítési modell	89
7.1.	A jelenlegi időzítési modellek problémája	89
7.2.	Az új modell leírása és előállítása	90
7.3.	Az új modell bemutatása egy példán	93
7.4.	Késleltetés-hőmérséklet függvények	94
7.5.	Logi-termikus szimuláció SDF késleltetések alapján	95
7.6.	Logi-termikus szimuláció nagy felbontású hőmérsékletfüggő késleltetésekkel	98
7.7.	A hagyományos és az új modell összehasonlítása	98
7.8.	A Liberty formátum bővítése	101
8.	Összefoglalás	103
8.1.	Az eredmények gyakorlati alkalmazásai	104
A	Adat-előkészítési idő szimuláció	106
B	Liberty fájlformátum	107
C	Rövidítésjegyzék	109

1. fejezet

Bevezetés

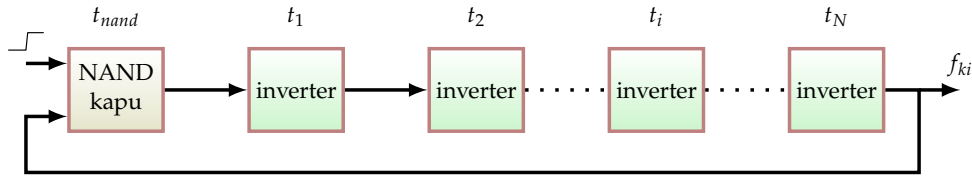
Napjaink elektronikus eszközeiben az integrált áramkörök működése egyre több termikus kérdést vet fel. A folyamatos méretcsökkenés és a számítási kapacitás növekedése miatt a mai nagy bonyolultságú, teljes rendszereket magukba foglaló integrált áramkörök (System on Chip, SoC, System in Package, SiP) megnövekedett fogyasztása miatt a hőmérsékleti kérdések a mikroelektronikai ipar egyik kulcskérdésévé váltak. Az áramkörök működés közben történő melegedése olyan problémákat vet fel, melyek már nem oldhatók meg egyszerűen a nyers erő módszerével, a hűtési rendszerek egyre magasabb szintű továbbfejlesztésével. Az ezredforduló környékén az integrált áramkörök termikus problémáit úgy próbálták meg kezelni, hogy a tokon egyre nagyobb méretű és teljesítményű aktív hűtést (hűtőbordát, ventilátort) alkalmaztak. Napjainkban ez a fajta túlmelegedés elleni védelem már olyan hűtőrendszerek létrejöttét hozta, melyek akár folyékony nitrogénnel vagy héliummal igyekeznek az áramkörök hőmérsékletét alacsonyan tartani[1]. Mára már világossá vált, hogy az áramkörök túlhevülésének kezelését más módszerekkel is segíteni kell.

A melegedési problémák azért kritikusak, mert bizonyos hőmérséklet felett a félvezető eszközökben olyan változások jönnek létre, melyek az integrált áramkör működését megváltoztatják. Az elektronikus eszközökben minden alkotóelem működése valamilyen mértékben hőmérsékletfüggő. A hőmérsékletfüggések miatt az eszközök működési karakterisztikái megváltoznak és ezeket a változásokat a lehető legpontosabban modellezni kell, hogy már a tervezéskor számolni lehessen velük.

Az integrált áramkör egy kis területén jelentkező lokális (túl)melegedést angol terminológiával *hot-spot*-nak, forró pontnak nevezünk. Ezek a forró pontok az áramkör adott részén ronthatják el a működést, megváltoztathatják a tranzisztorok karakterisztikáját, az időzítési- és jelintegritást, melynek következtében az áramkör funkcionálisan működésképtelenné válik. Az ilyen hibalehetőségek időben, jóval a gyártás előtt történő detektálása rendkívül fontos.

A hőmérséklet változásának hatását a működésre egyértelműen demonstrálni lehet egy egyszerű gyűrűs oszcillátor áramkörös. A gyűrűs oszcillátor egy lehetséges sematikus vázlatát mutatja be az 1.1. ábra. A bemutatott gyűrűs oszcillátor egy NAND kaput és páros számú invertert tartalmaz sorba kapcsolva. Az utolsó inverter kimenetét visszacsatolva a bemenetre állandó oszcilláció mérhető a kimeneten a NAND és inverter kapuk véges késleltetése miatt.

A gyűrűs oszcillátor kimeneti frekvenciája (1.1) alapján számítható, ha az inverterek késlel-



1.1. ábra. Egy gyűrűs oszcillátor sematikus vázlatja

tetését egyformának vesszük.

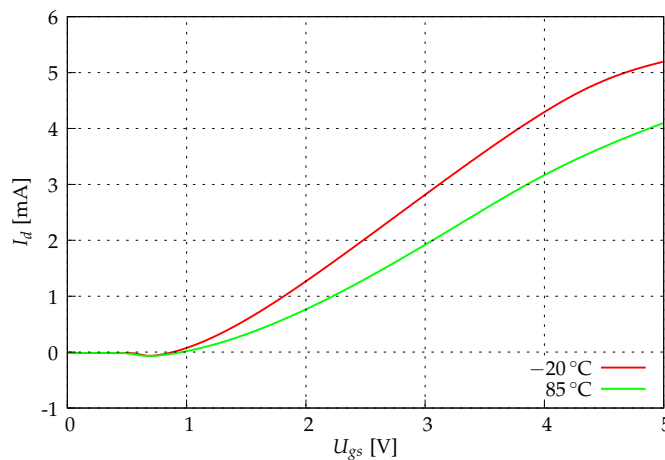
$$f_{ki} = \frac{1}{2 \cdot (t_{nand} + N \cdot t_i)}, \quad (1.1)$$

ahol t_{nand} a NAND kapu késleltetése, t_i egy inverter késleltetése és N az inverterek száma ($N = 2k, k \in \mathbb{N}^+$), f_{ki} a gyűrűs oszcillátor kimeneti frekvenciája.

Számításokkal és kísérletekkel is bizonyítható, hogy a gyűrűs oszcillátor kapuinak késleltetése megváltozik a hőmérséklet változás hatására. Ennek oka az, hogy a tranzisztorok áram-feszültség karakterisztikája a hőmérséklet változásának hatására eltolódik. A MOS tranzisztor karakterisztikájának hőmérsékletfüggését a töltéshordozó mozgékonyosság (μ_0) és a küszöbfeszültség (V_t) hőmérsékletfüggése okozza [2, 505.o.]. A MOS tranzisztor drain áramának hőmérsékletfüggése tehát (1.2) szerint

$$\frac{\partial I_d}{\partial T} = -\frac{3}{2} \frac{I_d}{T} - \frac{2I_d}{(U_{gs} - V_t)} \cdot \frac{\partial V_t}{\partial T}. \quad (1.2)$$

(1.2) alapján megállapítható, hogy a hőmérséklet növekedésével I_d áram csökken, tehát az egységnyi idő alatt a drain-en átfolyó töltés csökken, ami azt jelenti, hogy a kimeneti terhelés kapacitását több idő alatt tudja csak feltölteni, tehát megnő a kapu késleltetése. A kapukésleltetés növekedésével (1.1) alapján a kimeneti frekvencia lecsökken. Egy nMOS tranzisztor transzfer karakterisztikáját mutatja az 1.2. ábra -20°C -ra és 85°C -ra. Látható, hogy magasabb hőmérsékleten a drain áram lecsökken, tehát egységnyi idő alatt kevesebb töltés áramlik át a drainen.



1.2. ábra. nMOS FET tranzisztor transzfer karakterisztikája különböző hőmérsékleteken

Mint a bemutatott példából is látszik, minden integrált áramkörben a hőmérséklet változása az eszköz működésének megváltozásával jár. Az integrált áramkörök hőmérsékletfüggő működése tehát a növekvő teljesítmények és működési frekvenciák miatt kiemelt fontosságú és újabb megoldásokat igénylő tématerület. Erre a problémára jelent egyfajta megoldást az integrált áramkörök logikai és termikus együttes szimulációja, mellyel disszertációm is foglalkozik.

Az eszközök hőmérsékletének növekedésével egyre több működési hiba lép fel, és egy bizonyos hőmérséklet felett a félvezető alapú integrált áramkör végleg meghibásodik. Napjainkban az áramkörök melegedési problémáit különböző architektúrális szinteken próbálják meg kezelni. Ilyen megoldások például a dinamikus feszültség és frekvencia skálázás (Dynamic Voltage and Frequency Scaling, DVFS), dinamikus termikus menedzsment (Dynamic Thermal Management, DTM), dinamikus hőmérsékletfüggő időzítés (Dynamic Thermal Aware Scheduling, DTAS), órajel kapuzás (clock gating), feladatok és szálak újraelosztása, migrálása más processzormagra. Ezek a megoldások az áramkör által végzett feladatokat úgy ütemezik és osztják ki, hogy a hőmérséklet eloszlása a rendszerben homogén legyen [3].

A mai digitális integrált áramkörök, mint a többmagos processzorok, DSP-k, mobil eszközök SoC processzorai, stb. úgynevezett *sztenderd cellás* tervezési módszerrel készülnek. A *sztenderd cellák* olyan RTL (Register Transfer Level) szintű építőelemek, melyek különböző logikai műveleteket és sorrendi hálózati funkciókat valósítanak meg. Ilyenek például az egyszerű logikai kapuk (ÉS, VAGY, NEM-ÉS kapuk, stb.), multiplexerek, dekóderek, flip-flop cellák. A felhasználható *sztenderd cellák* egy *cellakönyvtárban* helyezkednek el és különböző tervező szoftverek segítségével készül el belőlük a digitális integrált áramkörök terve.

A digitális integrált áramkör tervezés ma igen magas absztrakciós szinten történik. A magas szintű viselkedést leíró nyelvek mint a SystemC, Algorithmic C, SystemVerilog, SystemVHDL mind olyan leírónyelvek, melyek segítségével magas szinten lehet egy digitális integrált áramköri rendszert megtervezni és verifikálni. A magas szintű leíró nyelveken megfogalmazott működést szinte teljesen automatizáltan lehet a különböző szintézis alkalmazások segítségével *sztenderd cellás* tervvé alakítani. A szintézis során a specifikált működést megvalósító *sztenderd cellás* digitális integrált áramkör terve készül el. Amíg a tervezés idáig eljut, nagyon sok vizsgálatra, szimulációra, optimalizációra és iterációra van szükség annak érdekében, hogy a legyártott áramkör a valóságban is úgy működjön, ahogy a tervezőasztalon specifikálták.

A magas szinten megtervezett bonyolult integrált áramkörök funkcionális és időzítési vizsgálatát logikai szimulációs rendszerekkel lehet elvégezni. A működés hőmérsékletfüggése miatt a logikai szimulációkat a hőmérsékleti hatások figyelembe vételével kellene végrehajtani. Szükség van a digitális integrált áramkörökben kialakuló hőmérsékleti viszonyok feltérképezésére, mivel a hőmérsékletek változása az áramköri működés megváltozásával jár. Ezekkel a hatásokkal már a tervezés fázisában számolni kell.

Jelenleg az iparban nem áll rendelkezésre olyan eszköz, amely a logikai működést és az időzítési viszonyokat úgy képes modellezni és szimulálni, hogy figyelembe veszi a termikus viszonyokat. Ennek oka az, hogy az ilyen típusú szimulációk rendkívül számításigényesek, ezért olyan szimulációs eljárást és modelleket kell kidolgozni, amelyek képesek a logikai működés és a termikus hatások együttes vizsgálatára elfogadható ideig tartó szimulációkkal. Olyan eszközre van szükség, amely mind a saját melegedést, mind a környezeti hőmérsékletek változását figyelembe tudja venni a logikai szimuláció során és számítási ideje közel esik egy tisztán logikai szimulációéhoz.

Ilyen modellek és szimulációs eszközök kutatására és fejlesztésére jött létre az STMicro-

electronics elektronikai gyártó cég igényeinek kielégítésére a THERMINATOR integrált projekt, ami nagyban épített a jelen értekezésben bemutatott eredményekre.

Értekezésemben olyan modellezési és szimulációs eljárásokat mutatok be, melyek segítségével lehetőség nyílik digitális integrált áramkörök logikai és termikus együttes vizsgálatára, szimulációjára. A továbbiakban az ilyen együttes szimulációt *logi-termikus* szimulációnak hívom.

1.1. THERMINATOR projekt

A modern nanoméretű mikroelektronikai integrált áramkörök termikus problémáinak vizsgálata és ezek megoldása aktuális és fontos tématerület. Értekezésemben olyan eljárásokat és módszereket mutatok be, melyeket sikerrel alkalmaztak az európai FP7-es keretrendszerbeli THERMINATOR integrált projektben[4, 5]. A hároméves projekt 2010-ben indult és célja olyan innovatív tervezési módszerek kutatása és fejlesztése, melyek segítségével a legmodernebb mikroelektronikai integrált áramkörök és rendszerek termikus modellezése, és szimulációja válik lehetővé.

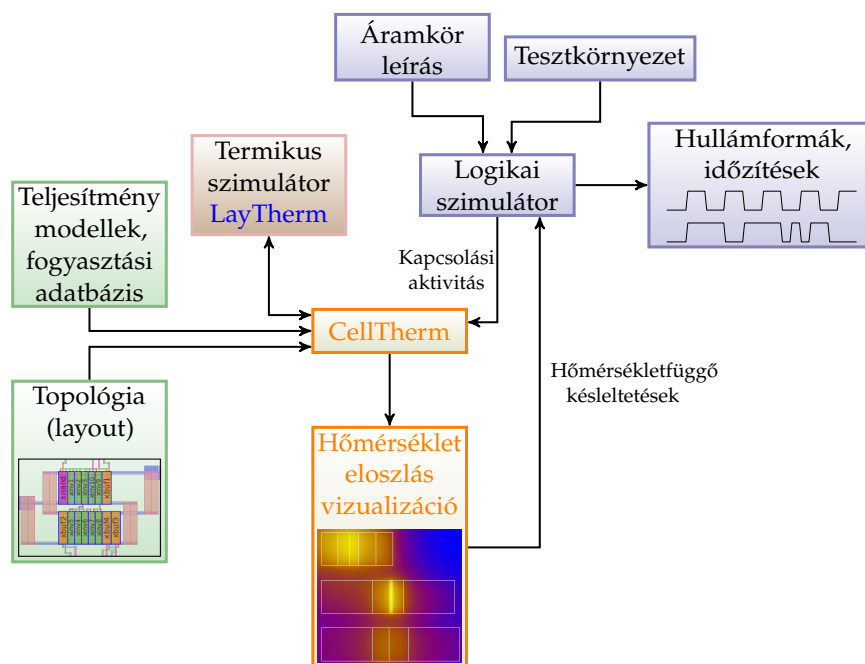
A projektben csaknem 20 nemzetközi cég és felsőoktatási intézmény vett részt, többek között az NXP, az Infineon, a Synopsys, a Torinói Műszaki Egyetem és a Bolognai Műszaki Egyetem. A projekt fő koordinátora az STMicroelectronics cataniai telephelye, ahol integrált áramköri gyártósorok is üzemelnek. Az STMicroelectronics gyárában 65 nm-es csíkszélességen gyártanak modern analóg és digitális integrált áramköröket. A THERMINATOR projektben a Budapest Műszaki és Gazdaságtudományi Egyetem Elektronikus Eszközök Tanszéke elektrotermikus szimulátorok kutatásával és kifejlesztésével vett részt. Az értekezésemben bemutatott eljárásokat és szimulációs megoldásokat a THERMINATOR projekt keretében ipari bonyolultságú, nanométeres csíkszélességű integrált áramkörök szimulációjára használták. Az általam kidolgozott és bemutatott eljárások olyan modelleket használnak fel, melyek a nanométeres méretcsökkentésből adódó másodlagos, kvantumfizikai hatásokat is figyelembe veszik.

A THERMINATOR projektben alapkövetelmény volt, hogy a különböző tervező és gyártó cégek és partnerek kutatási és fejlesztési eredményei minden partner számára elérhetők és feldolgozhatók legyenek. Ennek érdekében a kidolgozott eljárások implementációját szabványos eszközökkel és környezetekben kellett végezni. Az általam kidolgozott szimulációs eljárás illeszkedik a modern EDA (Electronic Design Automation) iparban használt szabványos környezetekhez és alkalmazásokhoz.

Disszertációmban a digitális sztenderd cellás áramkörök melegedési problémáival illetve azok megoldásának lehetőségeivel foglalkozom. Munkámban olyan eljárásokat mutatok be, melyek segítségével a digitális integrált áramkörökben kialakuló hőmérsékleti állapotokat fel lehet térképezni és a kialakult hőmérséklet eloszlás elemzésével különböző, a funkcionális működést befolyásoló, időzítési- és jel-integritási problémákat fel lehet tárni még a tervezés fázisában. Értekezésemben olyan szimulátort mutatok be, mely a logikai és termikus szimulátorok tulajdonságait ötvözi és melynek segítségével a logikai működésből következtetni lehet a digitális integrált áramkörben kialakuló hőmérsékleti viszonyokra. A logi-termikus szimulációk segítségével lehetőség nyílik a logikai működésből adódó melegedési problémák kezelésére.

A logi-termikus szimuláció elvi működését mutatja be vázlatosan az 1.3. ábra. A logi-

termikus szimulációban központi szerepet játszik a termikus és a logikai szimulátor. A kettő között az általam megtervezett, kifejlesztett és az értekezésben bemutatott *CellTherm* alkalmazás teremti meg a kapcsolatot. A logikai szimulátor bemeneti adatként a szimulálandó áramkör leírását és a teszt környezetet (testbench) kapja meg Verilog formátumban. A *CellTherm* alkalmazás a kapcsolási aktivitást szimuláció közben folyamatosan frissíti a logikai szimulátor adatai alapján. A fogyasztási adatbázisból a kapcsolásokhoz tartozó energiákat kinyerve a *CellTherm* a termikus szimulátor segítségével kiszámítja az IC felületén kialakuló hőeloszlást. A hőmérsékleti hatások miatt megváltozott időzítéseket ezután visszaküldi a logikai szimulátornak, ami ezekkel számol tovább.



1.3. ábra. A logi-termikus szimuláció egyszerűsített vázlata

Disszertációmban a bevezető fejezetet követően a 2. fejezetben ismertetem a tudományterület jelenlegi állását (state-of-the-art) és röviden bemutatom az iparban alkalmazott logikai és termikus szimulációs eszközöket, melyek különbözőképpen szimulálják az áramkörök melegedési hatásait (végelelemes módszer (FEM), véges differenciák módszere (FDM), Fourier módszer stb.).

A 3. fejezetben bemutatom az Elektronikus Eszközök Tanszékén kifejlesztett termikus szimulátort és a termikus kompakt RC modelleket létrehozó eljárást melyeket felhasználok az értekezésemben bemutatott logi-termikus szimulátorban.

A 4. fejezetben bemutatom az iparban jelenleg használt teljesítmény és energia modelleket, melyek alapvető fontosságúak egy logikai és termikus együttes szimuláció során. A teljesítmény modellek segítségével lehet szimulációs úton meghatározni egy áramkör működés közben felvett teljesítményét, mely az eszköz melegedését okozza. A disszipált teljesítmény ismerete nélkül a termikus szimuláció nem lenne lehetséges.

Az 5. fejezetben módszert mutatok be, melynek segítségével a sztenderd cellás digitális integrált áramkörök logi-termikus szimulációja végrehajtható.

A 6. fejezetben a digitális integrált áramkörök hőmérsékletfüggő időzítési viszonyaival

foglalkozom. Módszert mutatok be az áramkör hőmérsékletfüggő késleltetéseinek szimulációba történő visszacsatolására. Ennek segítségével tranziens időben vizsgálható a sztenderd cellák késleltetés-változása és az időzítések integritása.

A 7. fejezetben új modellt mutatok be, melynek segítségével az integrált áramkörök logi-termikus szimulációja során a cellák hőmérséklettől függő időzítéseit hőmérséklet-késleltetés függvényekkel lehet figyelembe venni. A jelenlegi hőmérséklet-késleltetés modellek (pl. SDF) csak korlátozottan veszik figyelembe a hőmérsékleti hatásokat. Az iparban jelenleg használt modellek jellemzően néhány előre meghatározott hőmérsékleti szélsőértéken rögzítik az időzítési paramétereket, így a cellák hőmérséklettől függő időzítési adatai nem állnak rendelkezésre egy széles hőmérsékleti tartományban. A sztenderd cellák hőmérséklettől függő késleltetéseit előzetesen karakterizálva ezek a függvények felhasználhatóak a hőmérsékletfüggő időzítések jelenleginél pontosabb modellezésére és szimulációjára tetszőleges hőmérsékleti tartományban.

A 8. fejezetben összefoglalom az elért eredményeket és javaslatot teszek a további kutatási irányokra, fejlesztési lehetőségekre.

2. fejezet

Logikai és termikus szimulációs eszközök

A félvezető ipar rohamos fejlődésével a modern integrált áramkörök méretei egyre csökkennek, számítási teljesítményük egyre növekszik. A Gordon Moore által először 1965-ben publikált trend[6] –mely szerint az integrált áramkörökön megvalósított tranzisztorok száma 18 hónaponként megduplázódik– kisebb korrekciókkal a mai napig helytálló [7]. A mai legmodernebb processzorokon akár 1,5 milliárd tranzisztor is található. A 2010-ben bejelentett UltraSPARC® T3 sorozatú 16 magos, 2117 kivezetést tartalmazó SoC (System on Chip) processzor több mint 1 milliárd tranzisztort tartalmaz [8]. A processzort a TSMC 40 nm-es technológiáján valósították meg. A chip mérete 377 mm².

Ilyen tranzisztorsűrűség és magas számítási teljesítmény mellett a disszipált teljesítmény is számottevő, melynek következtében az integrált áramkör melegevé is jelentős. Az ilyen áramkörök melegedési problémáinak kezelésére már a tervezés korai fázisában fel kell készülni. Az áramkör működés közbeni melegedése olyan működési hibákat okozhat, melyekkel számolni kell jóval a gyártás előtt. Ha a tervezés során nem tartjuk szem előtt az áramkör termikus viselkedését, akkor egy hőmérsékleti viszonyokból adódó esetleges működési hibára csak a gyártás után derül fény.

A modern mikroelektronikában a folyamatos méretcsökkentés újabb és bonyolultabb fizikai effektusok figyelembevételét igényli. A nanométeres csíkszélességű (akár 22nm, 2013-ban) tranzisztorok térhódításával a digitális sztenderd cellás áramkörök is egységnyi területen nagyságrendekkel több cella elhelyezését teszik lehetővé. A méretek csökkenése és az előtérbe kerülő kvantumfizikai hatások miatt az egységnyi területre eső fogyasztási értékek megközelítik egy sugárhajtómű teljesítménysűrűségét. A nanométeres csíkszélességű mikroelektronikai cellák kvantumfizikai hatásainak figyelembe vételére (ballisztikus transzport, küszöbfeszültség közelében történő működés, növekvő szivárgási áramok, „forró elektronok”, stb.) megfelelő modelleket kellett a mikroelektronikai iparnak kitalálnia. Ilyen modellt és leíró formátumot biztosít a 4. fejezetben bemutatott Liberty formátum és a CCS modell.

A THERMINATOR projekt keretében 65 nm-es csíkszélességű, ipari méretű és bonyolultságú sztenderd cellás digitális áramkörök logikai és termikus szimulációjához kellett eljárást kidolgoznom. A projekt fő koordinátora, az STMicroelectronics olaszországi kirendeltségével és a Torinói Műszaki Egyetemmél (Politecnico di Torino) együttműködésben az általam kidolgozott és az értekezésben bemutatott eljárásokat és modelleket ipari áramkörökön használták fel.

A napjainkban gyártott legtöbb digitális integrált áramkör *sztenderd cellás* tervezési könyvtáron alapul. A sztenderd cellák olyan építőelemei az áramkörnek, melyek logikai, kombinációs

és szekvenciális műveleteket valósítanak meg. Az ilyen cellák előre megtervezett és karakterizált tranzisztoros kapcsolásokat tartalmaznak, melyek a kívánt funkciókat látják el. A cellák felületi topológiája is egy bizonyos szabályrendszert követ. Jellemzően a cellák magassága állandó, a táp- és föld sínek előre definiált helyeken futnak, így a cellák egymás mellé helyezésével automatikusan létrejön a táp-, és föld sínek kapcsolata. Az így létrehozott cellákat változtatás nélkül, diszkrét építőelemekként lehet felhasználni a digitális rendszerek tervezésénél.

A sztenderd cellás tervezést magas szinten támogatják az ipari szabványok és tervező alkalmazások. Jellemzően egy összetett működést megvalósító digitális integrált áramkört (pl. processzor, DSP, stb.) magas absztrakciós szinten terveznek meg. A magas absztrakciós szinten teljesen elvonatkoztatunk a kapcsolástechnikától, architektúrától, az áramkör felépítésétől és a kívánt funkciót közvetlenül határozzuk meg valamilyen magas szintű programozási vagy matematikai nyelven. Az ilyen magas szintű tervezés előnye, hogy a tervező a megvalósítandó funkcióra összpontosíthat, anélkül, hogy figyelembe venné a megvalósítás részleteit. Az adott funkció implementációját és verifikációját szoftveresen automatizálják különböző magas szintű szintézis eszközökkel. A sztenderd cellák elhelyezését, huzalozását, a terv logikai szimulációját, az időzítési és jel-integritási szimulációkat felett szoftveres alkalmazások támogatják és automatizálják.

Egy logikai szimulátor használatával meg lehet figyelni a digitális áramkör működését. A logikai szimulátorok fel vannak készítve az áramkör időzítéseinek elemzésére és akár az elhelyezett és vezetékezett¹ (Place & Route), kivezető tappancsokkal ellátott teljes áramkör logikai, funkcionális és időzítési vizsgálatára is.

A logikai szimulátorok azonban nem veszik figyelembe a szimulált áramkörök melegedési tulajdonságait (önmelegedés, időzítések megváltozása a hőmérséklet változás miatt, fogyasztás, stb), így olyan módszerre van szükség, mely a logikai és funkcionális működést a hőmérsékleti tényezők figyelembevételével tudja ellenőrizni. Az ilyen jellegű szimulációval a tényleges, legyártott áramkör működését az eddiginél pontosabban lehet modellezni és szimulálni.

A hőmérsékletfüggés beépítése a logikai szimulációkba arra is felhasználható, hogy az elhelyező és huzalozó algoritmusok peremfeltételeit adaptívan változtassa a működési hőmérséklet függvényében. Ha a sztenderd cellákat a működés közben kialakuló hőmérsékletek ismeretében helyezzük el és vezetékezzük a szeleten, akkor a gyártás után esetlegesen jelentkező, hőmérsékleti viszonyok miatt kialakuló hibák jelentős része megszüntethető.

Disszertációmban olyan eljárást mutatok be, melynek segítségével a logikai és termikus szimuláció együttesen hajtható végre. A logi-termikus szimuláció első sorban abban különbözik az elektro-termikus szimulációtól, hogy a tranzisztor szintnél magasabb absztrakciós szinten vizsgálja az áramkör működését. Egy digitális áramkör SPICE alapú tranzisztoros szimulációjából előállnak az áramkör pontos fogyasztási adatai, melyek segítségével meghatározható az áramkör termikus viselkedése is. Azonban már egy egyszerű, néhány cellát tartalmazó digitális áramkör tranzisztor szintű szimulációja is nagyságrendekkel időigényesebb, mint a logikai szimuláció. Ma általában a logikai szimulációnál magasabb absztrakciós szinten vizsgáljuk az áramkör működését. Ezen a szinten már a viselkedési leírásból *szintézissel* előállított *strukturális leírást* szimuláljuk. A logikai és fizikai szintézis során a viselkedést specifikáló magas szintű leírásból automatizáltan készül egy strukturális leírás, mely ugyanazt a logikai működést valósítja meg, mint a specifikáció, de a megvalósítás sztenderd cellákkal történik.

¹ Az IC-n a sztenderd cellák elhelyezését és összekötését automatikusan, a felhasználó által definiált kényszerek alapján szoftveresen automatizálják.

Értekezésemben olyan logi-termikus szimulátort mutatok be, melynek segítségével a szten-derd cellás digitális integrált áramkör felületén kialakuló hőmérséklet eloszlás a logikai szimu-lációval egyidőben nyomon követhető és a kiugróan magas hőmérsékletű területek (hot-spotok) felderíthetőek. A szimulátor a hőmérsékleti- és teljesítmény-eloszlás térképet a logikai szimulá-ció közben folyamatosan frissíti, így ezek változása időben és térben is nyomon követhető.

A logi-termikus szimuláció elvégzéséhez különböző eszközökre van szükség. A tisztán funkcionális és időzírtési vizsgálatához szükség van egy logikai szimulátorra, mellyel az integ-rált áramkör funkcionális működését lehet szimulálni. Az áramkör termikus viselkedésének szimulációja egy termikus szimulátorral végezhető el. A termikus szimulációhoz a bemenő teljesítmény adatok a logikai szimulátorból kinyert kapcsolási aktivitás ismeretében teljesít-mény és energia modellek segítségével határozhatók meg. A teljesítmény és energia modellek a lehető legjobban kell, hogy modellezzék a nanométeres integrált áramkörök másodlagos fizikai effektusait.

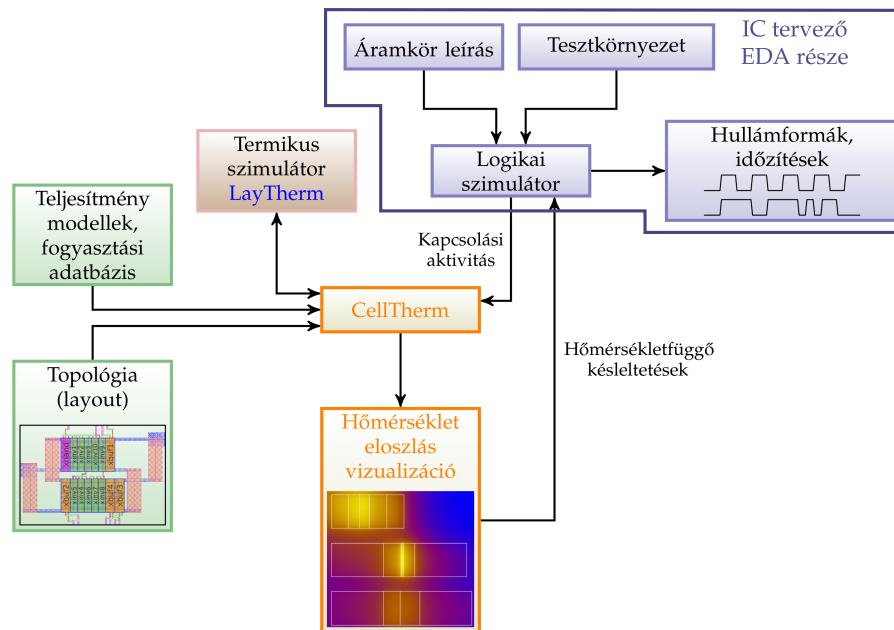
A logi-termikus szimulációhoz tehát egy logikai és egy termikus szimulátor, valamint megfelelő teljesítmény és energia modellek szükségesek. Az értekezésemben bemutatott logi-termikus szimulációs eljárás az iparban meglévő és kiforrott logikai és termikus szimulátorokat használ fel. A termikus számításokat szintén szabványos és az iparban elterjedt teljesítmény és energia modellek segítségével végzem. Munkám központi témája ezen szimulátorok és mo-dellek olyan módon történő összekapcsolása, melynek segítségével az integrált logi-termikus, együttes szimuláció szabványos, ipari EDA környezetben valósul meg.

A következő szakaszokban az iparban jelenleg használt szimulátorokat és modelleket tekintem át.

2.1. Logikai szimulátorok

A logikai szimulátor helyét a logi-termikus szimulációban a 2.1. ábra mutatja. A logikai szi-mulátorok az integrált áramkörtervező rendszerek részét képezik és a logi-termikus szimuláció logikai oldala a segítségükkel valósul meg.

A logikai szimulációkkal digitális áramkörök működését lehet vizsgálni. Egy digitális integrált áramkör tervezésében kulcsfontosságú szerepet tölt be a logikai szimulátor. A fejlesztő környezetek rendszerint egy ilyen szimulátor köré épülnek, lehetőség van a környezeten belül magas szintű viselkedési leírást készíteni, és azt ellenőrizni. Az integrált fejlesztőrendszerek előnye, hogy egy felületen lehet a forráskódot szerkeszteni és az időzírtési szimulációkat elvégezni. A digitális tervezés során a fejlesztő úgy tudja a viselkedési leírást elkészíteni, hogy annak helyes működését folyamatosan képes szimulációkkal ellenőrizni. Kezdetben, a működés vizsgálatát jellemzően zérus vagy egységnyi kapukésleltetésekkel végzik. Ekkor pusztán a logikai működést kell tesztelni. A fejlesztés későbbi szakaszában lehetőség van tényleges időzírtési adatokkal is szimulálni a tervet. Egy digitális szinkron áramkörben a különböző jelek egymáshoz viszonyított időzírtése kulcsfontosságú. Csak akkor lesz a valóságban is működőképes az áramkör, ha az időzírtések megfelelnek a specifikációnak. Az ilyen időzírtési vizsgálatokat lehet elvégezni a logikai szimulátorral.



2.1. ábra. A logikai szimulátor helye a logi-termikus szimulációban

2.1.1. Funkcionális hibák, hazardok

A digitális integrált áramkörökben fellépő hazardjelenségeket a rendszer alkotóelemeinek különböző időzítése, késleltetése okozza. A hazardoknak különböző típusai fordulhatnak elő a digitális rendszerekben. Ezek közül több már a logikai tervezés szintjén kiküszöbölhető, azonban vannak olyan hazardok, melyek a logikai tervezés során nem szüntethetők meg. A hazardok e két fajtáját ismerteti a következő felsorolás:

- A logikai tervezés szintjén kiküszöbölhető
 - statikus hazard,
 - dinamikus hazard,
 - kritikus versenyhelyzet,
 - órajel-elcsúszás (clock skew).
- A logikai tervezés szintjén nem kiküszöbölhető
 - funkcionális hazard,
 - lényeges hazard.

Amikor egy viselkedési leírásból egy szintézer alkalmazás automatizáltan elkészíti a kapusintű, strukturális leírást, akkor a logikai tervezés szintjén kiküszöbölhető hazardokat hazardmentesítő elemekkel egészíti ki. A logikai tervezés szintjén nem megszüntethető hazardok felderítését és utólagos kezelését segíti a logikai szimulátor, melyben időzítési analíziseket lehet végezni az áramkörön.

A logikai szimulátorban nem csak a tervezési folyamat elején lehet vizsgálatokat végezni, hanem az áramkör szintézise és a tényleges kialakítás elkészülte után úgynevezett elhelyezés

és huzalozás utáni (post-layout) szimulációk is végrehajthatók. A post-layout szimulációkban már a sztenderd cellák és a jelutak karakterizált, tényleges késleltetéseivel, időzítéseivel lehet számolni. Az ilyen szimuláció már megfelelően közelíti a fizikai működést és jelentősen lecsökkenti a gyártás utáni, tervezésből adódó hibák lehetőségét.

A magas szintű viselkedési leírás automatikus szintézise során ún. *hardver leíró nyelven* (Hardware Description Language, HDL) készül el a strukturális leírás. A két legismertebb és az iparban használt hardver leíró nyelv a *Verilog HDL*[9]² és a *VHDL*[10]³. Mindkét nyelv szabványos specifikációja szabadon elérhető. Ezek a hardver leíró nyelvek közvetlenül hardverre alakítható módon írják le az áramkörök működését.

A digitális elektronikai iparban rendkívül sokféle logikai szimulátor van használatban. A "három nagy" elektronikai tervezőrendszer-gyártó cég, a Mentor Graphics, a Cadence Design Systems és a Synopsys is rendelkezik saját fejlesztésű logikai szimulátorral.

- Mentor Graphics: ModelSim vagy QuestaSim
- Cadence:
 - IUS/NCSim (Verilog, VHDL és SystemVerilog)
 - NCSC (SystemC)
 - AMS Designer (Verilog/AMS, VHDL/AMS)
- Synopsys: VCS

A két legismertebb FPGA (Field Programmable Gate Array) gyártónak (Altera és Xilinx) is van saját logikai szimulátor alkalmazása. Ezek a logikai szimulátorok közvetlenül elérhetőek a tervező keretrendszerből.

- Xilinx: ISE Webpack
- Altera: Quartus II Design Framework, ModelSim-Altera

Léteznek nyílt forráskódú, ingyenes logikai szimulátorok is, mint például az Icarus[11]. A felsorolt logikai szimulátorok mindegyike támogatja a Verilog HDL és VHDL nyelven írt viselkedési és strukturális leírásokat. A szintetizált sztenderd cellás áramkörök logi-termikus szimulációjához olyan logikai szimulátorra van szükség, mely támogatja az áramkör szintetizált strukturális leírásának cella szintű logikai szimulációját. Ezt a funkciót a felsorolt logikai szimulátorok támogatják. A cella szintű logikai szimuláció biztosítja, hogy a tranzistor szintnél magasabb absztrakciós szinten lehessen az áramkör viselkedését szimulálni.

Az általam kidolgozott logi-termikus szimulációs eljárás megvalósításakor azt tartottam szem előtt, hogy az eljárás szorosan illeszkedjen a meglévő és az iparban széles körben használt szimulációs eljárásokhoz és szabványokhoz. Az eljárást úgy dolgoztam ki, hogy azt bármely említett logikai szimulátor segítségével végre lehessen hajtani az implementáció megváltoztatása nélkül. A módszer segítségével az általam megvalósított alkalmazás bármely logikai szimulátorral kompatibilis és transzparensen futtatható.

² IEEE Std 1364-2005 – IEEE Standard for Verilog Hardware Description Language.

³ IEEE Std 1076-2008 – IEEE Standard VHDL Language Reference Manual.

A logikai- és termikus szimulátorok összekapcsolását úgy terveztem meg, hogy az a Verilog HDL nyelv szabványos programozói interfészét használja fel. Az interfész neve Verilog PLI (Programming Language Interface), melyet az 5.2.1. alszakaszban ismertetek. Eljárásom implementációjához a Mentor Graphics cég QuestaSim logikai szimulátorát használtam, mely az Elektronikus Eszközök Tanszékén rendelkezésre állt és támogatja a Verilog PLI programozói interfészt.

2.1.2. A QuestaSim logikai szimulátor tulajdonságai

A munkám során felhasznált logikai szimulátor, mely szerves részét képezi a logi-termikus szimulátornak az ipar egyik legmodernebb logikai szimulátora. A QuestaSim alkalmazás a digitális integrált áramkörök tervezését és logikai szimulációját több funkcióval segíti:

- több leíró nyelv és szabvány támogatása (Verilog, VHDL, SystemVerilog, SystemC),
- a HDL leírás hibakeresése (debug) egységes környezetben,
- időzítési analízis, hazárdok, időzítési hibák felderítése,
- nagy számítási kapacitás, szimulációk futtatása több processzormagon,
- szabványos verifikációs eljárások támogatása,
- teszt környezet automatizálás,
- automatikus teszt vektor generálás (ATPG) lefedettség vizsgálata (coverage analysis).

Hangsúlyozni kell azonban, hogy a felsorolt tulajdonságok mellett a logikai szimulátorok nem képesek a logikai szimulációkat úgy futtatni, hogy az áramkör működése által okozott hőmérsékleti változásokat is figyelembe vegyék. Így tehát a tisztán logikai szimuláció az áramkör hőmérsékleteinek figyelmen kívül hagyásával történik. Kutatásomban ezért ötvöztem a logikai és termikus szimulátorok kedvező tulajdonságait, melynek segítségével a logikai szimulációk a hőmérsékleti hatásokat is figyelembe tudják venni.

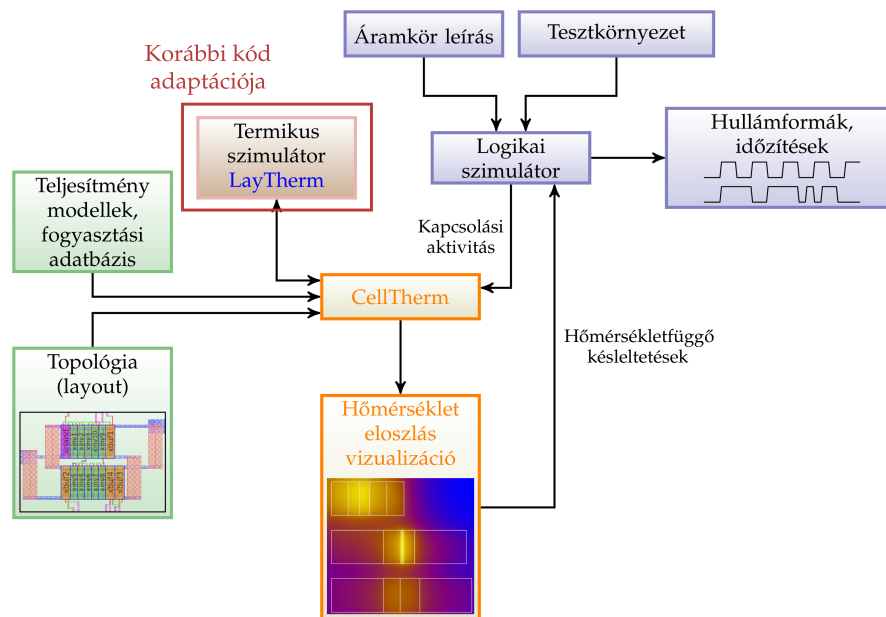
2.2. Termikus szimulátorok

A logi-termikus szimulációhoz elengedhetetlen egy termikus szimulátor is, mely a logikai szimuláció alapján kiszámolt fogyasztásokból határozza meg az áramkör hőmérsékleti viszonyait. Termikus szimulációkat többféleképpen is végre lehet hajtani. A különböző módszerek között a kívánt felbontás, a szimulációs idő és a pontosság alapján lehet választani. A termikus szimulációknak létezik véges elemes (Finite Element Method, FEM), véges differencia (Finite Difference Method, FDM), véges térfogat modellel megvalósított, vagy Fourier módszeres, stb. megoldása is.

A termikus szimulátor helyét a logi-termikus szimulációban a 2.2. ábra mutatja.

Az iparban jelenleg használt, tisztán termikus szimulációra alkalmas rendszerek az alábbiak.

- ANSYS: Multifizikai szimulátor, beépített termikus modellel. Képes a hővezetési, hőszárazási, hőáramlási és fázisváltozással járó jelenségek modellezésére FEM szimulációval [12].



2.2. ábra. A termikus szimulátor helye a logi-termikus szimulációban

- COMSOL Multiphysics: multifizikai szimulátor hőtranszfer modullal, hasonló az ANSYS-hoz. A hőmérsékleti szimulációkat FEM módszerrel hajtja végre[13].
- Mentor Graphics *FloTherm*: FEM alapú termikus szimulátor hővezetési, hőszugárzási, hőáramlási modellekkel [14].
- Gradient Design Automation *HeatWave*: integrált áramköri termikus szimulátor, mely képes teljes chippek és többretegű SiP (System in Package)⁴ rendszerek termikus szimulációjára [15]. A fogyasztási adatokat egy külső áramkörszimulációs alkalmazásból nyeri.

A logi-termikus szimulációhoz olyan termikus szimulátorra van szükség, melyet szorosan lehet integrálni a teljesítmény adatokat szolgáltató alkalmazásba. Az eljárás implementációjához olyan termikus szimulátort kellett választanom, melyet illeszteni lehet egy EDA rendszer logikai szimulátorához. A szoros integráció feltétele, hogy lehetőleg forráskód szinten a termikus szimulátor csatolható legyen a logikai szimulátorhoz. A logikai és termikus szimulátorok közötti alacsony szintű csatolás segítségével érhető el a legnagyobb teljesítmény a logi-termikus szimulációban. A forráskód szintű csatolást az Elektronikus Eszközök Tanszékén korábban kifejlesztett Therman[16] alkalmazás adaptációjával lehetett megoldani. Az adaptáció során a Therman szimulációs motort úgy kellett átalakítani, hogy az EDA környezethez és az általam kidolgozott logi-termikus eljáráshoz illeszkedjen. A termikus szimulációs motor adaptációja a *LayTherm* nevet kapta.

A *LayTherm* termikus szimulációs motor működését a 3. fejezet mutatja be részletesen. A

⁴ SiP: Teljes elektronikai rendszer egy tokba integrálva. Különbség a SoC (System on Chip) megoldáshoz képest, hogy a SiP-ben a különböző funkciók a tokon belül vannak integrálva, általában több hordozón. A SoC esetében a rendszer funkcióit egy chipen valósítják meg.

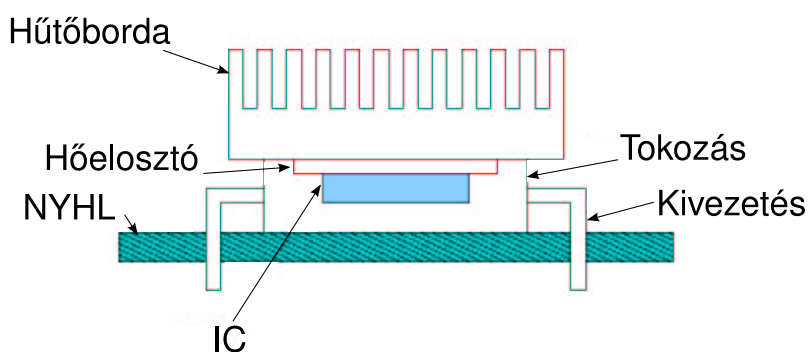
termikus modell generálását és ennek termikus szimulációját a LayTherm alkalmazás végzi a struktúra fizikai paramétereire és a számított fogyasztások alapján.

3. fejezet

Termikus szimuláció

Értekezésemben olyan termikus szimulációs eljárást mutatok be, melyet elsőként az Elektronikus Eszközök Tanszékén fejlesztettek ki és több nemzetközi konferencián és folyóiratban került bemutatásra. Ezeket a publikációkat a mai napig hivatkozzák nemzetközi cikkekben és konferenciákon. Az eljárást eredetileg Dr. Székely Vladimír, az MTA rendes tagja adaptálta tetszőleges rétegszámra akadémiai doktori értekezésében[17, 18].

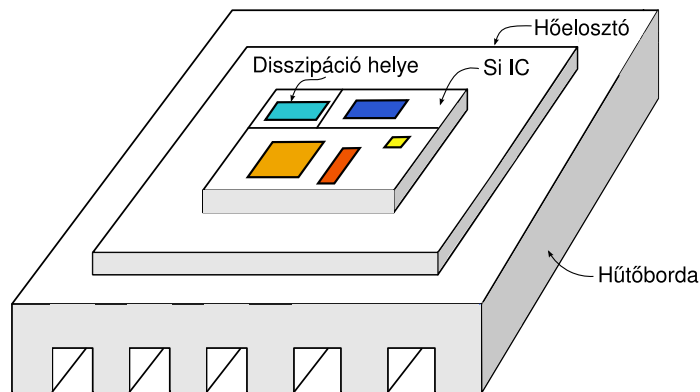
Az integrált mikroelektronika jellemző struktúráiban a termikus hatásokat az IC-k tokozásán belül, a szilícium lapkától a tokozáson és különböző hőelvezető struktúrákon keresztül egészen a külső környezetig vizsgáljuk. Egy jellemző struktúrát mutat a 3.1. ábra. A nyomtatott huzalozású lemezre (NYHL) kerül beültetésre a tokozott integrált áramkör. Az IC tokon belül található a szilícium lapka (die), melyen az áramköri elemek vannak kialakítva. A lapka hátoldalán egy hőelosztó réteg (termikus interfész anyag, TIM) gondoskodik a működés közben keletkezett hőmérséklet szétosztásáról és átvezetéséről a hűtőborda felé. A tok tetején található a hűtőborda, mely rendszerint valamilyen hővezető anyag közvetítésével érintkezik az áramköri tokkal.



3.1. ábra. Tipikus mikroelektronikai tokozási struktúra

Az integrált áramkörben a kialakított eszközök, tranzisztorok, analóg és digitális elemek a szilícium lapka tetején (front-side) helyezkednek el. A disszipáció ezen a rétegen keletkezik az áramköri elemek működése következtében. Ezekben a struktúrákban a keletkező Joule-hő, mely melegíti az eszközt, szinte kizárólag hővezetéssel kerül elvezetésre. A hőszállítás és konvekcióval (hőszállítással) eltávozó hőteljesítmény elhanyagolható mértékű a lapkán (die).

A 3.2. ábra egy sematikus képet mutat be az integrált áramkör felületéről és a disszipáció forrásainak helyéről az áramköri lapkán.



3.2. ábra. Az integrált áramkör felületi disszipáló elemei

Az általam a logi-termikus szimuláció során használt termikus szimulációs eljárás az Elektronikus Eszközök Tanszékén kifejlesztett Therman[16], SISSI[19] és Thermodel[20, 21] alkalmazásokban implementált módszereket használja fel.

3.1. A termikus modell előállítás és szimulációja

Egy mikroelektronikai struktúra termikus modelljének előállítását és ennek szimulációját több lépésben lehet végrehajtani.

1. Az integrált áramköri struktúrában –mint ami a 3.2. ábrán látható– különböző paramétereket kell meghatározni a termikus modell generálásához.
 - A struktúra laterális méretei (szélesség, magasság),
 - a struktúra rétegeinek agyagparaméterei (hőkapacitás, hővezetési együttható),
 - a struktúra rétegeinek vastagsága,
 - a struktúra szélein lévő peremfeltételek (adiabatikus, izotermikus, hővezetés),
 - a disszipáció helyei (az áramköri elemek mérete és pozíciója).

A termikus szimulátor a megadott adatok alapján megoldja a hővezetés időfüggő Laplace differenciálegyenletét (3.1).

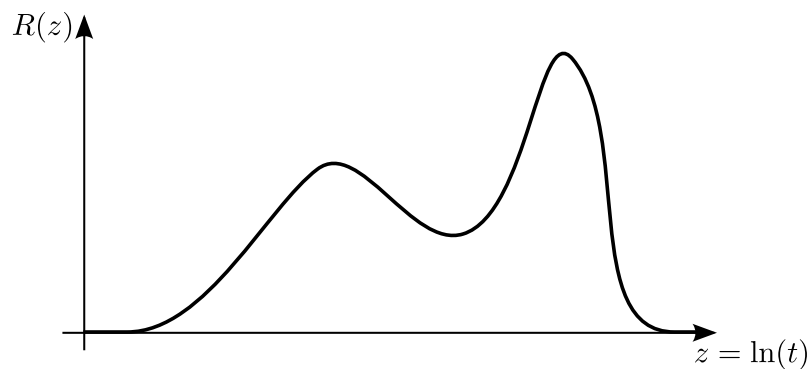
$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = \frac{c}{\lambda} \frac{\partial T}{\partial t}, \quad (3.1)$$

ahol T a hőmérséklet, x , y és z háromdimenziós koordináták, c az anyag térfogategységre vonatkoztatott hőkapacitása, λ a termikus vezetőképesség és t az idő. A megoldás algoritmusát és menetét [22, 12.2.2. szakasza] tárgyalja.

Az algoritmus segítségével meghatározható a rendszer egységugrás gerjesztésre adott ugrás-válasza, az $a(t)$ függvény. Az ugrás-válasz függvényt több disszipációs forrás esetében minden forrásra meg kell határozni. Az algoritmus figyelembe veszi a különböző disszipációs források egymásra hatását is, ezért N forrás esetében N^2 ugrás-válasz függvényt kapunk.

A mikroelektronikai rendszerek termikus időállandói széles spektrumot ölelnek át, akár több nagyságrendnyi különbség is lehet közöttük. Annak érdekében, hogy a számításokat széles időtartományban tudjuk végezni, célszerű az időt logaritmikus léptékben ábrázolni, így az $a(t)$ ugrás-válasz függvény helyett az $a(z)$ függvényt használjuk $z = \ln t$ helyettesítéssel.

2. A rendszer ugrás-válaszainak ($a(z)$) ismeretében dekonvolúció segítségével meghatározható az adott forráshoz tartozó *időállandó spektrum* $R(z)$. Ennek menetét a [20, 21] hivatkozások ismertetik. Az időállandó spektrum akár az időtartománybeli, akár a frekvenciatartománybeli ugrásválaszból előállítható. Az eljárást az említett irodalomban *hálózati identifikáció dekonvolúcióval* (Network Identification by Deconvolution, NID) néven említik. Az időállandó spektrumra mutat sematikus példát a 3.3. ábra.

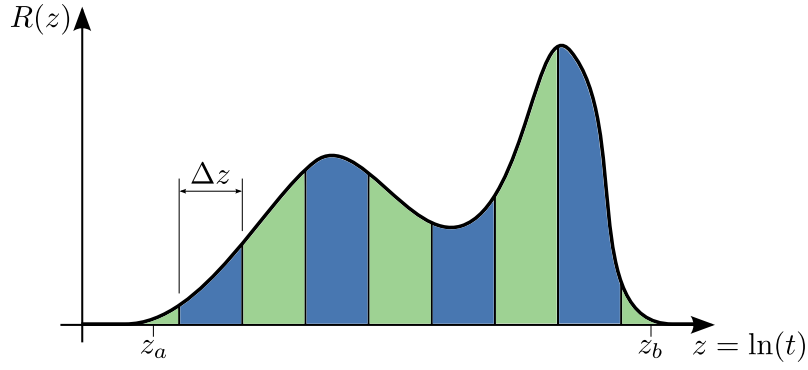


3.3. ábra. Időállandó spektrum

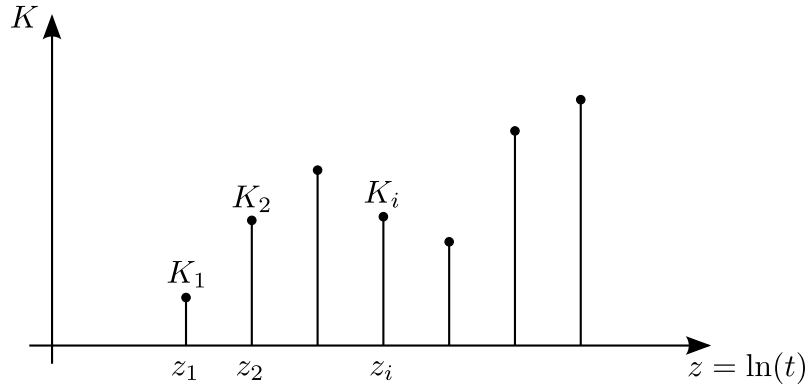
3. A rendszerhez tartozó időállandó spektrumok jól modellezik a struktúra idő- és frekvenciatartománybeli termikus viselkedését és lehetőséget biztosítanak kompakt RC modellek létrehozására. Az így létrehozott kompakt RC hálózat modellek a struktúrát elektromos RC hálózatként reprezentálják, aminek nagy előnye, hogy a hőmérsékleti tranzienseket analóg SPICE szimulációkkal meg lehet határozni.

A kompakt modell generálásához szükség van az időállandó spektrum diszkretizálására, hiszen az időállandó spektrum egy folytonos függvény. Annak érdekében, hogy a származtatott kompakt modellt analóg áramkörszimulációs rendszerben vizsgálni tudjuk, az időállandó spektrumot kvantálni kell, véges számú RC tagra kell redukálni az eredő kompakt modellt. A redukció elvégzéséhez a folytonos időállandó spektrumot N egyenlő részre osztjuk fel a 3.4. ábra szerint.

Az így keletkező szakaszok közepén határozzuk meg a diszkrét z_i időállandó értékeket a 3.5. ábra szerint.



3.4. ábra. Az időállandó spektrum felosztása



3.5. ábra. Kvantált időállandó spektrum

A z_i pontokhoz tartozó K_i függvényértékeket az adott szakaszhoz tartozó görbe alatti terület adja meg (3.2) alapján.

$$K_i = \int_{z_a + (i-1)\Delta z}^{z_a + i\Delta z} R(\zeta) d\zeta. \quad (3.2)$$

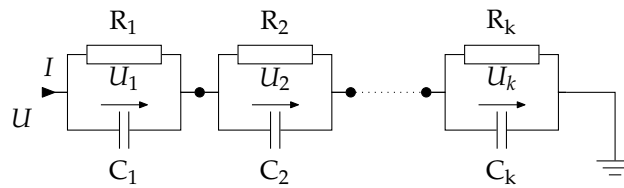
4. A kapott (z_i, K_i) párokból egyenesen adódnak a kompakt modell Foster RC létrájának elemértékei. A z_i értékek lineáris időtartományba való visszatranszformálása után a (τ_i, K_i) értékekből (3.3) és (3.4) segítségével számítható a kompakt modell ellenállásainak és kapacitásainak értéke:

$$R_i = K_i, \quad (3.3)$$

$$C_i = \frac{\tau_i}{K_i}. \quad (3.4)$$

Az előálló Foster RC kompakt modell esetében jellemző és megfelelő pontosságot adó közelítés a 12 RC tagból álló kompakt modell[J1],[C1]. Egy Foster RC kompakt modellre mutat példát a 3.6. ábra.

5. A kompakt RC modellek ismeretében hagyományos hálózatanalízis segítségével a bemenő teljesítmények függvényében számíthatóak az eredő hőmérsékletek.



3.6. ábra. Foster RC létra

Munkám során a most bemutatott termikus szimulációs eljárást használtam fel. Az Elektronikus Eszközök Tanszékén kifejlesztett Therman termikus szimulátort úgy alakítottam át, hogy az EDA környezetekhez illeszkedjen (LayTherm). A termikus szimulációs motort a CellTherm logi-termikus szimulátorba ágyaztam be, ezáltal biztosítottam, hogy bármely EDA eszközzel használható legyen.

A kompakt RC modellek számítását saját algoritmussal végzem, analóg SPICE szimulátor használata nélkül. A számítás algoritmusát a 3.2. szakaszban mutatom be.

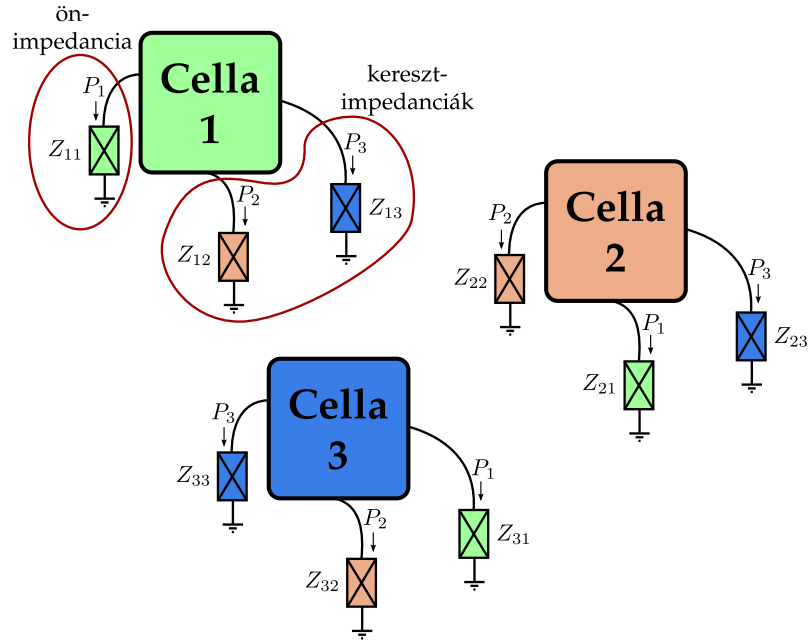
3.2. A termikus ekvivalens RC hálózat számítása

A CellTherm program a logikai szimulátor segítségével kinyert kapcsolási aktivitás és egy fogyasztási adatbázis adatai alapján meghatározza a cellák által felvett teljesítményt. Ez a teljesítmény melegíti az eszközt működés közben. A CellTherm alkalmazás a szimuláció kezdetén a felületi topológia, a cellák elhelyezkedése és a rétegstruktúra alapján felépíti az eszközt termikusan reprezentáló ekvivalens RC modellt. A termikus-elektromos analógia miatt a hőellenállás és hőkapacitás elektromos ellenállásként és kapacitásként modellezhető és a lineáris hálózatanalízis szabályai érvényesek rájuk. Az ilyen termikus RC hálózatok SPICE elektromos szimulációkkal analizálhatóak. A termikus RC hálózatok esetében az *áram* megfelelője a *teljesítmény*, a *feszültség* analóg mennyisége a *hőmérséklet*.

A CellTherm alkalmazás a 3.1. szakaszban ismertetett módon állítja elő a struktúra termikus időállandó-spektrumait. Ezekből a folytonos spektrumokból előállítja a véges elemszámú (véges időállandót tartalmazó) Foster RC létrák hálózatát. Az így létrejött hálózatot már áramkör-szimulációs (pl. SPICE) eszközökkel lehet analizálni és a bemenő teljesítmények függvényében az eredő hőmérsékleteket számítani. Mivel a CellTherm alkalmazásban az áramkör legkisebb építőeleme a sztenderd cella, a termikus Foster hálózatok bemeneti feszültségei (U) az egyes sztenderd cellák hőmérsékleteit jelölik. Minden cellához tartozik egy önimpedancia (Foster-létra) és $n - 1$ darab kereszt-impedancia, ami a többi cella hőmérsékletének hatását jelképezi az adott cellára, ahol n a cellák száma az áramkörben.

A 3.7. ábra szemlélteti, hogy minden cella eredő hőmérsékletét a cella saját melegedése, valamint a többi cella melegedése együttesen határozza meg. Egy cella eredő hőmérsékletét az önimpedanciáján „átfolyó” saját teljesítményárama és a többi cella hatását modellező kereszt-impedancián átfolyó teljesítményáramok adják. A kereszt-impedanciákon folyó teljesítményáram az a teljesítmény, amit a másik cella fogyaszt. Ezt mutatja (3.5) az 1-es cellára.

$$T_1 = P_1 Z_{11} + P_2 Z_{12} + P_3 Z_{13}. \quad (3.5)$$



3.7. ábra. Termikus ön- és kereszt-impedanciák

A (3.5) egyenlet általános esetben, ahol n darab cella van a rendszerben (3.6) szerint alakul.

$$T_1 = P_1 Z_{11} + P_2 Z_{12} + \dots + P_n Z_{1n} \quad (3.6)$$

$$T_2 = P_1 Z_{21} + P_2 Z_{22} + \dots + P_n Z_{2n} \quad (3.7)$$

$$\vdots$$

$$T_n = P_1 Z_{n1} + P_2 Z_{n2} + \dots + P_n Z_{nn}. \quad (3.8)$$

Mátrix alakban felírva kapjuk (3.9)-et.

$$\begin{pmatrix} T_1 \\ T_2 \\ \vdots \\ T_n \end{pmatrix} = \begin{pmatrix} Z_{11} & Z_{12} & \dots & Z_{1n} \\ Z_{21} & Z_{22} & \dots & Z_{2n} \\ & & \dots & \\ Z_{n1} & Z_{n2} & \dots & Z_{nn} \end{pmatrix} \cdot \begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \end{pmatrix}. \quad (3.9)$$

A (3.5)–(3.9) egyenletekben a Z impedanciáknak az egyes csomópontokhoz tartozó Foster-létrák felelnek meg, ahogy az a 3.6. ábrán látható[J1],[C1].

Az ilyen egyszerű hálózatban a feszültségek számítását jelentősen le lehet egyszerűsíteni[J2], nem szükséges egy lineáris hálózat-megoldó algoritmust írni az analízishez. A számítás a következőképpen történhet. Tegyük fel, hogy a Foster RC létra bemenő áramát ismerjük (ez termikus hálózat esetében a teljesítményáram lesz). Ki kell számítani az U csomópontához tartozó feszültséget. Ez a feszültség megfelel a (3.8) képletben egy $P_n Z_{nn}$ tagnak. Határozzuk meg a Foster RC létrában a pólusok feszültségét (U_1 – U_k). Ehhez felírhatjuk minden pólusra az ott folyó áram összefüggését (3.10)–(3.12).

$$I = \frac{U_1}{R_1} + C_1 \cdot \frac{dU_1}{dt} \quad (3.10)$$

$$I = \frac{U_2}{R_2} + C_2 \cdot \frac{dU_2}{dt} \quad (3.11)$$

$$\vdots$$

$$I = \frac{U_k}{R_k} + C_k \cdot \frac{dU_k}{dt}. \quad (3.12)$$

A kapacitás árama az $I_c = C \cdot \frac{dU_c}{dt}$ összefüggés alapján számítható, ahol $\frac{dU_c}{dt}$ differenciálhányadost a numerikus számítás miatt differenciahányados közelíti (3.13) alapján.

$$\frac{dU_k}{dt} \approx \frac{U_k(t_i) - U_k(t_{i-1})}{\Delta t}. \quad (3.13)$$

Itt $U_k(t_i)$ a feszültség *aktuális*, $U_k(t_{i-1})$ az *előző* értékét jelenti.

$$t_i = t_{i-1} + \Delta t. \quad (3.14)$$

Δt az időlépés, amennyi időközönként a szimulátor új feszültség értéket számít. Jelen esetben a feszültség a csomópont hőmérsékletét jelenti, Δt pedig azt az időtartamot, amelyre az időben változó teljesítményeket átlagoljuk. Mivel a digitális rendszer jellemző időállandói (kapcsolási frekvencia, órajel periódus) jóval kisebbek, mint a termikus rendszer időállandói, ezért a hőmérséklet számításokat jóval nagyobb időközönként lehet végezni. Például, ha egy digitális rendszer órajel frekvenciája 100 MHz, az ehhez tartozó periódusidő 10 ns. A termikus hálózat időállandói az ezredmásodperctől egészen a több perces tartományba is eshetnek. A kétfajta időállandó közötti több nagyságrend különbség miatt termikusan elégséges csak például 10 ms-onként átlagolni a fogyasztást. Az ez idő alatt bekövetkezett fogyasztások átlagértéke fogja adni a Foster RC létrák bemenő teljesítményáramát. Δt ebben a példában így 10 ms lesz.

(3.12)-t átrendezve kapjuk (3.15)-öt

$$U_k(t_i) = \frac{I + \frac{C_k}{\Delta t} U_k(t_{i-1})}{\frac{1}{R_k} + \frac{C_k}{\Delta t}}. \quad (3.15)$$

(3.15)-öt kiszámítva a Foster RC létra minden pólusára és összegezve, megkapjuk az U csomópont feszültségét (hőmérsékletét).

$$U = U_1(t_i) + U_2(t_i) + \dots + U_k(t_i) \quad (3.16)$$

$$U = \sum_{k=1}^{\text{pólusok száma}} U_k(t_i). \quad (3.17)$$

Az U csomóponton kiszámított feszültség (3.17) egy tagnak felel meg az egy cellához tartozó hőmérséklet egyenletben (3.8). Egy cella eredő hőmérséklete a saját melegedésből és a környező cellák melegedésének hatásából adódik. Minden tag kiszámítása és összeadása (3.8)-ban az előzőleg bemutatott módszerrel a cella aktuális hőmérsékletét adja az adott szimulációs időlépésben. Ezt a számítást minden cellára el kell végezni, hogy megkapjuk minden cella eredő hőmérsékletét.

A bemutatott számítási algoritmus implementációját úgy készítettem el, hogy modern, több processzort, vagy processzor magot tartalmazó munkaállomásokon a hálózat Foster RC létráinak számítása párhuzamosan menjen végbe. Mivel az egyes Foster létrák be- és kimeneti adatai –az összegzési lépésig– függetlenek egymástól, ezért nagyfokú párhuzamosítás érhető el és így a számítási idő tovább csökkenthető. A jelenlegi implementációban a szabványos *OpenMP*[23] párhuzamos programozási interfészt használtam. Az algoritmus más párhuzamos programozási modellel is megvalósítható, mint például a *CUDA*[24] vagy az *OpenCL*[25].

3.3. HotSpot

Az Egyesült Államokbeli Virginia Egyetem Műszaki és Alkalmazott Tudományok karának Informatika Tanszéken a 3.1. szakaszban bemutatott elvhez hasonló módszerrel dolgoztak ki termikus modellt mikroelektronikai rendszerek vizsgálatához. Az eljárás segítségével az integrált áramkörök tervezésekor a hőmérsékleti és fogyasztási tényezőket is figyelembe lehet venni. Ezt a módszertant a szakirodalomban *temperature-aware design*-nak hívják. A Virginia Egyetemen kidolgozott termikus modell a *HotSpot* nevet viseli [26, 27]. A *HotSpot* modell egyben egy szimulációs környezet is, melynek segítségével az integrált áramkörök fogyasztása alapján meghatározhatók az áramkör termikus viszonyai is. A *HotSpot* szimulátor a termikus és az elektromos hálózati számítások analógiáját felhasználva az anyagi paraméterekből és a szimulált struktúra fizikai méreteiből olyan modellt állít elő, ahol a struktúrát hőellenállásokkal és hőkapacitásokkal jellemzi. A fűtőteljesítményt áramforrással jelképezi a termikus hálózat megoldásánál. Az így előállított modell független a peremfeltételektől és a kezdeti hőmérsékletektől.

A *HotSpot* szimulátor által előállított modellben a struktúra különböző rétegeinek száma kötött. A struktúra rétegeinek (hűtőborda, hőelosztó, szilícium chip) anyagi paramétereiből és méreteiből a *HotSpot* meghatározza a kompakt RC modell elemértékeit. Alapvetően a hőellenállások értékét a vizsgált réteg vastagságából, felületéből és hővezetési együtthatójából számítja ki (3.18) alapján.

$$R = \frac{d}{\lambda \cdot A}, \quad (3.18)$$

ahol d a réteg vastagsága, A a felület és λ az anyag hővezetési együtthatója. A kapacitás értékek meghatározása szintén a réteg vastagságából, keresztmetszetéből és az egységnyi térfogatra eső hőkapacitásból adódik, ami anyagi paraméter (3.19).

$$C = c \cdot d \cdot A, \quad (3.19)$$

ahol c az egységnyi térfogatra eső hőkapacitás, d a réteg vastagsága és A a felület.

A modellben a fizikai struktúra rétegszámán nem lehet változtatni, a hűtőborda, a termikus határfelületi anyag (Thermal Interface Material, TIM), és az integrált áramkör chipje mindig jelen van. A szimuláció konfigurációja során a szilícium chipen elhelyezkedő architektúráis egységeket, blokkokat lehet definiálni. Ez felel meg a felületi topológiának (layout-nak). A HotSpot szimulátort úgy fejlesztették ki, hogy a szilícium chipen különböző architektúráis egységek hőmérsékleti viszonyait lehessen megfigyelni. Az egyes architektúráis blokkokat a vizsgált áramkör határozza meg. [26]-ban a HotSpot készítői egy Compaq Alpha 21364 processzor szimulációjáról számolnak be, melynek különböző architektúráis részei (például lebegőpontos egység, egész regisztertömb, lebegőpontos összeadó, szorzó) alkották a szimuláció blokkjait a szilícium chipen.

A HotSpot szimulátor kezdeti verziói csak egyutas hőterjedéssel számoltak a hűtőborda felé. A szimulátor a fejlesztéseknek köszönhetően azóta képes kétutas hőterjedéssel is számolni, aminek segítségével figyelembe lehet venni a kivezetéseken a nyomtatott huzalozású lemez felé és a vezetékezési rétegeken (interconnect layer) eltávozó hőt. Számításaik alapján a teljes hőtermelés 30 %-a a másodlagos hőáramlási úton távozik el.

A HotSpot továbbfejlesztése révén olyan modell létrehozására is mód van, ahol az egyes blokkok nem az architektúráis részegységek, hanem a szilícium felületét egy rács bontja fel több egyforma területegységre. Az RC modellt ezután ezekre a rács-területekre hozza létre a HotSpot. A módszer segítségével finomítani lehet a felületi hőeloszlás térkép felbontását, mivel így egy hőmérséklet érték nem egy teljes architektúráis blokkot jellemez, hanem adott egységek kisebb méretű blokkokra oszthatóak.

Az egyes blokkok fogyasztási adatait a *Wattch* architektúráis szintű teljesítmény analízis eszközzel határozzák meg [28]. Ezek a teljesítmények szolgálnak az RC hálózatot gerjesztő áramforrások bemeneti adatául. A *Wattch* teljesítmény analizátor minden architektúráis egység fogyasztását nyomon követi és minden időlépésben frissíti ezeket az adatokat az RC hálózatmegoldó algoritmus számára.

A HotSpot modellt és szimulátort a szerzők többféleképpen verifikálták. A számítások helyességét végeselemes (FEM) termikus szimulátorral verifikálták, illetve mérésekkel ellenőrizték egy termikus teszt integrált áramkörön. A verifikáció során 5 % alatti hibát kaptak a HotSpot szimulációkkal.

A HotSpot termikus modell és szimulátor a Virginia Egyetem Informatikai Tanszékének honlapjáról (<http://lava.cs.virginia.edu/HotSpot>) ingyenesen letölthető. A készítőik kidolgoztak egy termikus hatásokat is figyelembe vevő, a HotSpot modellen alapuló elhelyező és vezetékező alkalmazást is HotFloorplan néven, mely része a HotSpot csomagnak. Az alkalmazás segítségével fizikai elhelyezést és vezetékeezést az integrált áramkörön a kialakuló hőmérsékletektől függően lehet szabályozni.

A HotSpot szimulátor az RC hálózat választását negyedrendű Runge-Kutta (RK4) módszerrel határozza meg. Az általam a 3.2. szakaszban bemutatott módszer a lineáris RC hálózat megoldásához gyorsabb és egyszerűbb algoritmust használ. Korlátja a HotSpot modellnek, hogy a szimulált struktúra rétegei adottak, és nem lehet tetszőleges mennyiségű és anyagi paraméterű réteget modellezni. Az értekezésemben bemutatott eljárás segítségével tetszőleges rétegszámú és anyagú lehet a struktúra és lehetőség van akár tokozások mérésekből származó kompakt modelljeinek csatolására is.

3.4. Termikus szimuláció bemutatása egy példán

Egy integrált áramkör tisztán termikus szimulációhoz a 3.1. táblázatban összefoglalt bemeneti adatokra és paraméterekre van szükség.

3.1. táblázat. Bemenő adatok termikus szimulációhoz

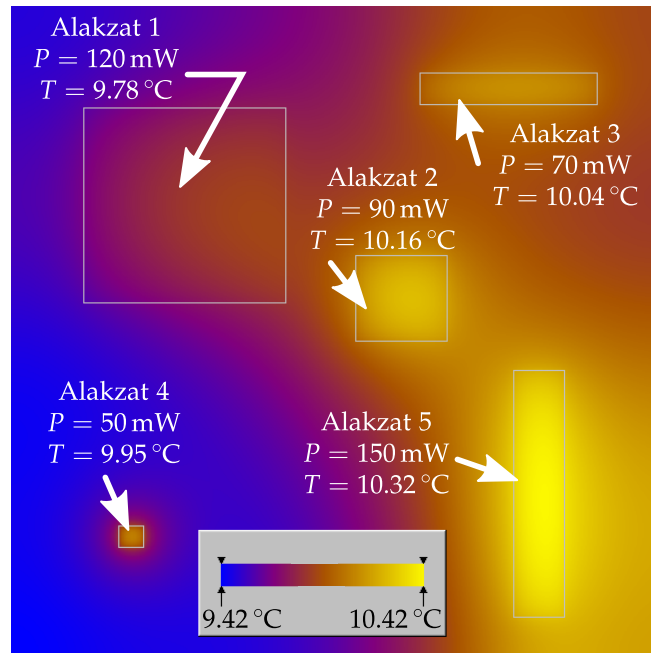
Bemeneti paraméterek	
Struktúra paraméterek	IC laterális méretei rétegvastagságok
Rétegek anyagparaméterei	hőellenállás/hővezetés hőkapacitás
Disszipáló alakzatok	helye az IC-n mérete az alakzaton disszipált teljesítmény (P)
Peremfeltételek a struktúra szélein	adiabatikus izotermikus hővezetés, konvekció

A 3.8. ábra egy tisztán termikus szimuláció állandósult állapotát (DC szimuláció) mutatja be. Az ábrán öt alakzat figyelhető meg különböző elrendezésben és méretben. Az ábrán látható minden alakzat disszipált teljesítménye és az alakzatok helyén kialakult hőmérséklet is. A 3.8. ábra a stacionárius állapot hőmérsékleti eloszlását mutatja. A szimulációt 1024×1024 -es felbontással, a hővezetési egyenlet Fourier-módszerrel történő megoldásával végezte a termikus szimulátor. Az alakzatok eredő hőmérséklete az alakzaton belüli hőmérséklet-eloszlás számtani átlaga.

Logi-termikus szimulációk esetében a disszipáló alakzatok (sztenderd cellás áramkör esetében a cellák) termikus kompakt modelljeit határozza meg a szimulátor. A kompakt RC modellek hálózatában minden cella egy termikus csomópontot képvisel, ezért egy cella hőmérsékletét a hozzá tartozó kompakt RC modell egy csomópontja határozza meg. Így a logi-termikus szimulációban a cellákon belül nem értelmezhető hőmérsékleti gradiens, a cellák belsejében a hőmérsékleti eloszlást homogénnek tekintjük melynek értéke a hozzá tartozó termikus csomóponti hőmérséklet.

A termikusan szimulált struktúra fizikai paramétereit a 3.2. táblázat mutatja be. A struktúra három rétegből áll. Egy Kovar réteg a kivezető keretet (leadframe) jelöli. Erre egy ragasztó réteggel illeszkedik a felső szilícium, ahol a disszipáció történik.

A bemutatott példából látszik, hogy egy termikus szimuláció során a struktúra fizikai paraméterein kívül az egyes alakzatok disszipált teljesítménye határozza meg a hőmérsékleti eloszlást. Ezért minden termikus szimuláció bemenő paramétere a disszipált teljesítmény amely Joule-hővé alakul és melegíti az eszközt. A logi-termikus szimulációhoz tehát elengedhetetlenül fontos ismerni az áramköri elemek (cellák) disszipációját. A 4. fejezetben bemutatom azokat a teljesítmény és energia modelleket, melyekkel egy sztenderd cellás integrált áramkörben meghatározható a cellák disszipációja. A kapcsolási aktivitás és a fogyasztás ismeretében a logi-termikus szimuláció lehetővé válik.



3.8. ábra. Tisztán termikus szimuláció

3.2. táblázat. A struktúra fizikai paramétereit

	Jelölés	Érték
Felület	A	$10 \text{ mm} \times 10 \text{ mm}$
Si réteg vastagsága	d_{Si}	$300 \text{ }\mu\text{m}$
Si termikus vezetőképessége	λ_{Si}	$156,3 \frac{\text{W}}{\text{m K}}$
Si hőkapacitása	c_{Si}	$1,596 \cdot 10^6 \frac{\text{W s}}{\text{m}^3 \text{ K}}$
Ragasztó réteg vastagsága	d_{rag}	$50 \text{ }\mu\text{m}$
Ragasztó termikus vezetőképessége	λ_{rag}	$4 \frac{\text{W}}{\text{m K}}$
Ragasztó hőkapacitása	c_{rag}	$1,5 \cdot 10^6 \frac{\text{W s}}{\text{m}^3 \text{ K}}$
Kovar réteg vastagsága	d_{Kovar}	$500 \text{ }\mu\text{m}$
Kovar termikus vezetőképessége	λ_{Kovar}	$16,3 \frac{\text{W}}{\text{m K}}$
Kovar hőkapacitása	c_{Kovar}	$2 \cdot 10^6 \frac{\text{W s}}{\text{m}^3 \text{ K}}$
Alsó hűtés	HTC	$500 \frac{\text{W}}{\text{m}^2 \text{ K}}$
Környezeti hőmérséklet	T_{amb}	0 °C

Az általam a 3. fejezetben bemutatott és a CellTherm logi-termikus szimulátorban felhasznált termikus szimulátor (LayTherm) alapjául szolgáló Thermodel/Thermanal alkalmazás az ANSYS termikus szimulátorral lett verifikálva[21]. Az ANSYS végesesemes (FEM) szimulációs környezetben egy SP10 nyomásmérő szenzor modelljét szimulálva az így kinyert termikus ugrás-válasz szolgált az időállandó spektrum meghatározásának alapjául. A FEM szimuláció és az általam felhasznált eljárás eredményei azonosak[21, 55. o.].

4. fejezet

Teljesítmény és energia modellek EDA környezetben

A integrált áramkörök szimulációs vizsgálatához minden esetben szükség van olyan modellekre, melyek a fizikai működést a lehető legjobban írják le. A modellek segítségével és a szimulációkkal az áramkörök valóságos működését tudjuk közelíteni. A tényleges fizikai működés és a szimulációs eredmények között annál kisebb lesz az eltérés, minél pontosabban írja le a modell a fizikai hatásokat.

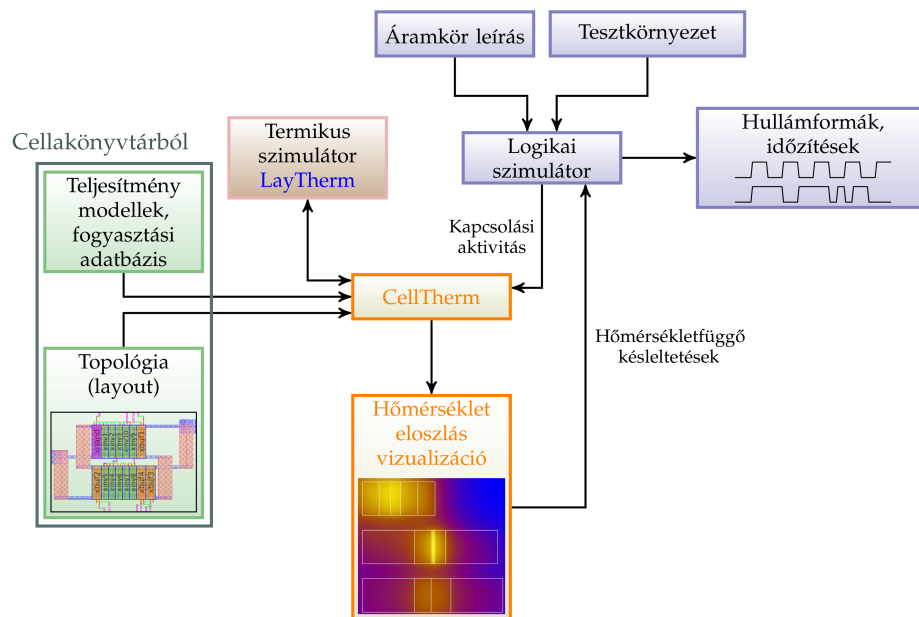
A sztenderd cellás tervezés esetében a szimulációkhoz, és az áramkörök tervezőrendszerekben történő megalkotásához különböző modellekre van szükség. A sztenderd cellák működését leíró modelleknek számos fizikai információt kell reprezentálniuk, lehetőleg minél pontosabban. Ilyen fizikai tulajdonságok az időzítések, a fogyasztási adatok, a különböző zajparaméterek, stb. Ezek a paraméterek általában csatoltan függenek egymástól, így modellezésük bonyolult. A gyártási bizonytalanságok, amik a mai nanoméretű integrált áramkörök esetén különösen jelentősek, a változó hőmérséklet és az ingadozó tápfeszültség többek között olyan fontos paraméterek, melyektől a cellák paraméterei erősen függenek. Egy alkalmas modell tehát ezeket a változásokat, bizonytalanságokat is magában foglalja.

Logi-termikus szimuláció során a hőmérsékleti hatások meghatározásához szükség van az áramköri elemek statikus és dinamikus disszipációinak ismeretére. Ezeket az adatokat szolgáltatják a gyártó által karakterizált teljesítmény és energia modellek és adatbázisok.

A teljesítmény és energia modellek helyét a logi-termikus szimulációban a 4.1. ábra mutatja. A modellek és a teljesítmény adatokat tartalmazó adatbázisok jellemzően a cellakönyvtár részét képezik.

Ahogy az integrált áramkörök a nanométeres mérettartományba léptek, egyre inkább előtérbe kerültek az eddig másodlagosként kezelt fizikai hatások. A méretek csökkenésével olyan modellekre van szükség, ahol a másodlagos, akár kvantumfizikai hatások is reprezentálva vannak.

Az ezredforduló környéke óta használnak az integrált áramkörtervezők olyan nemlineáris modelleket az időzítések, fogyasztás és zajparaméterek leírására, melyek segítségével a szubmikronos hatásokat is figyelembe lehet venni. A nemlineáris modellek előtt a cellák késleltetési és fogyasztási adatait csak egy előre rögzített adattal írták le, ami a másodlagos effektusok előtérbe kerülésével egyre nagyobb szimulációs hibákat okozott.



4.1. ábra. A teljesítmény modellek helye a logi-termikus szimulációban

4.1. Disszipáció CMOS áramkörökben

Modern sztenderd cellás áramköröket az esetek jelentős többségében CMOS kapcsolástechnikával valósulnak meg. Ez azt jelenti, hogy a sztenderd cellákat felépítő kapcsolásokban p és n csatornás MOS tranzisztorok valósítják meg a kívánt funkciót. Az ilyen áramkörökben a disszipáció (és az eszköz melegedése) szempontjából két fontos mennyiséget lehet megkülönböztetni: a szivárgási és a dinamikus teljesítményt.

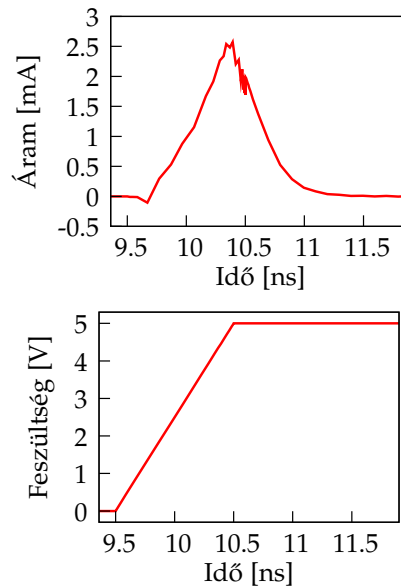
Egy áramkör kapcsolási aktivitás hiányában is fogyaszt valamennyi teljesítményt, ezt nevezzük *szivárgási teljesítménynek* (leakage power). Ez abból adódik, hogy a lezárt állapotban lévő tranzisztorokon is folyik valamennyi áram (küszöb alatti áram, nA-es nagyságrend). Ezek az áramok általában nagyságrendekkel kisebbek ugyan, mint a kapcsolási aktivitás töltő/kisütő áramai, de többmillió tranzisztorszámnál már jelentős fogyasztást okoznak.

Kapcsolási aktivitáskor, amikor a cella a kimenetén lévő terheléseket tölti vagy kisüti, megnő a tápáram felvétel. A tápáram felvétel kapcsoláskor történő megnövekedése a *dinamikus fogyasztásnak* felel meg. Ide tartozik az a fogyasztás is, mely CMOS technológiánál a p és n csatornás tranzisztorok egymásba nyitása miatt lép fel. Ekkor egy bizonyos időre a táp és a föld között közvetlen áramút alakul ki, mely szintén növeli a fogyasztást. A teljes teljesítménynyelvételt a statikus (szivárgási) és a dinamikus fogyasztás összege adja.

A 4.2. ábra egy inverter cella tápáram görbéjét mutatja a bemenet logikai 0–1 átmeneténél (TSMC 0,35 μm -es CMOS technológia).

4.1.1. Szivárgási áram és teljesítmény

A nanométeres csíkszélességű integrált áramkörökben rendkívül fontossá vált a szivárgási áramok modellezése. A méretek csökkentése miatt a tranzisztorok működési tartománya már szorosan a küszöbfeeszültség körül található. Ez azt jelenti, hogy már nem beszélhetünk a



4.2. ábra. Áramfüggvény egy inverter cella táp lábán

tranzisztor teljesen lezárt és teljesen nyitott állapotáról. A küszöbfeszültség környékén történő működés egyre nagyobb szivárgási fogyasztást eredményez a tranzisztor „lezárt” állapotában is. Ezeket a szivárgási áramokat és teljesítményeket modellezni kell.

A szivárgási áram és teljesítmény közötti összefüggést (4.1) írja le:

$$P_{\text{szivárgási}} = I_{\text{szivárgási}} \cdot U_{\text{táp}}. \quad (4.1)$$

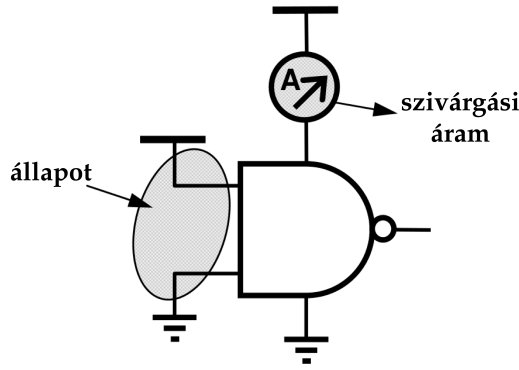
A szivárgási teljesítményt meg lehet határozni a cella belső állapotából. Kombinációs celláknál a belső állapot csak a bemeneti kombinációtól függ. Sorrendi celláknál az aktuális állapot nemcsak a bemeneti kombinációtól, hanem az előző állapottól is függ. Általában a belső állapot kikövetkeztethető a kimeneten lévő kombinációból. Megfelelő kezdeti állapotból egy cella bármely tetszőleges állapotába átvihető és itt egy egyszerű DC szimulációval a szivárgási áram meghatározható (4.3. ábra). A szivárgási teljesítmény könnyen számítható (4.1) segítségével és minden lehetséges állapotra (bemeneti és kimeneti kombinációra) a szivárgási fogyasztás eltárolható egy adatbázisban.

4.1.2. Dinamikus és belső energia

A dinamikus fogyasztás modellezése azért fontos, mert az áramkör működése és a kapcsolási aktivitás következtében ezzel a mennyiséggel lehet figyelembe venni a tisztán a kapcsolásokból származó disszipációt. Egymástól külön kell kezelni a statikus és a dinamikus fogyasztást, mivel így pontosabb teljesítményanalízis hajtható végre a teljes rendszeren. A statikus és dinamikus energiák ismeretében meghatározható, hogy mekkora teljesítményt fogyaszt az áramkör üresjáratban és működés közben.

A dinamikus energia két összetevőre bontható:

- a kapcsolási aktivitás által a terhelő kapacitást töltő és kisütő energia,
- a megmaradt belső energia.



4.3. ábra. A szivárgási áram meghatározása

A belső energia definíció szerint a dinamikus energia és a kapcsolási energia különbsége:

$$E_{\text{belső}} = E_{\text{din}} - E_{\text{kapcs}} \quad (4.2)$$

$$E_{\text{kapcs}} = C_{\text{terhelés}} \cdot U_{\text{táp}}^2 \quad (4.3)$$

A kapcsolási energia tulajdonképpen a külső kapcsolási energia. A cella által látott kimeneti kapacitás a terhelő kapacitások összege valamint a kimeneti kivezetés kapacitása.

$$\begin{aligned} E_{\text{kapcs, teljes}} &= (C_{\text{kivezetés}} + C_{\text{terhelés}}) \cdot U_{\text{táp}}^2 \\ &= E_{\text{kapcs, belső}} + E_{\text{kapcs}} \end{aligned} \quad (4.4)$$

Egyfokozatú celláknál a dinamikus energia a rövidzárási energiából és a kapcsolási energiából áll. Ebben az esetben a belső energia a rövidzárási energia és a belső kapcsolási energia összege. Többfokozatú celláknál a belső energiába beszámít a rövidzárási energia és a köztes fokozatok teljes kapcsolási energiája is.

Egy logikai átmenethez tartozó dinamikus energiát az átvitt teljes töltésmennyiségből és a potenciálkülönbségből lehet meghatározni:

$$E_{\text{din}} = Q_{\text{din}} \cdot U_{\text{táp}} \quad (4.5)$$

A töltés a dinamikus áram idő szerinti integráljával egyenlő:

$$Q_{\text{din}} = \int_0^{+\infty} I_{\text{din}}(t) dt \quad (4.6)$$

4.1.3. Statikus és dinamikus fogyasztás analízis

A cellák fogyasztásának meghatározása egy áramkörben SPICE szimulációkkal történhet, melynek során minden táp- és föld lábra meghatározzuk az ottani áramfüggvényt. A pillanatnyi fogyasztást az áramfüggvény és a tápfeszültség szorzata adja. Ezzel könnyen meghatározható az átlagos és a csúcsteljesítmény az áramkör különböző részein. Bonyolult és nagy méretű digitális integrált áramkörökben azonban a szimulációs idők nagyon hosszúvá nyúlhatnak, így ezeknél a SPICE szimulációk végrehajtása nem célszerű.

A fogyasztás analízis felgyorsításához a különböző cellák állandósult állapotbeli és kapcsolási aktivitástól függő teljesítményfelvételét karakterizálni lehet. Ezeket a karakterizációs adatokat a teljesítmény modellben lehet eltárolni. A kapu szintű kapcsolás logikai szimulációja szolgáltatja a kapcsolási aktivitást a szimuláció közben. A teljesítményfelvétel jelentősen függ a kapcsolási aktivitástól. Teljesítmény analízist két különböző módon lehet végrehajtani:

1. esemény alapú (dinamikus analízis),
2. átlagos kapcsolási aktivitás alapú (statikus analízis).

Az esemény alapú analízis egy VCD (Value Change Dump) fájlt vagy hasonló dinamikus kapcsolási aktivitás információt használ az áramkör celláiban bekövetkezett események azonosítására. Minden esemény hatására az eseményhez tartozó energia kiolvasásra kerül az adatbázisból és egy fogyasztási görbe készül. Az ilyen típusú analízisnél a pontosság a legfontosabb kritérium.

Az átlagos kapcsolási aktivitás alapú analízis egy SAIF (Switching Activity Interchange Format) fájlt vagy egyéb statikus kapcsolási aktivitást tartalmazó információt használ a cellák átlagos teljesítményének számítására. A könyvtár minden táblázatában az összes lehetséges esemény rögzítve van. Az egyes események valószínűsége alapján kiszámítható az adott cella átlagos teljesítmény fogyasztása. A SAIF-alapú analízist gyakran használják teljesítmény optimalizálásra is. A statikus teljesítmény analízisnél nagy hangsúlyt fektetnek az analízis gyorsaságára.

4.2. A Liberty fájlformátum

A Liberty[™] fájlformátumot a Synopsys cég dolgozta ki a különböző teljesítmény-, időzítés- és zajmodellek paramétereinek tárolására. A Liberty formátum egy szöveges adatbázis állomány sztenderd cellás tervezőrendszerek részére, melyből az EDA eszközök a cellák karakterisztikáit ki tudják olvasni. Az elhelyező és huzalozó alkalmazások például ezt az adatbázist használják fel a sztenderd cellás áramkör topológiájának (layout) elkészítésekor. A Liberty formátum ezen kívül a cellakönyvtárak pontos validációjára, karakterizációjára és a szimulációs eszközökre is tartalmaz ajánlást. A Liberty fájlformátumra egy példát a B. függelékben a B.1. kódrészlet mutat be.

4.3. Nem lineáris teljesítmény és időzítési modellek (NLPM, NLDM)

A Synopsys cég több nemlineáris modellformátumot is kifejlesztett annak érdekében, hogy a másodlagos fizikai hatásokat figyelembe lehessen venni. Az iparban jelenleg legelterjedtebben használt modellek a nemlineáris késleltetési és teljesítmény modell, NLDM (Non-linear Delay Model), NLPM (Non-linear Power Model). Ezek a nemlineáris modellek a cellák időzítéseit, fogyasztási adatait és zajparamétereit táblázatokként tárolják el. A táblázatok értékei például a bemeneti jel felfutó élének meredekségével és a kimeneti terhelő kapacitással vannak paraméterezve. Ebből következik, hogy egy ilyen táblázat három dimenziós: minden bemeneti

meredekség és terhelő kapacitás értékhez tartozik egy késleltetés, vagy energia érték. Az ilyen modelleket a mai napig előszeretettel használják az integrált áramkörgyártó cégek és a fejlesztői eszközkészletek (PDK, Process Design Kit) részeként is ilyen formátumú adatbázisokat biztosítanak a tervezők részére.

Az NLDM és NLPM modellek pontossága attól függ, hogy milyen pontosan karakterizálták az adott cellakönyvtárat. A modellek paramétereit általában SPICE analóg szimulációkkal határozzák meg. A karakterizáció során a gyárnak meg kell találnia az optimumot a karakterisztika pontjainak száma és az adatbázis mérete között. A modell annál pontosabb lesz, minél több munkapontban hajtják végre a SPICE szimulációkat, viszont ettől az adatbázis mérete is megnő. A nagyobb adatbázisban az IC tervező alkalmazások lassabban tudnak keresni, ezért ez a szimulációk és egyéb műveletek sebességének csökkenésével jár. A karakterizáció során a mintavételi pontok helyét igyekeznek úgy beállítani, hogy a mintavételezett görbe a legkisebb hibával visszaállítható legyen.

Az NLDM, NLPM nemlineáris modellek a mintavételezés kvantáltsága, és az adatbázis méretének elfogadható értéken tartása miatt a SPICE szimulációk eredményeitől némileg eltérnek¹.

A dinamikus áramgörbék függenek a cella kezdeti állapotától és a bemeneten lévő kombinációtól ami az eseményt okozta (útvonalfüggés). Továbbá az áramgörbe függ az átmenet időtartamától (a bemenő jel meredekségétől) és a kimeneteken lévő elosztott RC hálózattal reprezentált terheléstől. A kezdeti állandósult állapot és a végső állandósult állapot közötti teljes állapotváltozást tekintve az állapotváltozás energiáját elsődlegesen az RC terhelés kapacitív összetevője határozza meg. Tehát az elosztott RC terhelést egy lineáris kapacitással helyettesítve –melynek kapacitása megegyezik az RC hálózat eredő kapacitásával– az adott eseményhez tartozó dinamikus energia értéke megközelíti az eredeti áramkör energiafogyasztását.

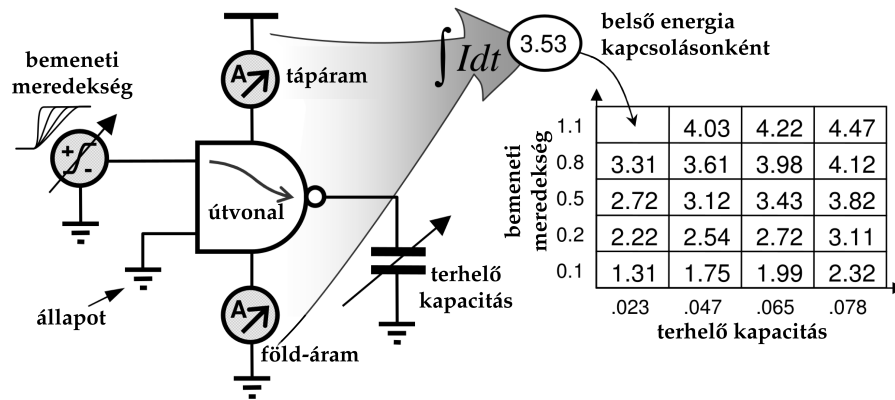
Bármely áramkörhöz a bemeneti jelek meredeksége meghatározható (statikus) időzírtési analízisek segítségével. A kimeneti terheléseket becslő ún. *wireload* modell és a bemeneti kapacitások értékei szintén rendelkezésre állnak a Liberty adatbázisban. A kapcsolási energia a kimeneti kapacitásból számítható (4.3)-mal. Így a belső energiafogyasztás értéke eltárolható a bemeneti jel meredeksége és a kimeneti terhelés függvényében. Ezt a táblázatos modellezési megközelítést mutatja a 4.4. ábra.

Minden állapothoz és bemeneti lábhoz szimulálható egy bemeneti esemény néhány bemeneti jelmeredekség és kimeneti kapacitás kombinációval. A dinamikus energia megkapható (4.5) és (4.6) segítségével. Végül a belső energia a dinamikus energia és a kapcsolási energia különbségeként adódik (4.2) alapján. Az így kapott belső energia értékeket a Liberty formátum táblázatokban tárolja a bemeneti meredekség és a kimeneti terhelőkapacitás értékekkel indexelve.

A nemlineáris teljesítmény modell (NLPM) táblázatokban az állapotfüggő szivárgási teljesítmények illetve az állapot- és útvonalfüggő belső energia táblázatok vannak eltárolva. Ennek a modellnek a célja, hogy a karakterizálás modellezési fázisában utófeldolgozást lehessen végrehajtani és az így kapott teljesítmény adatokat egy tömör formában a feldolgozó alkalmazások számára elérhetővé lehessen tenni.

Ennek a megközelítésnek az előnyei:

¹ A SPICE szimulációt tekinthetjük etalonnak, a fizikai működés legjobb modelljének, mivel a modern tranzistor modellek már a legtöbb nanométeres fizikai effektust is tartalmazzák.



4.4. ábra. Táblázat alapú modellek skálár értékekkel

- tömör cellakönyvtár,
- gyors szivárgási- és dinamikus teljesítmény-analízis.

A főbb hátrányok:

- speciális alkalmazásokra szabott,
- korlátozott pontosságú.

További hátránya az NLPM modellnek a dinamikus feszültségesés (IR-drop) analízis során jelentkezik. Ehhez az analízishez a táp-lábakon mért áramfüggvények alakjának pontos ismerete szükséges. Ezek az áramgörbék a karakterizáció során rendelkezésre állnak, de a táblázatban csak egy származtatott áramintegrál mennyiség van eltárolva. Ebből a táblázatból lehetetlen az eredeti áramfüggvény alakjának olyan pontosságú visszaállítása, mellyel tápsín analízist lehet végezni. Egy másik lényeges információ, ami hiányzik az NLPM modellből a kapcsolást nem végző cellák ekvivalens parazita hatása. Hagyományosan ezeket a cellákat vagy teljesen figyelmen kívül hagyják, vagy az általuk képviselt terhelő kapacitással helyettesítik egy tápsín analízis során. Bármelyik megoldás pontatlan dinamikus tápsín analízist eredményez.

4.4. Skálázható polinom modellek (SPPM, SPDM)

A nemlineáris modellek hibáinak csökkentésére az NLDM, NLPM modellek újabb generációja jelent meg. A skálázható polinom késleltetési és teljesítmény modellek (SPDM, Scalable Polynomial Delay Model), (SPPM, Scalable Polynomial Power Model) a mintavételi pontokban lineáris polinomfüggvénnyel közelítik a mintavételezett görbét. A modell előnye a hagyományos nemlineáris modellel szemben, hogy két mintavételi pont között a lineáris interpoláció helyett a polinomfüggvény adja meg a függvény értékét. A polinommal történő közelítés az esetek többségében pontosabb közelítést tesz lehetővé a lineáris interpolációnál. Problémája a polinom modellnek, hogy a mintavételi tartományon kívül a polinomfüggvény nagyon el tud távolodni a mintavételezett függvényről, ezért ezeken a szakaszokon a hiba hatványozottan növekszik. Egy másik kellemetlen jellemzője a polinom modellnek, hogy amíg a hagyományos

NLDM, NLPM modelleknél a késleltetés és energia értékek közvetlenül kiolvashatóak az adatbázisból, addig a polinom modelleknél az adatbázisban a polinomok együtthatói vannak eltárolva. Az együtthatókból csak további matematikai számítások után kaphatjuk meg a függvény adott helyen vett értékét. Ez szintén a sebesség csökkenésével jár.

Az előzőekben bemutatott nemlineáris modellek általános problémája, hogy nagyon nehezen skálázhatóak különböző tápfeszültség, hőmérséklet és gyártási paraméter értékekre. Jelenleg a cellakönyvtárakat előre meghatározott, rögzített tápfeszültség, hőmérséklet és gyártási szórás értékekre karakterizálják. A nemlineáris modelleket tartalmazó adatbázisokban ezek a szélsőértékek rögzítve vannak és minden különböző szélsőérték-kombinációra új karakterizáció szükséges és másik adatbázis keletkezik. Egy szimulátor vagy egy elhelyező algoritmus, mely ezeket az adatbázisokat használja fel, csak a megadott szélsőértékek szerint tudja elvégezni a szimulációt vagy az elhelyezést, huzalozást.

A nemlineáris modellek említett hibáinak kiküszöbölésére és a modern, nanométeres csíkszélességű mikroelektronikai sztenderd cellák másodlagos fizikai effektusainak leírására új modelleket dolgozott ki a Synopsys és a Cadence cég. Az összetett áramforrású (CCS, Composite Current Source) és az ECSM (Effective Current Source Model) modellekkel olyan időzítési és fogyasztási adatbázisokat lehet generálni, melyekben a származtatott mennyiségek (például teljesítmény) helyett áramgörbék vannak eltárolva. Ezek az adatbázisok a bemeneti jellemzőség és a kimeneti terhelő kapacitás paraméter értékekhez nem egyetlen fogyasztás vagy késleltetés értéket tárolnak el, hanem egy áram-idő függvényt. Így tehát az ilyen táblázatok négy dimenziósak lesznek. A módszer előnye, hogy az áram-időfüggvényből a legtöbb mennyiség számítható, mint például a fogyasztás, a töltések mennyisége, stb. Ez a fajta karakterizáció lehetővé teszi, hogy a különböző tápfeszültségekre és hőmérsékletekre könnyebben lehessen skálázni az adatbázist. A fogyasztás meghatározásához például elegendő a cella táp lábához tartozó áram-idő függvényt megszorozni a tápfeszültséggel. Így a fogyasztási adatok tetszőleges tápfeszültség értékre meghatározhatók, nem kell újrakarakterizálni a cellakönyvtárat arra az adott tápfeszültségre[29]. A CCS modellezési eljárást a Synopsys cég a Liberty adatbázis formátumba integrálta. A CCS modellezést mutatja be a 4.5. szakasz.

4.5. Összetett áramforrású teljesítmény modell (CCS)

A 90 nm alatti csíkszélesség tartományban a pontos működés modellezéséhez a könyvtári modelleknek pontosan reprezentálniuk kell a cellák összetett, tranzisztor szintű viselkedését. Az áram-alapú (current-based) modellezés az elmúlt néhány évben jelent meg az iparban és hatékony módjává vált a nanométeres időzítési és másodlagos jelenségek modellezésének. Mindezek ellenére azonban a pontos időzítési modell egymagában nem elégséges, mivel a különböző *időzítés*, *teljesítmény* és *zaj* hatások kölcsönösen függenek egymástól. Az egymástól függő paraméterek hatékony kezelése érdekében az integrált áramkör tervezőknek olyan modellekre van szükségük, melyek lehetővé teszik a három kategória egyidejű analízisét és optimalizációját.

Ebben az összefoglalásban bemutatom azokat a nanométeres tervezési nehézségeket, melyek az áram-alapú modellek használatához vezettek és azt, hogy a CCS modellezés hogyan ad választ a nanométeres tervezés egyre növekvő követelményeire. Mérések alapján a CCS

modellek a SPICE alapú analóg szimulációk eredményeit kevesebb mint 2%-os pontossággal közelítik meg [30].

4.5.1. A nanométeres tervezés kihívásai

Az NLDM és NLPM modelleket az integrált áramkör tervező ipar az ezredforduló óta használja. Ezek a modellek a cellák késleltetés és teljesítmény adatainak táblázatait tartalmazzák minden lehetséges bemeneti meredekségre² és terhelő kapacitás értékre. 90 nm alatti technológiáknál megjelenő sok új hatás már nem modellezhető ezzel a megközelítéssel. Néhány ezek közül:

- Nagyimpedanciás összeköttetések.

Az összeköttetések hálózata több vezetékezési rétegen teremti meg a kapcsolatot a különböző áramköri elemek között. Figyelembe kell venni a vezetékek közötti kapacitív csatolást és az ebből adódó áthallási jelenségeket (pl. a vezeték oldalfali kapacitása miatt).

- Miller-hatás.

Ha a vezetékeзések szórt kapacitásai egy erősítő be- és kimenete között helyezkednek el, hatásuk megsokszorozódik. Az erősítő bemenetén sokszoros bemenő kapacitás mutatkozik. Ez nagyobb terhelést jelent egy előző fokozat számára.

- Dinamikus feszültségesés (a továbbiakban IR-drop).

Az integrált áramkörök vezetékezésén az átfolyó áram és a vezeték véges ellenállása miatt a vezeték egyik végén lévő feszültség nem egyezik meg a másik végén lévő feszültséggel. Emiatt például egy digitális rendszerben a logikai magas (1) érték feszültsége egy hosszú vezeték meghajtótól távolabbi végén akár a komparálási feszültség környékére is csökkenhet, ami a helyes működést is befolyásolhatja.

- Dinamikus feszültség és frekvencia skálázás (Dynamic Voltage and Frequency Scaling, DVFS).

A modern, fogyasztásra optimalizált rendszerekben a tápfeszültséget és a működési frekvenciát adaptívan állítják a végrehajtott feladat függvényében. Jellemzően amikor a rendszer üresjáratban van, vagy kevés, kis számításigényű műveletet kell végezni, a tápfeszültséget és a frekvenciát is lecsökkenti a rendszer a kisebb fogyasztás elérése érdekében.

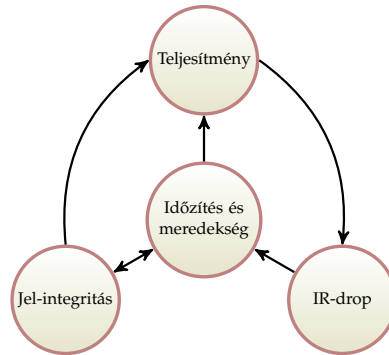
- Hőmérséklet inverzió

Nanométeres integrált áramkörökben a növekvő hőmérséklet nem feltétlenül okozza az áramköri elem működésének lassulását. Előfordulhat, hogy a megnövekvő hőmérséklettel az áramköri elem gyorsabban működik (lásd 6.1. szakasz).

Tovább nehezíti a modellezést, hogy több hatás ezek közül kölcsönösen összefügg a 4.5. ábra szerint. Az időzítés és a jelmeredekségek befolyásolják a fogyasztást, ami hatással van az

² Digitális jeleknél a logikai magas és alacsony értékek átmenete között eltelt idő. A fel- és lefutási idő jellemzően a logikai végértékek 10 %-a és 90 %-a között eltelt idő.

IR-drop-ra, amitől megváltozik az időzítés. Az időzítés hatással van a jel-integritásra, és a jel-integritás változása hatással lehet az időzítésre például az áthallások miatti késleltetések elcsúszása következtében[31]. Egy másik sokszor elhanyagolt tény, hogy a jel-integritás változása hatással lehet a fogyasztásra, ami közvetetten befolyásolja az időzítést. A tervezők sokszor figyelmen kívül hagyják azokat a tüskéket (glitch) a jelalakban, melyek nem jutnak el egy regiszterhez és nem okoznak funkcionális hibát. Mindazonáltal sok ilyen túske megnöveli a fogyasztást, és így tovább.



4.5. ábra. Különböző hatások kölcsönös függése

Az áram-alapú modellezési megközelítés legfőbb előnye, hogy a felvett áram-karakteristikák nem származtatott mennyiségek, de segítségükkel minden származtatott mennyiség egyszerűen számítható (például teljesítmény, feszültség egy adott ponton az impedancia ismeretében, stb.). A CCS modellezést megelőzően volt olyan áram-alapú időzítési modell, mely feszültség-görbéket tárolt el, ami egy származtatott mennyiség. Ezekből az adatokból az áramfüggvények nem, vagy csak korlátozottan állíthatóak vissza. További probléma a feszültség-görbék tárolásával, hogy a karakterizáció más tápfeszültség értékekre nem skálázható egyszerűen. Egy pontos áram-alapú modell közvetlenül az áram mennyiségeken alapul és eljárást tartalmaz az áram-idő függvények mintavételi pontjainak redukálására a pontosság megtartása mellett.

4.5.2. A CCS modellezés

A CCS modell szorosan kapcsolódik a Synopsys cég által kifejlesztett Liberty formátumú könyvtárhoz. A CCS modell által reprezentált időzítés, teljesítmény és zaj paraméterek áramfüggvényeit egy Liberty formátumú szöveges fájl tárolja.

A CCS modellek a cellák időzítés, teljesítmény és zaj karakterizációs adatait tárolják. Mivel a CCS modell áram-függvényeket tartalmaz, a cellák viselkedésének tápfeszültség- és a hőmérsékletfüggő skálázása egyszerűvé válik. A skálázás alkalmazható az időzítési, teljesítmény és a zaj szimulációknál egyaránt és sokkal pontosabb a hagyományos interpolációnál. Az egyszerű tápfeszültség és hőmérsékleti skálázhatóság miatt kevesebb szélsőértékre (corner) kell karakterizálni a cellakönyvtárat és nagy könnyebbséget jelent az alacsony fogyasztású és dinamikus feszültség/frekvencia skálázású (DVFS) áramkörök tervezésénél.

4.5.3. A CCS teljesítmény modell

A CCS-alapú teljesítmény modell segítségével nagyobb időbeli felbontás érhető el a hagyományos NLPM modelleknél. A CCS teljesítmény modell a felvett áramfüggvényeken alapul, így a teljesítmény modell tartalmazza a statikus szivárgás és a dinamikus kapcsolási aktivitás teljesítményeit is. A modell skálázhatósága miatt a különböző tápfeszültségekhez nem kell újra karakterizálni a cellakönyvtárat.

A modell minden cella eredő parazita kapacitását és ellenállását is tárolja, amikor a cella nem aktív. Ennek segítségével gyorsan lehet számítani az áramkörön kialakuló teljesítményeket és áramokat, beleértve a dinamikus feszültségesés (IR-drop) jelenségét a tápsín hálózatban [30].

Az NLPM könyvtár modell sokáig megfelelő volt, de mára világossá vált, hogy az egyszerű táblázatos modell olyan hiányosságokat tartalmaz, melyek gátolják széleskörű alkalmazását és a fejlett tervezési technikák használatát. Ez vezetett a CCS teljesítmény könyvtár modell megalkotásához. A CCS egy áram-alapú teljesítmény modell a következő jellegzetességekkel:

- széles tartományban alkalmazható könyvtár formátum (teljesítmény analízis, optimalizáció, megbízhatóság analízis),
- az NLPM modellhez képest nagyobb időbeli felbontású teljesítmény analízis támogatása,
- ekvivalens parazita hatások az IR-drop analízis végrehajtásához,
- sztenderd cella és makró-cella modellezés.

4.5.4. CCS szivárgási teljesítmény modellezés

A CCS modell a szivárgási teljesítményt szivárgási áramként modellezi, melynek segítségével több-tápfeszültségű cellák részletes áramanalízise is lehetővé válik. Ezek mellett a szivárgási áram nem függ a referencia feszültségtől, a szivárgási teljesítménnyel ellentétben. Ennek segítségével a feszültség-skálázás több könyvtár közötti interpolációval válik lehetővé.

4.5.5. CCS dinamikus teljesítmény modellezés

A CCS dinamikus teljesítmény karakterizáció jellemzői:

- a Liberty adatbázis áramfüggvényeket tárol,
- a töltés számítható az áramfüggvényekből

Dinamikus áram és töltés

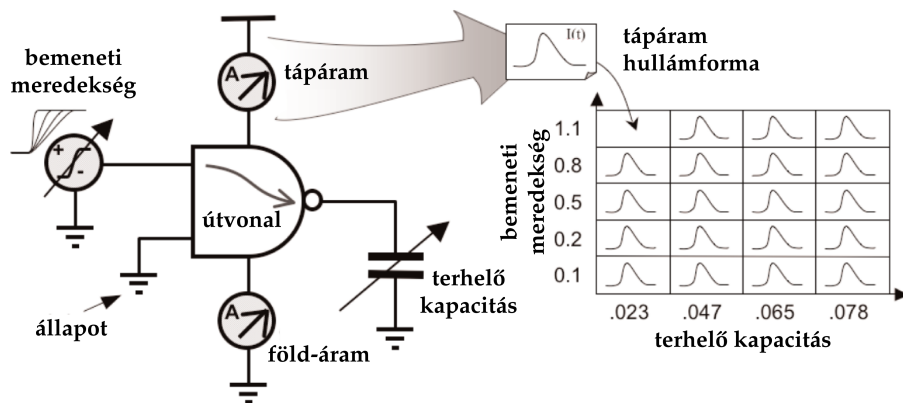
A szivárgási teljesítmény mellett szükség van a dinamikus teljesítmény adatok eltárolására olyan formában, mely független a referencia feszültségtől. A teljesítmény függ a feszültségtől, így olyan mennyiséget kell választani, mely független a feszültségtől, ugyanakkor könnyen számítható az áramfüggvényekből. Ilyen mennyiség például a *töltés*, mely az áram idő szerinti integráljából számítható. A dinamikus energia már számítható a töltésből és a tápfeszültségből (4.5)-tel.

A dinamikus energia meghatározható a belső és a kapcsolási energia összegeként is. Ezt a komponenst (4.3) segítségével számolhatjuk további könyvtári adatok hiányában.

CCS áramfüggvények

A CCS modellben a karakterizáció során közvetlenül az áramfüggvények kerülnek be a Liberty könyvtárba. Minden bemeneti meredekség és kimeneti terhelő kapacitás értékhez tartozik egy áram-idő függvény. Ez látható a 4.6. ábrán. Ezek az eredmények négy dimenziós táblázatok formájában foglalnak helyet a könyvtár fájlban. A négy dimenzió a bemeneti meredekség, a kimeneti terhelő kapacitás az idő és az áram. Másképp mondva, minden bemeneti meredekség és kimeneti terhelés kombinációhoz tartozik egy áram-idő függvény. Ez lényeges különbség az NLPM/NLDM modellekhez képest, ahol egy (bemeneti meredekség, kimeneti terhelés) párhoz egyetlen skalár érték tartozik. NLPM modell esetén ez a skalár érték energia mennyiség, NLDM modell esetén idő.

A Liberty adatbázis fájl méretének csökkentése érdekében az áram-idő függvények nem minden pontját tárolják el. A SPICE analóg szimulációk minden pontjának eltárolása felesleges és helypazarlás lenne, mely a feldolgozó alkalmazások sebességét is lassítaná. Ezt szem előtt tartva a könyvtárban az áram-idő görbéket szakaszonként lineáris függvényekként tárolják el. Az (I, t) értékpárok sorozata reprezentálja az áramgörbét, ahogy azt a 4.7. ábra is mutatja. A kiemelt (I_n, t_n) pontok kerülnek a könyvtárba. A karakterizációs kulcspontok helyének meghatározása a cellakönyvtárat karakterizáló gyártó dolga. A kulcspontok közötti értékeket lineáris interpolációval számítják ki az alkalmazások. A dinamikus áramfüggvényből a töltést



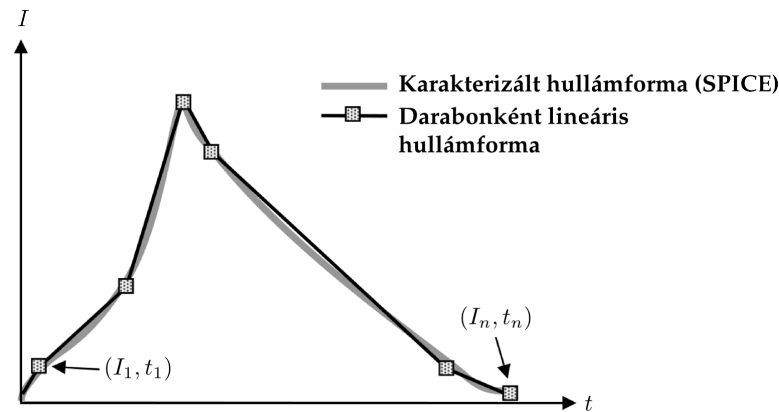
4.6. ábra. Dinamikus áramgörbe-mérés

numerikus integrálással lehet meghatározni (4.7) szerint [32]:

$$Q_{\text{din}} = \sum_{i=1}^n \frac{1}{2} (I_i + I_{i-1}) (t_i - t_{i-1}). \quad (4.7)$$

4.5.6. CCS összefoglalás

A CCS modellezési technológia a mai digitális áramkörök egyre bonyolultabb és szerteágazó tervezési problémáinak megoldását segíti. A modell figyelembe veszi a nanométeres mérettartomány másodlagos fizikai hatásait is. A nyílt forráskódú Liberty formátum tartalmazza azt az adathalmazt, mely szükséges az időzítés, teljesítmény és zaj analízisek pontos és hatékony



4.7. ábra. Szakaszonként lineáris áram hullámforma

elvégzéséhez a nanométeres csíkszélességű digitális áramkörökben. A Liberty formátum és a CCS modell bárki által szabadon bővíthető, így olyan funkciókkal is kiegészíthető, melyek jelenleg nincsenek benne a specifikációban.

Válaszul a Synopsys Liberty formátumára, a Cadence cég kifejlesztette saját áram-alapú megoldását, az ECSM (Effective Current Source Model) modellt [29]. Ez a modell szervesen épül a Liberty formátumra és abban újabb attribútumokat definiál főleg a statisztikai analízis megkönnyítésére (például Monte Carlo analízis). Az ECSM modell ugyanazokat a karakterizációs eljárásokat használja, mint a Liberty formátum, és a Liberty formátummal teljesen kompatibilis módon tárolja a karakterizált értékeket az adatbázisban. A Synopsys, Cadence és Mentor cégek széles körben használt digitális integrált áramkörtervező rendszerei mindkét formátumot támogatják.

A teljesség kedvéért meg kell említeni a Cadence cég által kifejlesztett TLF (Timing Library Format) formátumot is, mely szintén időzítési és teljesítmény adatokat tartalmaz az NLPM és NLDM modellek alapján [33]. A gyártók tervezési szabálykészletében (Process Design Kit, PDK) általában mind a TLF, mind a Liberty formátumú adatbázisok megtalálhatóak. Az integrált áramkör tervező szoftvercsomagok általában tartalmaznak konverter programokat, melyek segítségével a TLF és Liberty formátumok közötti átalakítást lehet végrehajtani.

A 5. fejezetben bemutatott eljárás kidolgozása során olyan tervezői eszközkészletet is felhasználtam, melyben nem állt rendelkezésre a Liberty adatbázis (Mentor Graphics ASIC Design Kit, ADK). A cellák fogyasztási adatait –melyeket a kialakult hőmérsékleti térkép előállítására használtam fel– az ADK eszközkészlet esetében SPICE szimulációkkal kellett meghatározni. A SPICE szimulációkat a CCS modellezés leírása alapján hajtottam végre. A SPICE szimulációk eredményeként adódó kapcsolási energia értékeket Liberty formátumú adatbázisban tároltam el. Az eszközkészlet részeként kapott illetve az analóg szimulátorral előállított adatok így egységes formátumban állnak a logi-termikus szimulátor rendelkezésére.

A teljesítmény karakterizációt analóg SPICE szimulációkkal a 4.6. szakasz mutatja be.

4.6. Teljesítmény karakterizáció SPICE szimulációkkal

A logi-termikus szimulátor a cellák fogyasztási adatait a tervezői eszközkészlet (Process Design Kit) részeként rendelkezésre álló Liberty adatbázisból veszi. Ezt az adatbázist SPICE szimulációk segítségével az integrált áramkör gyártó karakterizálja. Ezzel foglalkozik részletesen a 4. fejezet első része. A karakterizáció kiterjed az időzítések, zajparaméterek és a teljesítmények meghatározására is. Az iparban megtalálható tervezői eszközkészletek rendszerint tartalmazzák a Liberty adatbázist több különböző szélsőértékre (corner) karakterizálva. Ez alapján a cellákról pontos karakterisztikus adatokat kaphatunk. A szintézer, az elhelyező és huzalozó alkalmazások többek között a Liberty adatbázis segítségével hajtják végre feladatukat.

Abban az esetben, ha valamilyen okból az adott sztenderd cellás áramkör csatláchoz nem áll rendelkezésre ilyen Liberty adatbázis, tranzisztor szintű SPICE szimulációk futtatásával a hiányzó adatok meghatározhatóak[J1, J3, J4],[C2, C3, C4].

A kapcsolási aktivitás által okozott fogyasztást a következőképpen lehet SPICE szimulációkkal meghatározni. A cellákat megfelelő terheléssel lezárva és gerjesztésnek egy –az adott alkalmazásban előforduló– fel- és lefutási idejű négyszögjelet választva meg tudjuk határozni a cella fogyasztását bármilyen bemeneti kombinációra. Sorrendi cella esetén (pl. D flip-flop) a szimuláció megfelelő vezérlésével a cellát a vizsgálandó állapotba lehet hozni. Az így meghatározott fogyasztási adatok eltárolhatóak egy Liberty adatbázisban, amiből a logi-termikus szimulátor a kapcsolási energiákat veszi. Fontos megjegyezni, hogy a Liberty adatbázis a kapcsolási fogyasztási adatokat energia dimenziójú mennyiségként tárolja, jellemzően pJ-ban.³

Az energiaértékek meghatározásához ad segítséget a 4.8. ábra. A bemeneti gerjesztést egy külső négyszögjellel biztosítjuk a szimulációhoz. Hatékony és bevált módszer az is, ha a gerjesztő jelet nem egy ideális négyszögjel generátor segítségével biztosítjuk, hanem az adott cellakönyvtárból kiválasztjuk a legnagyobb meghajtóképességű puffert vagy két invertert, és ennek a bemenetére kapcsoljuk az ideális négyszögjel generátort. A puffer cella kimenetét kell ilyenkor a vizsgált cella bemenetére kapcsolni. Ezzel a megoldással a négyszögjel fel- és lefutási idejei a cellakönyvtárra jellemző idők lesznek[J1].

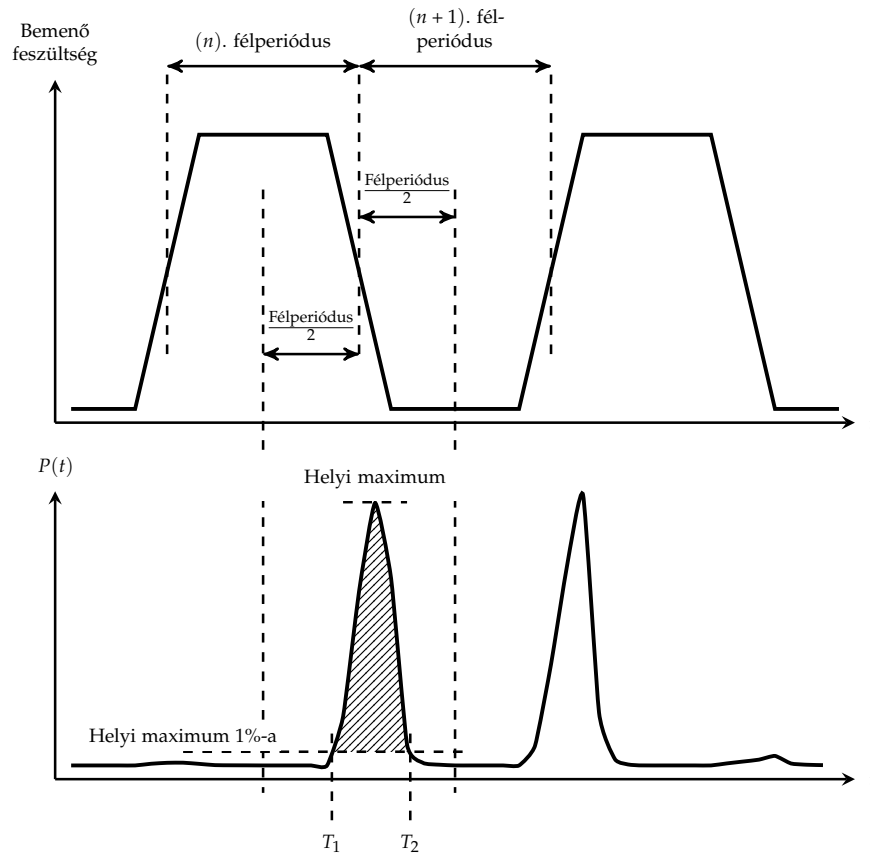
A SPICE szimulátorban a cellák fogyasztási tranziens görbáját közvetlenül meg lehet jeleníteni (a tápfeszültséget kell megszorozni a cella adott kimenetének áramával). A 4.8. ábrán látható, hogy a bemeneti tranziens hatására a kimenet $P(t)$ fogyasztási függvényében tüske keletkezik. Ez a dinamikus fogyasztás. A Liberty adatbázis számára a kapcsolási energiát a görbe alatti terület kiszámításával tudjuk meghatározni.

Ahogy a 4.8. ábrán megfigyelhető, a kapcsolási energiát a teljesítmény görbe kapcsolás körüli környezetének integrálja adja:

$$E_{kaps} = \int_{T_1}^{T_2} P(t) dt. \quad (4.8)$$

A T_1 és T_2 időket úgy lehet meghatározni, hogy vesszük a fogyasztási görbe helyi maximum értékét a kapcsolás környezetében. A maximum érték 1 %-ának metszéspontja a teljesítmény görbével adja az integrálási határokat. Megfelelő módszer lehet az is, hogy a logikai átmenet

³ Az energiák mértékegysége a Liberty adatbázisban meghatározható a feszültségek és a kapacitások mértékegységéből. Például: $E = C \cdot U^2$; [pF] · [V²] = [pJ].



4.8. ábra. Disszipáció logikai jelváltás esetén

közelében $\pm$ negyed periódussal integráljuk a teljesítmény görbét. Ez az érték tartalmazni fogja a statikus fogyasztást is és valamennyivel felül is becsli a tényleges kapcsolási energiát.

Az ilyen módon meghatározott és eltárolt energia értékeket fogja a logi-termikus szimulátor cellánként összegezni egy termikus időablakon belül (pl. $dt = 10$ ms), majd az összenergiát elosztva a termikus időablak szélességével megkapjuk az átlagos teljesítményt, amit a cella abban az időablakban fogyaszt és amiből a Joule-hő származik:

$$P_{\text{fűtő}} = \frac{E_{\text{kapcs}} \cdot n}{dt}. \quad (4.9)$$

5. fejezet

Logi-termikus szimuláció EDA környezetben

1. tézis. Új módszert és erre épülő szimulátort dolgoztam ki korszerű sztenderd cellás, digitális integrált áramkörök logi-termikus szimulációjára. A logi-termikus szimuláció a digitális integrált áramkörök cella szintű logikai és termikus együttes szimulációjára ad lehetőséget. A szimuláció során a pillanatnyi teljesítmény-eloszlást, mint folyamatosan változó gerjesztést csatolom egy termikus szimulátorba. A módszer segítségével az áramkör felületén működés közben kialakuló és változó hőmérséklet-eloszlás nyomon követhető és a kiugróan magas hőmérsékletű területek (hot-spotok) felderíthetőek. Két új módszert dolgoztam ki a logi-termikus szimulációk gyorsítására, melyek együttesen és egymástól függetlenül is alkalmazhatóak [J3],[C1, C2, C3].

1.1. altézés. Kidolgoztam egy modern EDA (Electronic Design Automation) környezetekhez illeszkedő, szabványos, logi-termikus elven működő szimulátort. A szimulátor a fogyasztások és időzítések meghatározására olyan adatbázist használ, amely a kvantumfizikai hatásokat is figyelembe veszi. A logikai és termikus szimulátorok összekapcsolása az általam kidolgozott kommunikációs eszköz segítségével egy hardver leíró nyelv szabványos illesztő felületén keresztül történik. A kidolgozott módszer szabványos interfészekon keresztül kommunikál az ipari logikai szimulátorokkal, szabványos bemeneti- és kimeneti fájlformátumokkal dolgozik. A módszer tetszőleges EDA környezethez illeszthető [J1, J3, J4],[C4, C5, C6].

1.2. altézés. Új gyorsítási módszert dolgoztam ki, amely a térbeli felbontás csökkentésén alapul. Az általam kidolgozott módszer a sztenderd cellás áramkör felületét szükség esetén partíciókra bontja, és ezekre a partíciókra számolja ki a sztenderd cellák fogyasztásából adódó hőmérsékleteket. Az áramkört felépítő sztenderd cellák számának növekedésével négyzetesen növekszik a termikus modell generálásához szükséges idő. A particionálással a sztenderd cellák számától függetlenné tehető a termikus modell generálásának időtartama. A particionálási módszer használata a felhasználó számára opcionális, és a termikus szimulációban alkalmazott térbeli felbontás mérete is szabadon megválasztható. Különböző partíció méretek és modellgenerálási idők vizsgálata alapján kimutattam, hogy a 10×10 -es particionálás a felbontás és a számítási idő szempontjából elfogadható [J2],[C7].

1.3. altézis. Új gyorsítási módszert dolgoztam ki a szimuláció időbeli felbontásának skálázásával. Egy modern digitális rendszer termikus időállandói nagyságrendekkel nagyobbak az elektromos időállandóknál (pl. órajel periódusidő). A logikai szimuláció nagy órajel frekvencián hosszú ideig tarthat a termikus állandósult állapot megközelítéséig. A nagyfrekvenciás rendszer működési frekvenciáját lecsökkentve és a kapcsolási tranziensekhez tartozó dinamikus fogyasztást arányosan növelve, a rendszer töredék idő alatt a termikus tranziens tetszőleges állapotába hozható. A kitűzött termikus állapotot elérve az eredeti működési frekvencián és fogyasztási értékekkel folytatható a szimuláció [C8].

5.1. A logikai- és termikus szimulátorok összekapcsolása

Az elektro-termikus szimulátoroknak kétféle implementációja lehetséges[34, 35]. Az egyik módszer a közvetlen (vagy direkt) módszer, melynek alkalmazása során a tranzisztor modelleket termikus lábakkal egészítik ki. Az áramkör tranzisztorainak termikus lábai a struktúra ekvivalens, termikus RC hálózatra kapcsolódnak. Így egy SPICE analóg szimulációval egy lépésben meg lehet határozni az eszköz elektromos és termikus viselkedését. A termikus-elektromos kölcsönhatások a modellbe vannak építve különböző differenciálegyenletek formájában. Ennek a megközelítésnek előnye, hogy pontos elektro-termikus szimulációk végezhetők el a kereszt-effektusok figyelembevételével. A közvetlen módszerű elektro-termikus szimuláció azonban az alacsony absztrakciós szint (tranzisztor szint) miatt egy bonyolult, többmillió tranzisztort tartalmazó áramkör esetében hosszú ideig tart[36, 37, 38].

Egy másik lehetséges módszer az ún. relaxációs módszer, melynek során a termikus és elektromos szimulátorok egymást váltva működnek[39]. A tisztán elektromos szimulátor (mint pl. a SPICE, vagy logikai szimulátor) az áramkör elektromos működését szimulálja és meghatározza minden alkatrész fogyasztását az adott szimulációs időlépésben. Az így nyert fogyasztási adatokat továbbküldve a termikus szimulátornak, az meghatározza az alkatrészek hőmérsékletét, ami aztán visszahelyettesíthető a következő elektromos szimulációba, ezzel megváltoztatva az eszközök karakterisztikáját.

Értekezésemben olyan logi-termikus szimulációs eljárást dolgoztam ki, mely a relaxációs módszert használja fel. A teljesítmény adatokat egy logikai szimulátor állítja elő a kapcsolási aktivitásból és a sztenderd cellakönyvtár fogyasztási táblázatai alapján. Ezeket a teljesítmény adatokat a LayTherm termikus szimulátor kapja meg, mely előállítja a felületi hőmérséklet-eloszlást az áramkör felszínén az adott szimulációs időablakban. A szimuláció futása közben minden időlépésben megtörténik a kapcsolási aktivitás kiszámítása, a teljesítmény adatok meghatározása és a termikus szimuláció. Az általam használt relaxációs módszer segítségével lehetőség nyílik magasabb absztrakciós szinteken (cella szinten vagy RTL szinten) a logi-termikus szimulációkra.

Magasabb absztrakciós szinten az előre karakterizált cellakönyvtár adatainak segítségével a szimuláció jelentősen felgyorsítható egy elektro-termikus szimulációhoz képest. Ez a tulajdonság megfigyelhető egy tisztán elektromos analóg és egy logikai szimuláció között is. Ha ugyanazt a digitális áramkört analóg SPICE szintű szimulációval vizsgáljuk tranzisztor szinten illetve strukturális cella szinten logikai szimulációval, a logikai szimuláció nagyságrendekkel rövidebb idő alatt fut le, mint az analóg szimuláció. Ha a cellakönyvtárat a integrált áramkör-

gyártó megfelelően karakterizálta, akkor a logikai szimuláció és az analóg SPICE szimuláció eredménye jó közelítéssel egyezni fog.

Az általam kidolgozott eljárásban a LayTherm termikus szimulátor és a QuestaSim logikai szimulátor összekapcsolását a Verilog HDL specifikációban[9] ismertetett Verilog PLI programozói interfésszel implementáltam. A Verilog PLI interfész használatát az 5.2.1. alszakaszban mutatom be.

5.2. A logi-termikus szimuláció folyamata

1.1. altézis. *Kidolgoztam egy modern EDA (Electronic Design Automation) környezetekhez illeszkedő, szabványos, logi-termikus elven működő szimulátort. A szimulátor a fogyasztások és időzítések meghatározására olyan adatbázist használ, amely a kvantumfizikai hatásokat is figyelembe veszi. A logikai és termikus szimulátorok összekapcsolása az általam kidolgozott kommunikációs eszköz segítségével egy hardver leíró nyelv szabványos illesztő felületén keresztül történik. A kidolgozott módszer szabványos interfészekén keresztül kommunikál az ipari logikai szimulátorokkal, szabványos bemeneti- és kimeneti fájlformátumokkal dolgozik. A módszer tetszőleges EDA környezethez illeszthető [J1, J3, J4],[C4, C5, C6].*

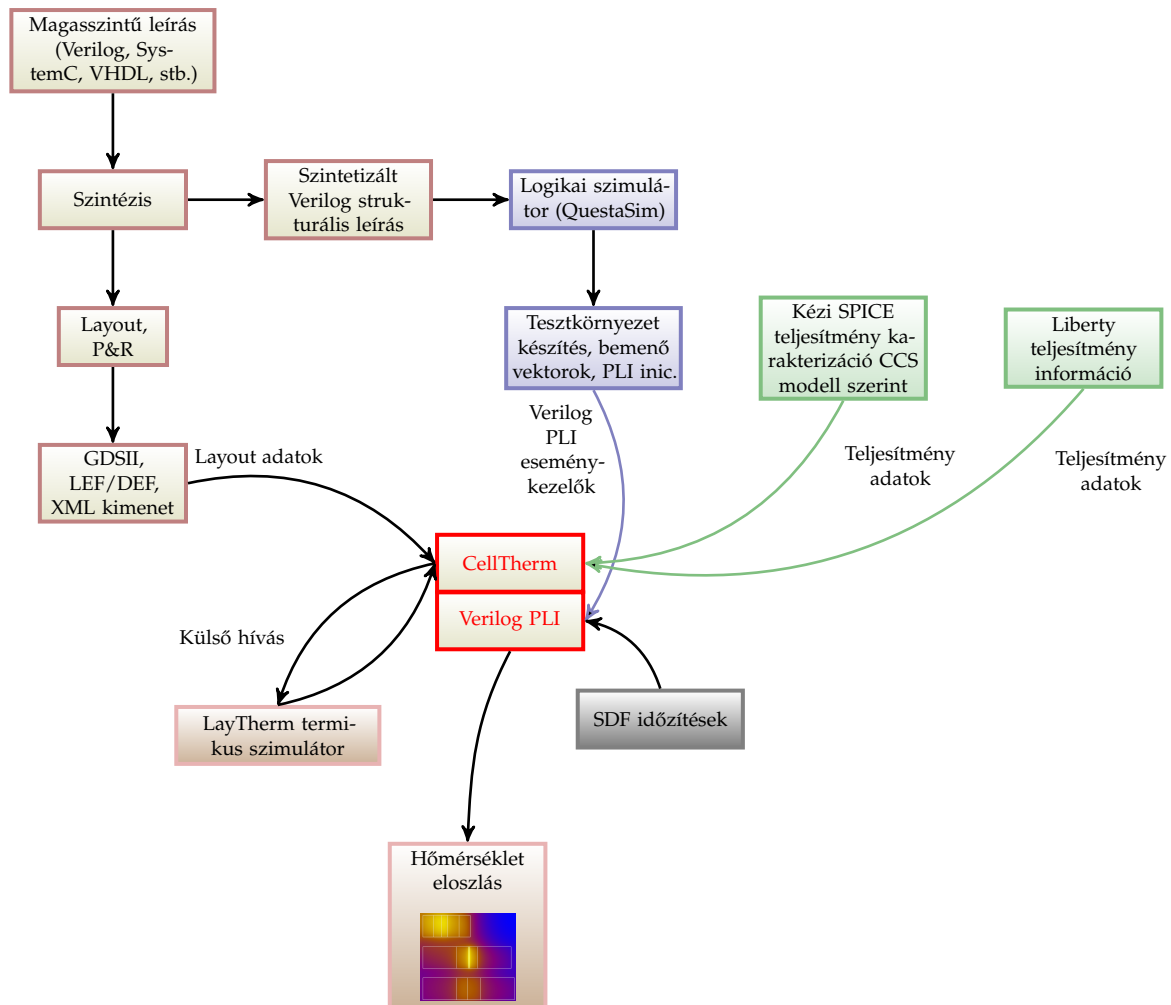
A logi-termikus szimuláció folyamatát az 5.1. ábra mutatja be. A folyamat egyes lépéseit részletesen a következőkben mutatom be.

A logi-termikus szimuláció folyamata két párhuzamos feladattal kezdődik. Egyrésről a hardver specifikációját meg kell valósítani valamilyen magas szintű, vagy hardver leíró nyelven. Ez a digitális hardver tervezésben megszokott módon történik. Másrészt a sztenderd cellakönyvtár celláinak teljesítményét karakterizálni kell, amennyiben ez nem áll rendelkezésre például egy Liberty formátumú adatbázis formájában. A két folyamat párhuzamosan történhet, hiszen teljesen függetlenek egymástól.

1. A folyamat elején az áramkör magas szintű leírásából, mint például a SystemC, rendszert egy hardverleíró nyelvű leírás készül Verilog vagy VHDL nyelven. Ezt a magasszintű szintézer alkalmazás automatikusan állítja elő. Az így elkészült HDL leírás már közvetlenül szintetizálható digitális hardverre a digitális cellakönyvtár és egy szintézer alkalmazás segítségével.
2. A hardverleírás szintézisével előállítjuk a szintetizált és konkrét digitális cellákkal megvalósított (angol terminológiával *mapped*) leírást. Az így előállított leírásból készül automatizáltan a felületi topológia terve (layout) és az áramkör strukturális leírása is.

Az *elhelyezés és huzalozás* (Place and Route, P&R) folyamat eredményeképpen keletkezik a fizikai topológia (layout), melyre a termikus viszonyok térbeli számításához van szükség. A szintézer, elhelyező és huzalozó alkalmazások szabványos formátumban állítják elő a fizikai topológiát leírását. A legismertebb szabványos formátumok a GDSII[40], LEF/DEF[41], Oasis[42]¹ és OpenAccess[43] formátumok. A különböző layout leíró formátumok konvertálhatóak egymás között és az integrált áramkörgyártó ezek segítségével készíti el a gyártáshoz szükséges maszkokat.

¹ SEMI P39-0308 - OASIS - Open Artwork System Interchange Standard.



5.1. ábra. A logi-termikus szimuláció működése

3. A strukturális leírást a logikai szimulátor segítségével szimuláljuk a megfelelő tesztkörnyezet definiálásával. A tesztkörnyezetet speciálisan úgy definiáljuk, hogy tartalmazza a tesztvektorokat és a logi-termikus szimulációhoz szükséges Verilog PLI csatoló függvényeket. Ezeken a függvényeken keresztül kommunikál a logikai szimulátor a termikus szimulátorral.

Ezen a szinten már nem a viselkedési leírást szimuláljuk, hanem azt az áramkört, amely sztenderd cellákból összeállítva azt a funkciót hajtja végre, amit a magas szintű viselkedési leírásban meghatároztunk. A szintézis alkalmazások dolga, hogy a szintézist úgy hajtsák végre, hogy a viselkedési leírásban meghatározott és a szintetizált áramkör működése funkcionálisan megegyezzen. A szintetizált strukturális leírást Verilog HDL formátumban állítom elő annak érdekében, hogy a logikai szimulátorból a Verilog PLI segítségével a termikus szimulátort meghívhassam.

4. A logi-termikus szimuláció elindításához rendelkezni kell a sztenderd cellák fogyasztási adataival. Ha tudjuk egy cella statikus és dinamikus energiafogyasztását, akkor a kapcsolási aktivitásból meghatározható az egy időlépésben fogyasztott teljesítmény, ami közvetlenül okozza az eszköz hőmérsékletének változását. Az energia információk

kinyerhetők a 4. fejezetben ismertetett Liberty adatbázisból, vagy ennek hiányában az analóg szimulációkkal meghatározott cella fogyasztási adatokból. A teljesítmények meghatározására az analóg szimulációkat a CCS modell által megfogalmazott irányelvek és mérési módszerek alapján végeztem.

5. A cellák késleltetési adatait a szintézer alkalmazás által automatikusan elkészített SDF (Standard Delay Format) adatbázis tartalmazza. Ezek az időzítések lehetnek az elhelyezés és huzalozás lépés utáni (post-layout) késleltetések is.

Amennyiben szeretnénk meghatározni a kialakuló hőmérsékletek hatását a cellák késleltetéseire, akkor a szimuláció megkezdése előtt szükség van a szintézer alkalmazás által elkészített cella-késleltetéseket tartalmazó SDF (Standard Delay Format) adatbázisra is.

6. A CellTherm alkalmazást adott időközönként Verilog PLI hívásokon keresztül meghívja a logikai szimulátor. A logikai szimulátor a felhasználó által beállított *termikus időlépésenként* hívja meg a termikus szimulátort a CellTherm-en keresztül. A termikus időlépés vagy időablak mérete jellemzően sokkal nagyobb, mint a modern digitális rendszerben előforduló időzítési értékek és órajel periódusidők. A vizsgált integrált rendszerek domináns termikus időállandói a másodperces, vagy akár a perces nagyságrendbe esnek, így a termikus tranziens számításoknál már a milliszekundumos időbeli felbontással is nagyon pontos vizsgálatot lehet végezni.
7. A CellTherm alkalmazás minden termikus időlépésben meghatározza a kapcsolási aktivitás és a logikai átmenetek energia fogyasztása alapján az eredő teljesítmény fogyasztást. A CellTherm ezeket az adatokat továbbítja az integrált termikus szimulátor felé (LayTherm), ami kiszámítja a bemenő teljesítmények és a fizikai topológia alapján a kialakuló felületi hőmérsékleti eloszlást.
8. A LayTherm termikus szimulátor az előállított hőmérsékleti adatokat visszaküldi a CellTherm-nek, ami elkészíti a termikus adatok alapján az áramkör hőmérsékleti térképét az adott időlépésben.

5.2.1. A Verilog PLI interfész

A Verilog PLI (Programming Language Interface)[9] egy olyan programozói felület, melynek segítségével a Verilog kódból C nyelven írt metódusokat lehet meghívni. A külső C kódban tetszőleges program implementálható. A logikai szimulátor először lefordítja (compile) a Verilog forráskódot, majd ezután feldolgozza és szimulációra előkészíti (elaborate). A Verilog HDL szabvány különböző *rendszerfüggvényeket* definiál, melyek a logikai szimulációt könnyítik meg. Ilyen rendszerfüggvény pl. a `$display()` függvény, melynek segítségével a szimuláció közben a sztenderd kimenetre lehet tetszőleges megjegyzéseket kiíratni. Minden Verilog rendszerfüggvény kötelezően a `$` karakterrel kezdődik. A Verilog PLI segítségével saját rendszerfüggvényeket definiálhatunk melyek mögött az általunk írt C kód fut. A felhasználó által definiált rendszerfüggvényeknek bármilyen nevet adhatunk, ami a `$` karakterrel kezdődik és megfelel a rendszerfüggvény elnevezési követelményeknek[9, 20.2. szakasz].

Az 5.1. kódrészletben egy egyszerű Verilog HDL kód látható, ahol meghívjuk a `$celltherm()` függvényt. A függvényhívás hatására az általunk írt C kód fut le, mely az 5.2. kódrészletben látható.

```

5 module pli;

  initial
    $celltherm();

endmodule

```

5.1. kódrészlet. Verilog kód

```

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

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

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

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

```

5.2. kódrészlet. PLI C kód

A felhasználó által definiált C függvényeket dinamikusan betöltődő könyvtárként (.so vagy .dll) kell lefordítani, így a logikai szimulátor ezeket a könyvtárakat futási időben tudja betölteni. Mivel a Verilog szabványos hardver leíró nyelvet támogató logikai szimulátorokat többfajta operációs rendszeren is implementálták (Linux, Unix, Windows), ezért az általam kidolgozott logi-termikus szimulációs eljárás is bármelyik operációs környezetben működőképes. Természetesen az architektúráis különbségek miatt a CellTherm alkalmazást külön le kell fordítani minden használni kívánt operációs rendszerre, ám a forráskódot olyan platform-független C++ osztálykönyvtár segítségével (Qt5) írtam meg, mely hordozhatóvá teszi a kódot különböző rendszerek között.

A Verilog PLI interfésznél a külső C függvényeket regisztrálni kell a logikai szimulátorba, hogy az képes legyen párosítani a Verilog rendszerfüggvényt a hozzá tartozó C kóddal. Ez a regisztráció az `mti_RegisterUserTF()` függvénnyel történik, ez látható az 5.2. kódrészlet 20. sorában.

A Verilog rendszerfüggvények nem szintetizálható nyelvi elemek, csak arra szolgálnak, hogy a logikai és időzíti szimulációkat megkönnyítsék. Az általam kidolgozott eljárás alapján egy digitális rendszer meglévő Verilog tesztkörnyezete (testbench) néhány függvényhívással kiegészítve logi-termikus szimulációt tesz lehetővé, a tesztkörnyezet megváltoztatása nélkül. A tesztkörnyezetre példa az 5.3. kódrészletben látható.

```
'include "ringosc.v"

#define TH_Timestep 1e6 // = 1ms

5  'timescale 1ns / 1ps
module testbench;
    reg clk;
    reg run;
    reg [3:0] data;
10  wire ring_out;
    wire [3:0] dffr_out;

    top dut (clk, run, data, ring_out, dffr_out);

15  initial
    begin
        $celltherm_initlayout(dut,10,10,"ringosc.lib","ringosc.lef","ringosc.def");
        $celltherm_init(dut,1,'TH_Timestep);

20  clk = 0;
    data = 0;
    run = 1;

    end

25  always #1000 clk = ~clk;
    always #2000 data = data + 1;

// Termikus motor hivasá pl. dt = 1 ms
30 // #1 = 1ns
    // #1e3 = 1us
    // #1e6 = 1ms
    always
    begin
35  #'TH_Timestep $celltherm_calc('TH_Timestep, 25);
    end

endmodule
```

5.3. kódrészlet. Verilog tesztkörnyezet

5.3. A szimuláció gyorsítása partíciókra bontással

1.2. altézis. Új gyorsítási módszert dolgoztam ki, amely a térbeli felbontás csökkentésén alapul. Az általam kidolgozott módszer a sztenderd cellás áramkör felületét szükség esetén partíciókra bontja, és ezekre a partíciókra számolja ki a sztenderd cellák fogyasztásából adódó hőmérsékleteket. Az áramkört felépítő sztenderd cellák számának növekedésével négyzetesen növekszik a termikus modell generálásához szükséges idő. A particionálással a sztenderd cellák számától függetlenné tehető a termikus modell generálásának időtartama. A particionálási módszer használata a felhasználó számára opcionális, és a termikus szimulációban alkalmazott térbeli felbontás mérete is szabadon megválasztható. Különböző partíció méretek és modellgenerálási idők vizsgálata alapján kimutattam, hogy a 10×10 -es particionálás a felbontás és a számítási idő szempontjából elfogadható [J2],[C7].

Az általam kidolgozott logi-termikus szimulációs eljárás során a fogyasztási értékeket és az így kialakuló hőmérsékleti eloszlást a sztenderd cellákon veszem figyelembe. A sztenderd cellákat fekete doboznak tekinti a szimulátor, ami azt jelenti, hogy nem ismeri a belső, tranzistor szintű felépítését, működését. A cellákról a logi-termikus szimuláció során csak annyit kell tudnia a szimulátornak, hogy az adott termikus időablakban mennyi kapcsolás történt a cellában (ezt a logikai szimulátor regisztrálja), és egy kapcsolás során mennyi energiát fogyasztott. Ezekből az adatokból számítható ki a fűtőteljesítmény, az a teljesítmény ami az áramkör melegítésére fordítódott.

A szimulált struktúra (chip, tokozás, hűtőborda, stb.) termikus ekvivalens RC helyettesesítőképet (kompakt modelljét) a disszipáló alakzatok helyének és méretének ismeretében tudjuk előállítani. A hőt termelő alakzatok esetünkben a szimuláció legkisebb egységei, a sztenderd cellák lesznek. A kompakt modell meghatározásához minden cella között meg kell határozni a modellt felépítő Foster RC létrát. Ez n cella esetében n^2 darab Foster létrát jelent.

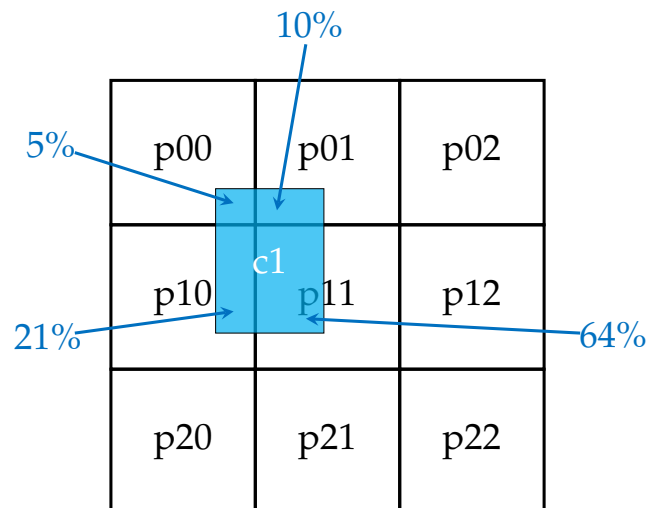
Napjaink digitális integrált áramköreiben a tranzisztorok száma meghaladja a milliárdot. Az Intel Ivy Bridge (Core i7 3770K) processzorában például 160 mm^2 felületen 1,4 milliárd tranzistor található[44]. Ez sztenderd cellákban mérve is több százezer cellát jelent. Ilyen cellamennyiségnél az n^2 darab Foster RC modell létrehozása és számítása nem kivitelezhető és nem is szükséges. A hőmérsékleti eloszlást tekintve az IC felületén egy cella mérete olyan kicsi a teljes chip méretéhez képest, hogy feleslegesen nagy felbontásban számítanánk ki a kialakuló hőmérsékleti térképet. Ezen a felbontáson lehet csökkenteni a gyorsítás érdekében. A csökkentés mértékét a szimuláció gyorsasága és a kívánt felbontás közötti optimális választással adhatjuk meg.

Az általam kidolgozott particionálási eljárás segítségével az integrált áramkör felületét egy ráccsal bonthatjuk fel kisebb területegységekre. A rács vízszintes és függőleges felbontását tetszőlegesen határozhatjuk meg. A logikai szimulátor változatlanul a strukturális leírást szimulálja, a termikus szimulátor viszont a ráccsal meghatározott területegységekre számítja ki a hőmérsékleteket. A „rács” helyett pontosabbnak tartottam a „partíció” megfogalmazást, mivel a rendszer elő van készítve nem egyenletes eloszlású ráccsal történő felosztásra is. Ez azt jelenti, hogy a felhasználó tetszőleges méretű, oldalarányú és mennyiségű téglalapra fel tudja bontani az integrált áramkör felületét. A partíciók méretének és mennyiségének szabad megválasztása lehetővé teszi, hogy az áramkör különböző funkcionális részegységeit (pl. gyorsítótár (cache), aritmetikai-logikai egység (ALU), processzor magok (core), stb.) külön

partícióként kezeljük. Így tetszőleges részegységek termikus aktivitását is figyelemmel tudjuk kísérni egy logi-termikus szimuláció során.

A termikus kompakt modellt a szimulátor a partíciókra határozza meg, és a hőmérsékleti eloszlás felbontását ezek a partíciók határozzák meg. A particionálás jelentősen fel tudja gyorsítani a logi-termikus szimulációt mivel ekkor például 10×10 -es, vagy 20×20 -as felbontást választhatunk, ami az utóbbi esetben 400 partíciót jelent a több százezer cellával ellentétben. A partíciók illetve a cellák számának növekedésével négyzetesen növekszik a termikus kompakt modell hálózat mérete, ami a számítási idők szintén négyzetes növekedésével jár együtt.

A logikai szimulációban a particionálás semmilyen változást nem jelent, ott ugyanúgy a strukturális sztenderd cellás leírást szimuláljuk és így határozzuk meg az áramkör fogyasztását is (a kapcsolási aktivitásból és kapcsolási energiákból). Az egyes partíciók eredő fogyasztását –amire a termikus szimuláció során lesz szükség– úgy számolja ki a szimulátor, hogy összegzi az egy partícióba eső cellák fogyasztását. Minden partíció fogyasztását a benne található cellák együttes fogyasztása adja. Abban az esetben, ha egy cella nem teljes területével tartozik egy partícióhoz, a fogyasztásának területarányos részével szól bele a partíció fogyasztásába. Ennek könnyebb megértését segíti az 5.2. ábra.



5.2. ábra. A cellák és partíciók átlapolódásának számítása

Az 5.2. ábrán a c1 cella éppen négy partíció határára esik. Az egyes partíciókba a cella területének 5, 10, 21 és 64 százaléka esik. A cella fogyasztásának így 5, 10, 21 és 64 százaléka jut a p00, p01, p10 és p11 partíciókba. A fogyasztások meghatározását mutatja az (5.1). egyenletrendszer:

$$P(p00) = 0,05 \cdot P(c1) \quad (5.1)$$

$$P(p01) = 0,10 \cdot P(c1)$$

$$P(p10) = 0,21 \cdot P(c1)$$

$$P(p11) = 0,64 \cdot P(c1).$$

A cellák eredő hőmérsékleteit a partíciók hőmérsékleteiből szintén az átlapolódással arányosan lehet meghatározni, ahogyan azt a teljesítmények esetében is tettük.

Az 5.2. ábra egy elfajult eseten mutatja be a partíciók teljesítményének kiszámítását, mivel a valóságban a partíciók célszerűen jóval nagyobbak az egyes cellák méreteinél, hiszen így tudunk gyorsítani a szimuláción. Egy több ezer cellát tartalmazó áramkörben a 10×10 -es particionálással jelentős sebességnövekedést lehet elérni a pontosság kismértékű romlásával. Egy 10×10 -es particionálással végzett szimuláció eredményét tovább lehet finomítani a partíciók közötti interpolációval is. A bilineáris interpolációt lehet alkalmazni a hőmérsékleti térkép megjelenítésnél is, amivel nagyobb felbontású képet kaphatunk.

5.3.1. A particionálás hibája

A particionálás során a hőmérsékleti térkép felbontását csökkentjük a sebesség növelésének érdekében. A felbontás csökkentése a tényleges cellaszintű hőmérsékleti térképhez képest valamennyi hibát visz be. A particionálás sűrűségével fordítottan arányos a bevitt hiba is, hiszen minél sűrűbbre vesszük a partíciókat, annál pontosabban tudjuk meghatározni a kialakuló hőmérséklet-eloszlást az áramkör felületén. A hiba a partíciók méretének növelésével együtt növekszik. A particionálás által bevitt hibát több partíció méretre is meghatároztam. A vizsgálatot egy speciálisan kialakított felületi topológián végeztem el. A topológiában az integrált áramkör felületén a disszipáló alakzatokat *kereszt* alakban helyeztem el a chip közepén. Az ilyen elhelyezéssel a különböző partíció méretek által okozott hibák könnyen meghatározhatóak. A disszipáló alakzatok oldalhosszainak méretét a teljes chip oldalhosszainak $\frac{1}{100}$ -ad részére méreteztem. Ez azt jelenti, hogy vízszintesen és függőlegesen is 100×100 darab disszipáló alakzatot lehetne elhelyezni a chipen. Mivel nullától különböző disszipációt csak a *kereszt* mintázat celláiban határoztam meg, így végeredményben a főátlóban és a mellékátlóban összesen $100 + 100 = 200$ darab cella található, melyek teljesítményét egyenként $1 \mu\text{W}$ -ban határoztam meg.

A partíciók méreteit a kerekítési hibák elkerülése miatt úgy határoztam meg, hogy a 100×100 -as eredeti felbontásba a partíciók maradék nélkül beférjenek. Ezek alapján a különböző particionálási felbontások 2×2 , 4×4 , 5×5 , 10×10 , 20×20 , 25×25 és 50×50 lettek. Az egyes partíciókra jutó teljesítmények a már ismertetett módon, a tartalmazott disszipáló alakzatok fogyasztásának összegeként állnak elő.

Az eredeti kereszt alakban elhelyezett 100×100 -as felbontással és a különböző méretű particionálással kapott hőmérsékleti térképeket összehasonlítottam, és meghatároztam a particionálás által bevitt hibát. A szimulált hőmérsékleti térképeket mutatja az 5.3. ábra (a számértékek $^{\circ}\text{C}$ -ban értendőek).

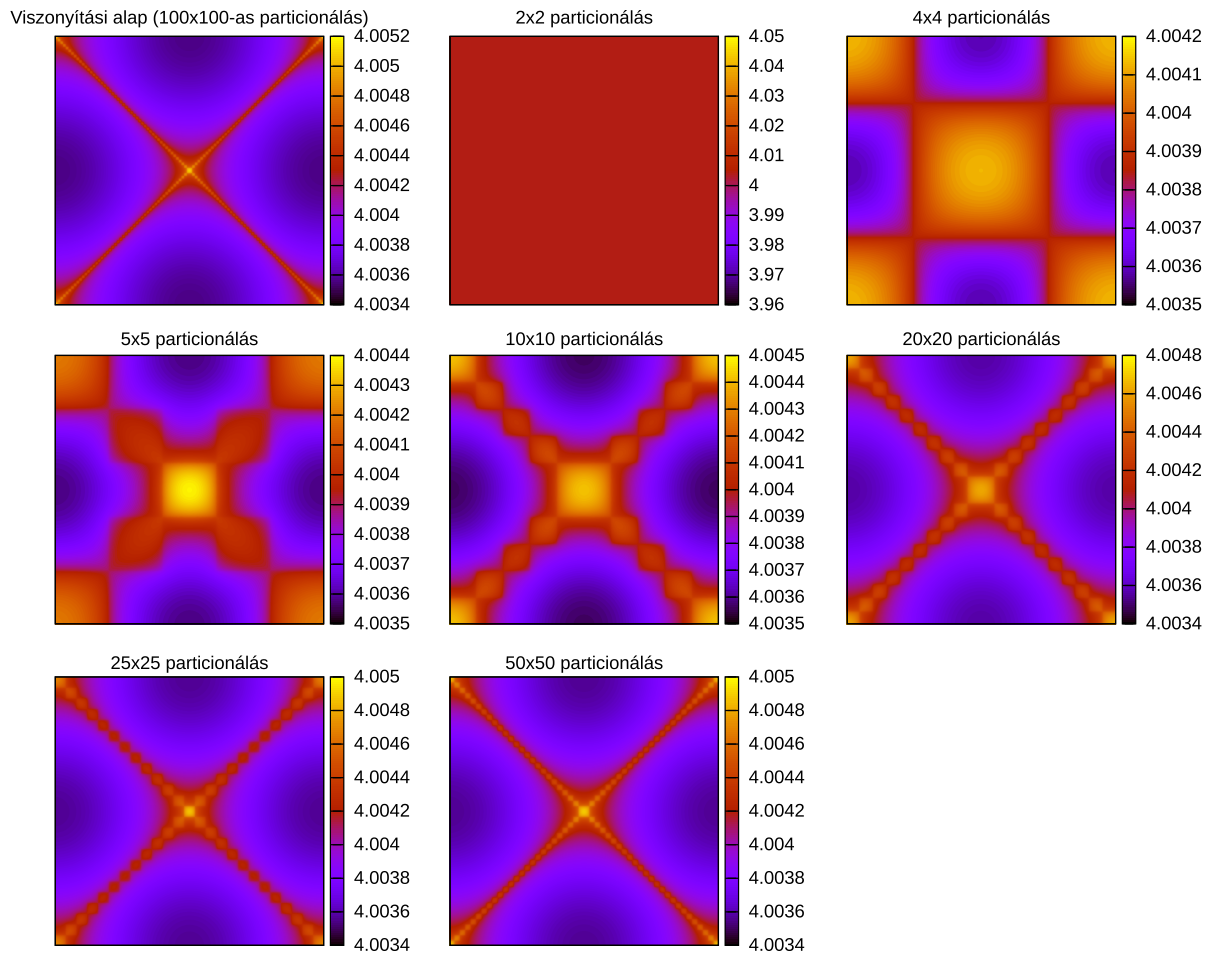
A különböző partíció méretekhez tartozó hőmérsékleti eloszlás térképeken megfigyelhető a középső kereszt alakú mintázat. A partíciók aktuális mérete is megfigyelhető az ábrákon.

Az eredeti és a particionált szimulációk közötti relatív eltérést szintén eloszlás-ábrák segítségével vizualizáltam. A hibák relatív százalékos eloszlását mutatja az 5.4. ábra különböző partíció méretekre. A relatív százalékos hibákat minden esetben az (5.2). képlettel számítottam:

$$\text{relatív hiba\%} = \frac{(E - P)}{E} \cdot 100 \%, \quad (5.2)$$

, ahol E a viszonyítási alap (etalon) és P a particionálással számított hőmérsékleti érték.

Megfigyelhető, hogy a particionálás felbontásának növelésével a hibák eloszlása és mértéke egyenletessé és egyre kisebbé válik, ahogy azt vártuk.



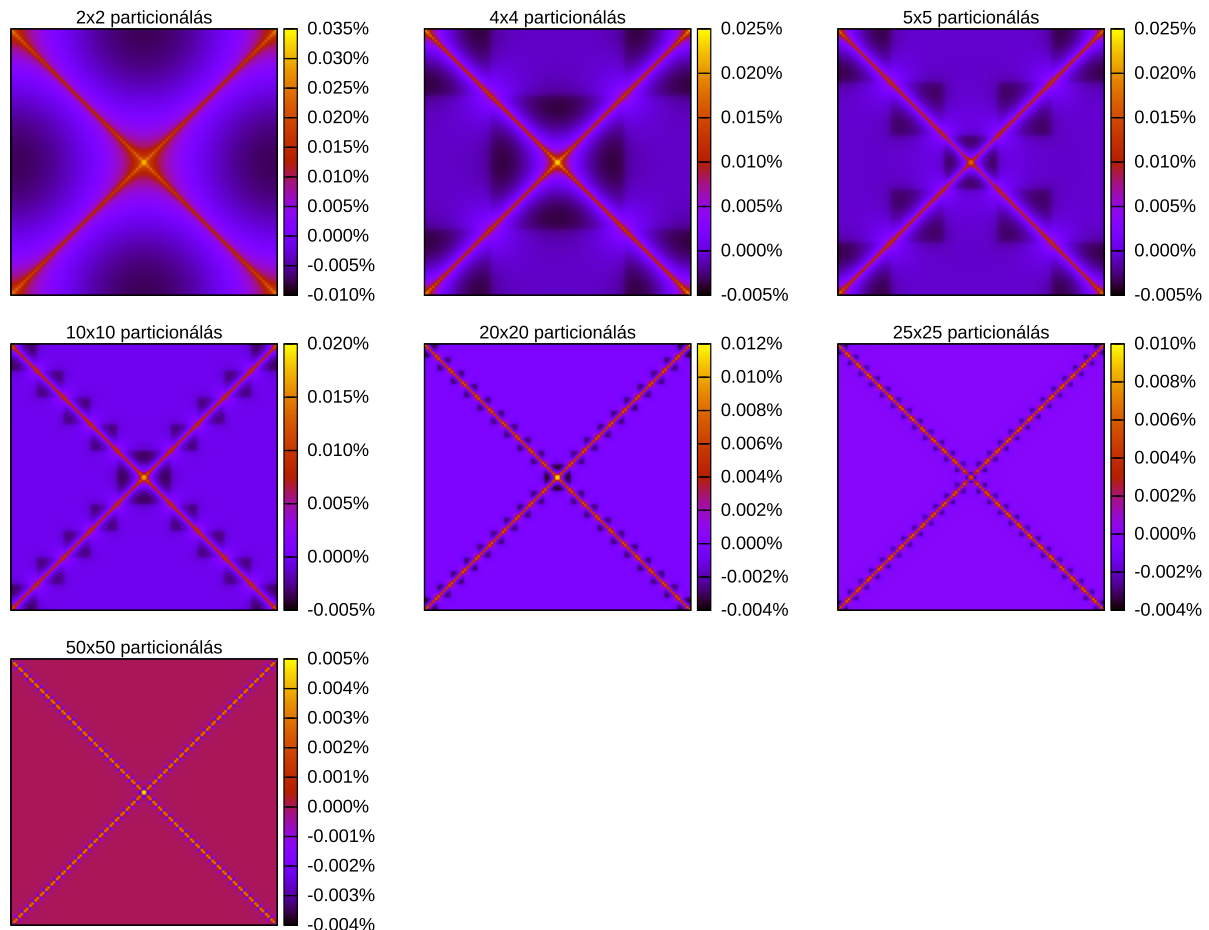
5.3. ábra. Hőmérséklet-eloszlás térkép különböző partíció méretek esetén

A hibák mértékét a particionálási felbontástól függően egy grafikonon is ábrázoltam, melyet az 5.5. ábra mutat. Látható, hogy a legnagyobb hibát a particionálási felbontás elvi alsó határánál (2×2) mérjük. A partíció méretek csökkenésével –tehát a felbontás növekedésével– a particionálás által bevitt hiba is csökken. Kvantitatíve értékelve a grafikont megállapítható, hogy a relatív hiba jóval 1 % alatt marad.

A különböző particionálási felbontások mindegyikéhez meg kell határozni az adott felbontáshoz tartozó termikus RC modellt is. Minél nagyobb a felbontás, tehát minél több partíció van, a modell annál bonyolultabb lesz. A termikus modell számítási ideje tehát növekszik. Összehasonlítottam az egyes felbontások esetében a termikus modell generálásához szükséges időket. A modell generáláshoz szükséges időket a különböző felbontásokra mutatja be az 5.6. ábra.

Megfigyelhető, hogy a felbontás növelésével a termikus modell elkészítéséhez szükséges idő exponenciálisan növekszik. Az idő adatok vizsgálatából megállapítottam és több ezer cellát tartalmazó integrált áramkörökön logi-termikus szimulációkkal kimutattam, hogy a bonyolult, sok cellából álló áramkörök logi-termikus szimulációjánál a 10×10 partícióból álló felbontás megfelelő a futási idő és a hőmérsékleti eloszlási térkép részletessége szempontjából.

A particionálási eljárás kidolgozásakor a 6.4. szakaszban részletesen bemutatott konkrét példa áramkörön is megvizsgáltam a particionálás által bevitt hibát. Két logi-termikus szimu-

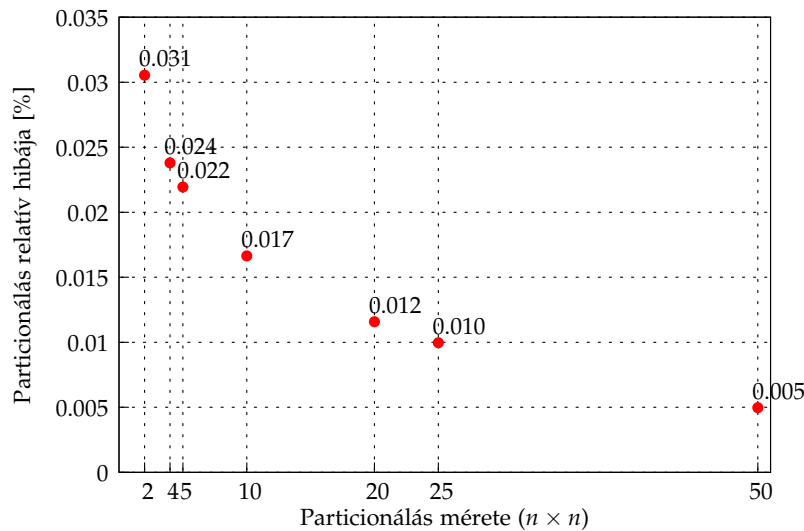


5.4. ábra. Relatív hiba-eloszlás térkép különböző partíció méretek esetén

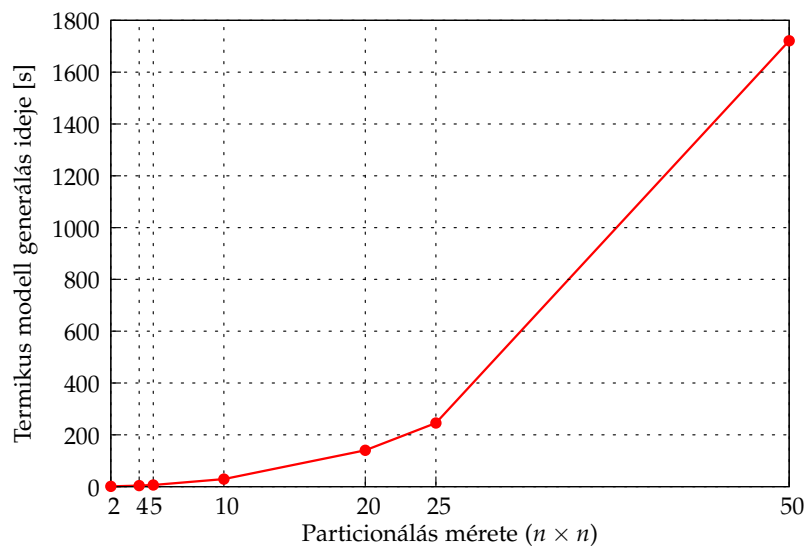
lációt hajtottam végre a demonstrációs áramkörön. Az egyik szimulációnál a particionálást kikapcsoltam, így a hőmérséklet értékek minden cellára külön-külön kiszámításra kerültek. A másik szimulációt a particionálás bekapcsolásával végeztem, a partíciók 10×10 -es rácsot alkottak a felületen. A két szimuláció hőmérsékleti tranziens görbéje közötti százalékos eltérést mutatja az 5.7. ábra.

Az 5.7. ábrán megfigyelhető, hogy a termikus tranziens görbék eltérése 0,2 % alatt marad a teljes szimuláció során. Egy ipari bonyolultságú digitális sztenderd cellás áramkör esetében, ami több ezer cellából épül fel a particionálás megfelelő gyorsítási lehetőségnek mutatkozott.

A particionálási eljárással tulajdonképpen több cella teljesítményét osztjuk el egy nagyobb felületen. A partíción belül az eredő teljesítményt a benne foglalt cellák együttes teljesítménye adja, ez a teljesítmény fogja fűteni az adott partíciót. Abban az esetben, ha egyes cellák teljesítménye jelentősen eltér a partíción belül található cellák átlagos teljesítményétől, akkor a particionálás által bevitt hiba ezekre a cellákra megnövekszik. Egy partícióra egy hőmérsékleti értéket kapunk a logi-termikus szimuláció egy időlépésében. Ha egy cella fogyasztása a partícióon belül például zérus, a partíció többi cellája viszont fogyaszt (és ezáltal hőt termel), akkor a szimulátor a partíció eredő hőmérsékletét helyettesíti be az inaktív cellába is. Ez akkor okozhat jelentős hibát, ha rossz hővezetési együtthatójú anyagon végezzük a szimulációt, mert ekkor az inaktív cella hővezetés szempontjából jobban el van szigetelve a környezetétől, tehát kevésbé melegszik fel a környező cellák aktivitásának hatására. A particionálási eljárás tehát olyan logi-



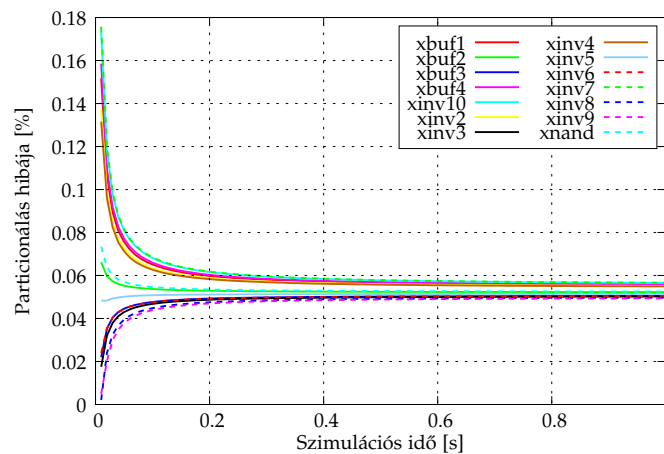
5.5. ábra. A particionálás hibája különböző partíció méretek esetén



5.6. ábra. A termikus modell generálásához szükséges idő különböző partíció méretek esetén

termikus szimulációknál alkalmazható kis hibával, ahol jó hővezetésű anyagon helyezkednek el a disszipáció forrásai illetve egy partíción belül a cellák fogyasztásai hasonlóak. Így például különböző funkciójú részegységeket egy processzoron belül érdemes külön partíciókba helyezni, mivel a kapcsolási aktivitás és így az eredő fogyasztás ezekben a részegységekben nagyon eltérhet egymástól. Egy konkrét példával élve, egy többmagos processzoron belül ajánlott az egyes magokat, aritmetikai-logikai és lebegőpontos egységeket, gyorsítótárat stb. külön partíciókba helyezni. Ezek a partíciók ezután tovább bonthatók kisebb részekre a felbontás növelése érdekében.

A particionálás mérete és a szimuláció sebessége között kompromisszumot kell kötni. Minél sűrűbben particionáljuk az áramkör felületét, annál több kompakt RC modellt kell meghatározni és kiszámítani minden szimulációs lépésben. Ez természetesen a szimulációs idő növekedését jelenti. A felbontás méretét az is korlátozza, hogy n partíció esetén n^2 kompakt mo-



5.7. ábra. A particionálás által bevitt hőmérsékleti tranziensek hibája a cellák vonatkozásában

dellt kell elállítani, ami például 100×100 -as particionálásnál 10000^2 RC modellt jelent. Ekkora felbontásnál a modellek generálása aránytalanul sok időt vesz igénybe. Összehasonlításképpen egy 1490 sztenderd cellát tartalmazó ipari áramkör szimulációjánál a termikus kompakt modellek generálásához szükséges időállandó-spektrumok meghatározása 76,5 másodpercig tart 10×10 -es particionálás esetén. Ha a particionálást 100×100 partícióra végeznénk el, akkor a számítási idő is négyzetesen növekedne, tehát közelítőleg 7650 másodperc lenne, ami 2 óra 8 percnél felel meg. Ez az idő csak a termikus időállandó-spektrumok generálására fordított idő. Az időállandó-spektrumokból még generálni kell az ekvivalens kompakt modelleket is. A szimuláció közben minden termikus időlépésben ki kell számítani a 10 000 partíció mindegyikére az eredő hőmérsékletet, ami szintén a szimulációs idő négyzetes növekedését okozza.

Az értekezésemben bemutatott szimulációs eredményeket (ha a particionálás be volt kapcsolva) 10×10 -es particionálással végeztem el. Az 5.7. ábrán láthatóan a particionálással bevitt hiba 0,2% alatt maradt. A durva felbontást utólagos grafikus feldolgozással (post-processing) tovább lehet finomítani bilineáris interpoláció segítségével. Az utófeldolgozással a szimuláció ideje nem változik, mégis nagyobb virtuális felbontás érhető el. Természetesen a partíciók közötti interpolációval csak a hőmérsékleti térkép vizualizációját lehet javítani, új hőmérsékleti információt nem nyerünk.

A logi-termikus szimulációk során megállapítottam, hogy a 10×10 -es, vagy 12×12 -es particionálási felbontás bonyolult ipari integrált áramkörök szimulációjánál is kielégítő felbontás-futási idő optimumot eredményez.

5.4. A szimuláció gyorsítása skálázással

1.3. altézis. Új gyorsítási módszert dolgoztam ki a szimuláció időbeli felbontásának skálázásával. Egy modern digitális rendszer termikus időállandói nagyságrendekkel nagyobbak az elektromos időállandóknál (pl. órajel periódusidő). A logikai szimuláció nagy órajel frekvencián hosszú ideig tarthat a termikus állandósult állapot megközelítéséig. A nagyfrekvenciás rendszer működési frekvenciáját

lecsökkentve és a kapcsolási tranziensekhez tartozó dinamikus fogyasztást arányosan növelve, a rendszer töredék idő alatt a termikus tranziens tetszőleges állapotába hozható. A kitűzött termikus állapotot elérve az eredeti működési frekvencián és fogyasztási értékekkel folytatható a szimuláció [C8].

Egy modern digitális integrált áramkör akár több gigahertzes frekvencián is működhet. A logikai szimulátorok az áramkörben az adott szimulációs időlépésben történt eseményeket (jelek változása) egy eseménysorba regisztrálják. Minden szimulációs időlépésben a logikai szimulátor feldolgozza az adott időlépéshez tartozó eseményeket, és az ezek által kiváltott újabb eseményeket előjegyzí valamely következő szimulációs időlépésbe az időzítési adatok alapján.

Egy bonyolult funkciót ellátó és magas órajelfrekvencián működő szinkron digitális áramkör egységnyi ideig tartó logikai szimulációja annál több szimulációs időt igényel, minél nagyobb az órajel frekvenciája, hiszen annál több logikai jelváltást kell vizsgálni és regisztrálni egy időlépésben.

A nagyfrekvenciás digitális alkalmazások gigahertzes órajeleinek periódusideje a nanoszekundumos nagyságrendbe esik. Egy integrált áramkör termikus időállandói jellemzően a milliszekundumos nagyságrendtől egészen a másodperces, perces időállandókig terjedhetnek. A termikus időállandó –ahogy az elektromos tartományban ismert– azt az időt jelenti, ami alatt a rendszer az állandósult állapotbeli hőmérséklet e -ad részét eléri. Ahhoz, hogy egy ilyen rendszer termikus viselkedését megvizsgáljuk jellemzően több másodpercig kell a megfigyelést végezni. A digitális órajel periódus és a termikus időállandók több nagyságrenddel eltérnek egymástól, így ha a gigahertzes órajelű rendszer logikai szimulációját több másodpercig akarjuk futtatni, akkor egy másodperc alatt több milliárd eseményt kell vizsgálnia és regisztrálni a logikai szimulátornak. A nagyfrekvenciás digitális áramkörök logikai szimulációját másodperces időig végezve a szimuláció időtartama rendkívül megnövekszik.

A logi-termikus szimuláció esetében azt szeretnénk meghatározni, hogy a rendszer működés közben milyen hőmérsékletekre melegszik fel és milyen hőmérsékleti eloszlás alakul ki a chip felszínén. Ezt csak úgy tudjuk szimulációkkal meghatározni, ha az áramkört a termikus időállandók nagyságrendjéig szimuláljuk.

Az általam kidolgozott eljárás segítségével a digitális rendszert alacsony órajel frekvencián meghajtva töredék szimulációs idő alatt az áramkör egy kívánt termikus tranziens állapotba hozható.

5.4.1. A módszer leírása

A digitális rendszer időzítési szimulációját a rendszert alkotó cellák késleltetéseinek ismeretében tudjuk elvégezni. A késleltetési adatokat egy sztenderd cellás áramkör tervezésekor általában a szintézer alkalmazás automatikusan állítja elő SDF (Standard Delay Format) típusú fájlokban. Ezek az adatbázisok tartalmazzák a cellák terjedési késleltetését. A megtervezett áramkör időzítései és működési frekvenciái is ezektől a késleltetés értékektől függenek.

Az általam kidolgozott logi-termikus szimulátor a logikai jelváltásokhoz rendelt energiákból határozza meg az áramkör fogyasztását és ennek segítségével számítja ki a hőmérsékleteket is. Ha minden cella késleltetési idejét egy f skálafaktorral megnöveljük, akkor az áramkör

működési frekvenciája a skálafaktorral arányosan lecsökken. Ez azt eredményezi, hogy a logikai szimuláció során egy időlépésben kevesebb jelváltás történik, így a szimuláció gyorsabb lesz. Egy példán demonstrálva az előbbieket, ha egy 1 GHz-es órajelfrekvencián működő áramkörben minden késleltetést tízszeresére növelünk, akkor a működési frekvencia 100 MHz-re csökken. A csökkentett frekvencián a szimuláció időtartama is lecsökken, hiszen egységnyi szimulációs időlépés alatt (pl. 1 ns) tízszer kevesebb jelváltás történik. Mivel minden időzítést a rendszerben arányosan növeltünk meg, a logikai működés és az időzítések viszonya egymáshoz képest nem változik. Úgy lehet elképzelni az így módosított rendszert, mintha egy gigahertzen működőképes áramkört egy nagyságrenddel kisebb frekvencián működtetnénk. Az áramkör nyilvánvalóan működőképes lesz, hiszen ha képes gigahertzes frekvencián működni, akkor nem okoz problémát a kisebb frekvencián történő működés sem.

Célunk a működési frekvencia csökkentésével, hogy egy szimulációs időlépés alatt kevesebb jelváltással kelljen számolni. Mivel a termikus szimulációban a bemenő teljesítmények a cellák adott idő alatti kapcsolási aktivitásából származnak, így a kisebb frekvencián az áramkör fogyasztása is kisebb lesz, tehát nem melegszik fel annyira, mint a névleges működési frekvencián.

Ha a fogyasztási adatbázisban (Liberty) minden cella kapcsolási tranzienshez tartozó energiáját az f skálafaktorral megnöveljük, akkor annak ellenére, hogy kisebb működési frekvencián időlépésenként kevesebb aktivitást detektálunk, egy kapcsolat az f faktorial megnövelt energiát fog fogyasztani. Így végeredményben a fűtőteliesség ugyanannyi marad. Ezt az összefüggést mutatja (5.3).

A terhelő kapacitás feltöltésére vagy kisütésére fordított energia meghatározható a kapacitás nagyságából és a tápfeszültségből.

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

és

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

ahol U a tápfeszültség, C a terhelő kapacitás és ν a kapcsolási frekvencia. Ha $\nu' = \frac{1}{10}\nu$ és $E' = 10E$ helyettesítéssel elünk, akkor

$$P' = E' \cdot \nu' = 10E \cdot \frac{1}{10}\nu = E \cdot \nu, \quad (5.3)$$

így

$$P' = P, \quad (5.4)$$

tehát a fűtőteliesség nem változik. A módszerrel megvalósított logi-termikus szimuláció a kapcsolási frekvencia csökkentésével és a fogyasztás arányos növelésével végeredményben rövidebb szimulációs időtartamot eredményez, de a névleges működéssel azonos termikus állapotba kerül a rendszer.

A késleltetési idők növelésének a termikus szimulációs időablak szélessége szab határt. A késleltetések megnövelésével valamennyi hibát elkövethetünk a kapcsolási aktivitás mintavételezésében, mivel a termikus szimulációs időablak méretét nem skálázzuk. A termikus

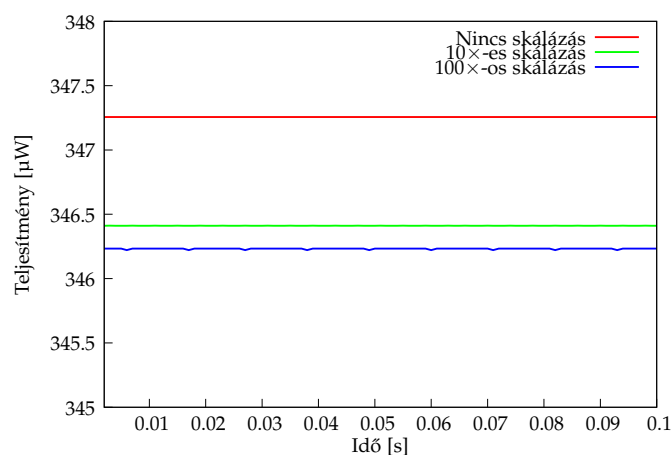
időablak mérete jellemzően nagyságrendekkel nagyobb még a felskálázott késleltetéseknél is, így például egy 10 ms-os termikus átlagoló időablak mellett a 100 MHz-es jelek periódusideje is nagyságrendekkel kisebb (10 ns). Az előbbi példában egy 1 ms-os termikus átlagoló időablakba 100 MHz-es jelek esetén is 100 000 jelváltás esik. A késleltetési idők felskálázását addig végezhetjük, amíg minden késleltetési idő a termikus átlagoló időablak szélességén belül marad. Ez azért fontos kritérium, mert a termikus időablak szélességénél nagyobb késleltetések esetében a kapcsolási tranziensek áttolódnak a következő termikus időablakok valamelyikébe, így az adott időablakban bekövetkezett fogyasztást csak egy későbbi időablakban vesszük figyelembe. Ezáltal a termikus szimuláció hamis eredményt fog szolgáltatni. A módszer kidolgozása és vizsgálata során arra a megállapításra jutottam, hogy gigahertzes működési frekvenciájú áramkörök vizsgálatánál az $f = 10$, vagy $f = 100$ skálázás már nagyságrendekkel gyorsabb szimulációkat tesz lehetővé a pontosság elhanyagolható romlásával.

Az értekezésemben ismertetett gyorsítási eljárással végeztem a 6. fejezetben bemutatott szimulációk nagy részét. Szimulációkkal kimutattam a skálázás által elkövetett hiba mértékét.

A skálázás hibáját legjobban a termikus időablakban átlagolt teljesítményeken lehet kimutatni. Az eredő fűtőtelsítményeknek a skálázás után is a névleges értékhez kell közel esniük. Ha a teljesítmények eltérése a névleges és a skálázott értékek között elhanyagolható, akkor az eredő hőmérsékletek eltérése is elhanyagolható lesz.

A logi-termikus szimulációkat egy egyszerű gyűrűs oszcillátor áramkörön végeztem el. Az áramkörben egy NAND kapu és tíz inverter van sorba kapcsolva. Az oszcillátor névleges kimeneti frekvenciája skálázás nélkül ($f = 1$) 1,52 GHz. A szimulációkat 100 ms szimulációs ideig futtattam. A termikus szimulációs időablak 1 ms széles volt. A szimulációs áramkör kiegészített változatát részletesen a 6.4. szakasz tárgyalja.

A szimulációkat több skálázási faktorial futtattam. A névleges szimuláció esetében az f skálafaktor 1, tehát a névleges eset jelenti a szimulációt teljes működési frekvencián, skálázás nélkül. A logi-termikus szimulációkat elvégeztem $f = 10$ és $f = 100$ skálafaktorokkal is. Az eredő számított fogyasztásokat az `inv5` cellára hasonlítottam össze, mely a gyűrűs oszcillátor ötödik invertere. Az inverter cella fogyasztását mutatja az 5.8. ábra $f = 1, 10, 100$ skálafaktor értékekre.

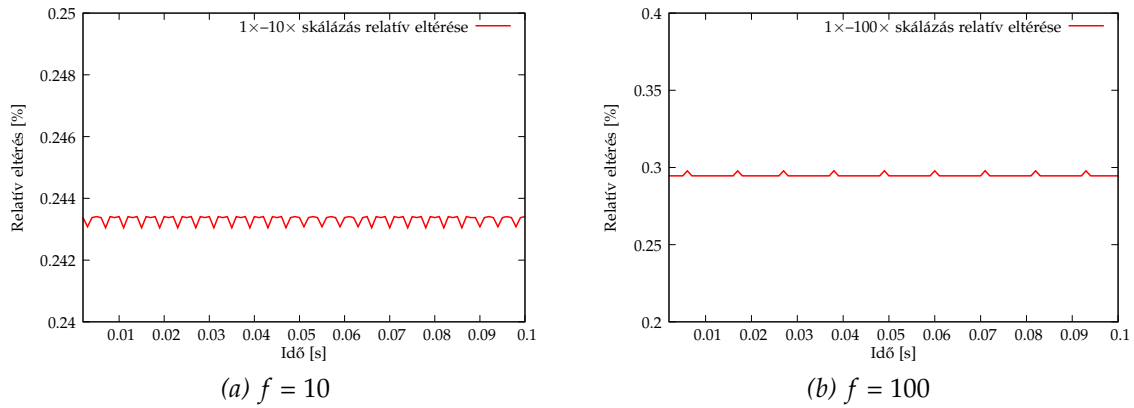


5.8. ábra. Az `inv5` cella eredő fogyasztásának függése a skálázástól

Az 5.8. ábrán megfigyelhető, hogy a skálázás hatására a névleges fogyasztáshoz képest

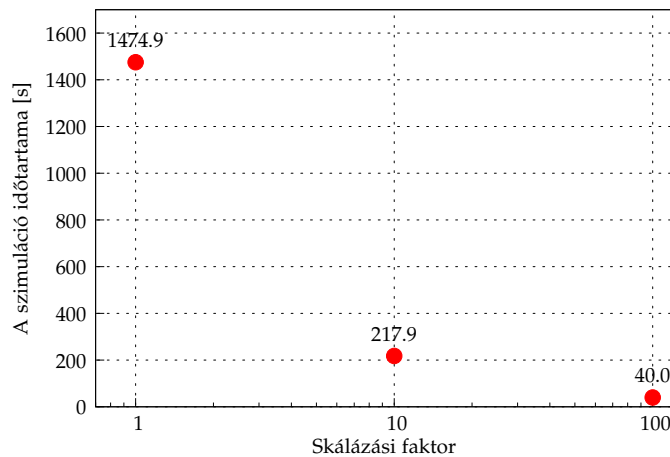
az $f = 10$ és $f = 100$ -szoros skálázás a fűtőteljesítményben $1\ \mu\text{W}$ körüli hibát eredményez. Ez 0,23 % relatív hibát jelent a névleges értékhez viszonyítva.

A névleges fogyasztási értékhez viszonyított relatív eltérések görbéjét az 5.9. ábra mutatja be $f = 10$ és $f = 100$ skálafaktor esetére. Megfigyelhető, hogy $f = 100$ -as skálafaktornál is az eltérés 0,3 % körüli érték alatt marad.



5.9. ábra. Az *inv5* cella fogyasztásának relatív eltérése

A szimulációs időtartamokat a CellTherm alkalmazásba beépített időmérővel határoztam meg. A szimulátor minden termikus számítás alkalmával kiírja a sztenderd kimenetre a számítás kezdete óta eltelt időt. A szimulációk időtartamát a különböző skálafaktorokra az 5.10. ábra mutatja.



5.10. ábra. Szimulációs idők különböző skálafaktoroknál

A névleges működési frekvencián a logi-termikus szimuláció 1475 másodpercig (~ 25 perc) tartott 100 ms-ig. A 100 ms-os szimulációs idő jóval kevesebb, mint a domináns termikus időállandók, így a termikus állandósult állapotot megközelíteni csak több másodpercig tartó szimulációval tudjuk. Ez a névleges működési frekvencián szimulálva napokat vehet igénybe.

Az $f = 10$ -szoros skálafaktoral a számítás időtartama 218 másodpercre csökkent. Ez a névleges számítási idő csaknem $\frac{1}{7}$ -ed része. Az $f = 100$ -szoros skálázással a szimuláció ideje 100 ms-ig 40 másodperc lett. Ez az eredeti időtartam közel $\frac{1}{37}$ -ed része, tehát a gyorsulás 37-

szerez. Ilyen gyorsítással már a több másodpercig tartó logi-termikus szimuláció is elfogadható idő alatt végrehajtható.

Az általam bemutatott és alkalmazott módszer segítségével az áramkör rövid szimulációs idő alatt a termikus tranziens tetszőleges állapotába hozható. A meghatározott termikus állapotot elérve a skálázást vissza lehet állítani a névleges értékre, így az áramkör szimulációja a meghatározott hőmérsékleti tartományban történik tovább. A CellTherm alkalmazás, mivel a teljes áramkör időzítéseit és fogyasztási adatait ismeri, és ezeket képes megváltoztatni a szimuláció közben is, a skálázási faktort bármelyik időlépésben módosíthatja. A rugalmasan változtatható skálázással a szimuláció futtatása közben bármikor átállíthatók az időzítések és a fogyasztási adatok a gyorsítás érdekében. A kívánt termikus állapotot elérve a CellTherm szimulátor a skálafaktort névleges értékre állíthatja, és a szimuláció ezen a hőmérsékleten futtat tovább a névleges működési sebességgel és fogyasztási értékekkel. A bemutatott módszert sikerrel alkalmaztam a 6. fejezetben ismertetett hőmérsékletfüggő időzítésekkel foglalkozó logi-termikus szimulációknál.

6. fejezet

Hőmérsékletfüggő időzítések

2. tézis. *Eljárást dolgoztam ki sztenderd cellás integrált áramkörök hőmérsékletfüggő időzítéseinek szimuláció közben történő nyomon követésére és az esetleges hőmérsékletfüggő funkcionális meghibásodások felderítésére. A tervezőrendszerekben használt logikai szimulátorok nem veszik figyelembe az áramkör működése közben kialakuló hőmérsékleti viszonyokat. Az általam kidolgozott módszer segítségével az áramkör felületén kialakuló hőmérséklet eloszlás jellege miatt megváltozott időzítési viszonyok meghatározhatóak. A módszer működése során a termikus szimulációs eredményeket visszacsatolom a logikai szimulációba az egyes cellák hőmérsékletfüggő késleltetéseként. A módszer segítségével a cellák időzítési viszonyainak változása nyomon követhető [J2, J4],[C6, C7].*

6.1. A hőmérsékletfüggő időzítések számítása

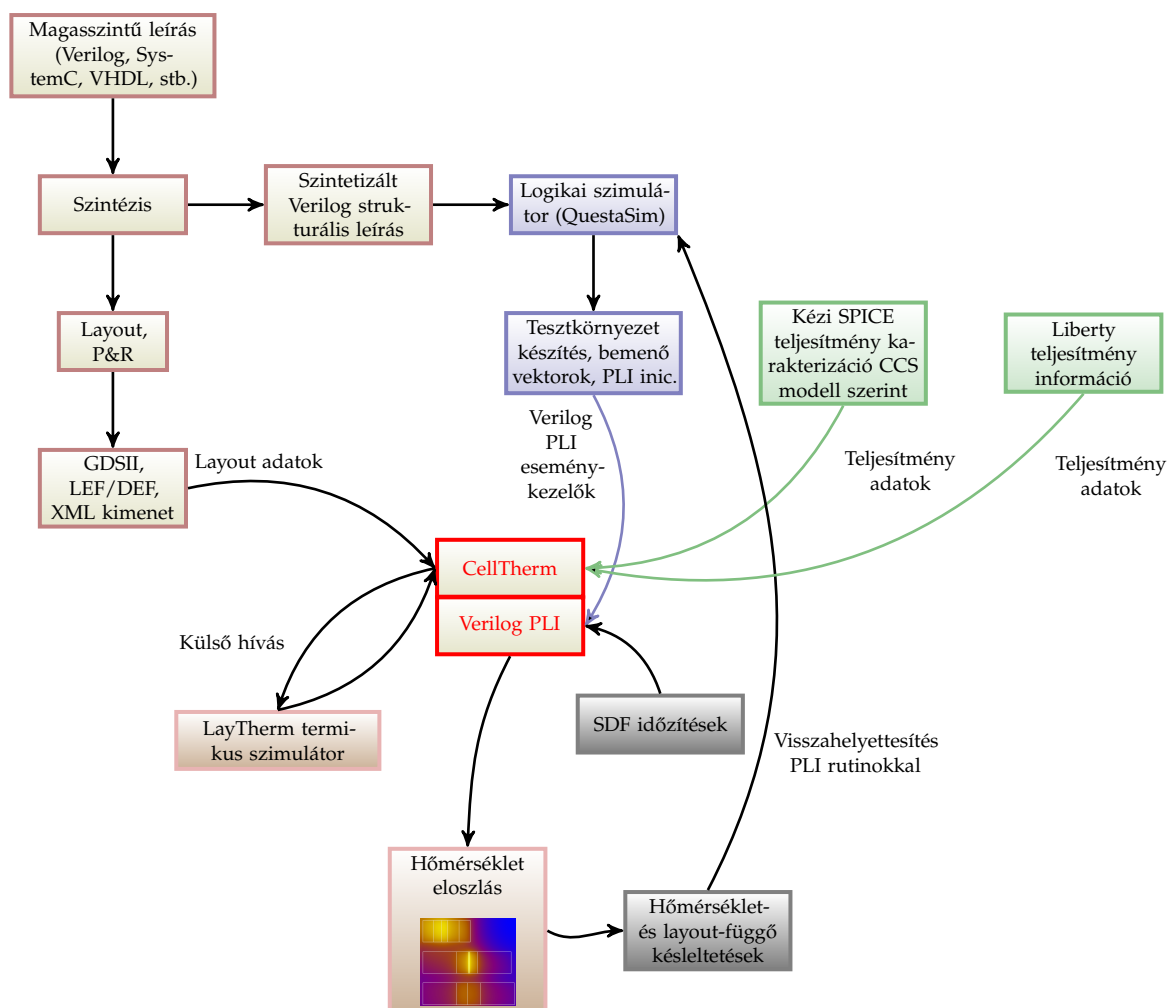
A logi-termikus szimuláció vázlatát a visszacsatolt hőmérsékletfüggő időzítésekkel kiegészítve mutatja a 6.1. ábra. A termikus szimulátor által előállított hőmérsékleti eloszlást hőmérsékletfüggő késleltetések formájában lehet a logikai szimulátorba visszacsatolni. A hőmérséklet-változás hatására kialakult késleltetés-változások a Verilog PLI interfész segítségével csatolhatóak a logikai szimulációba. A logi-termikus szimuláció a hőmérsékletfüggő késleltetések szimuláció közben történő folyamatos frissítésével válik teljessé.

A sztenderd cellás tervezésnél ideális esetben a cellák minden adata rendelkezésre áll a tervezői eszközkészletben (Design Kit). Az eszközkészlet tartalmazza mindazt az adathalmazt, melynek segítségével a logikai szimulátor, a szintézer, az elhelyező és huzalozó algoritmus (Place&Route), a DRC¹/LVS² ellenőrző működik. Ezek az adatok több fájlban foglalnak helyet:

- SPICE/ELDO/Spectre modell fájlok, tranzisztor modell paraméterek analóg szimulációkhoz,
- DRC/LVS szabályok, a fizikai elhelyezés és az összeköttetések vizsgálatára,
- LEF fájlok, a cellák layout prototípusai,

¹ Design Rule Check, tervezési szabály ellenőrzés.

² Layout versus Schematic, a fizikai terv és a kapcsolási rajz összevetése.



6.1. ábra. A logi-termikus szimuláció visszacsatolt hőmérsékletfüggő késleltetésekkel

- Liberty /TLF fájlok, időzítések, zajparaméterek, fogyasztási adatok adatbázisa,
- Verilog/VHDL fájlok, a cellák funkcionális viselkedési leírása (logikai szimulációhoz),
- SDF fájlok, a szintézer által generált időzítési adatok.

Ezek az adatbázisok a hőmérsékleti viszonyokat csak korlátozottan veszik figyelembe. A cellakönyvtár celláinak időzítéseit tartalmazó *Liberty*[45] formátum az időzítési, zaj és teljesítmény adatokat egy rögzített hőmérsékletre tárolják el. Különböző szélsőértékekre (corner) különböző Liberty fájlokat készítenek a karakterizációs eljárásokkal. A szintézer szoftverek többek között a Liberty adatbázisokból származtatják az elhelyezés és vezetékezés után azokat az SDF[46] időzítési adatokat, melyek általában három jellemző paraméter szélsőértékeire vannak megadva. Ezek a paraméterek a *tápfeszültség*, a *gyártási szórás* és a *hőmérséklet*. A három paraméter értékeit a gyártó *minimális*, *tipikus* és *maximális* értékekkel szokta megadni. Ezek a szélsőértékek reprezentálják a gyártás bizonytalanságait és a felhasználási környezet tulajdonságait (ilyen például a környezeti hőmérséklet). Az említett effektusokkal több szakirodalom is foglalkozik különböző áramkörök kapcsán[47, 48, 49].

A szintézer program által előállított SDF fájl az előbb említett *technológiai szélsőértékekre* (corner) adja meg minden sztenderd cella időzítési követelményeit és késleltetéseit.

Annak érdekében, hogy az áramkör időzítéseit és a cellák késleltetéseit hőmérsékletfüggően tudjuk figyelembe venni, szükség van a két mennyiség kapcsolatának megállapítására[50, 51]. Egy egyszerű logikai kapu esetében (például 2 bemenetű NAND kapu) szükség van a két bemenet és a kimenet közötti késleltetés (propagation delay) meghatározására a hőmérséklet függvényében. Sorrendi hálózatok esetében további hőmérsékletfüggő kritériumokat is kell vizsgálni. Egy D-flip-flop esetében például szükség van az *adat előkészítési idő* (setup time), az *adat tartási idő* (hold time), a *minimális pulzusszélesség* (minimum pulse width), stb. hőmérséklet-függésének számítására is. Az ilyen jellegű hőmérsékletfüggések egyik gyártó által biztosított tervezői eszközkészletben sem állnak *expliciten* rendelkezésre.

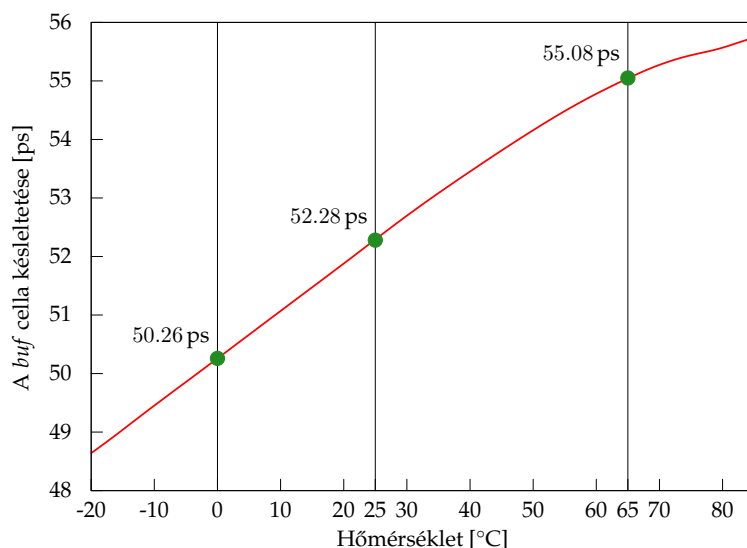
Munkámban egy egyszerű módszerrel közelítem az egyes könyvtári cellák hőmérsékletfüggő késleltetéseit. Az SDF adatbázisból ismertek a különböző hőmérsékleti szélsőértékekhez (minimális, tipikus, maximális) tartozó késleltetések minden cella minden portjára vonatkoztatva. A cellák portjainak hőmérséklet-késleltetés függvényeit lineáris interpolációval határozom meg a termikus szélsőértékek ismeretében [J4],[C7]. A módszer segítségével becslést lehet adni a cellák késleltetésének hőmérsékletfüggésére, ami a logi-termikus szimulációk során közvetlenül felhasználható.

Amennyiben mégis pontos hőmérséklet-késleltetés függvényekre van szükség, a cellakönyvtár celláin analóg SPICE szimulációkat futtatva ezek a görbék pontosan is meghatározhatóak. Ebben az esetben nagy probléma, hogy az analóg szimuláció futtatásához szükség van a sztenderd cellák tranzisztor szintű leírására, amit a gyártók szinte minden esetben titokban tartanak, hiszen a tranzisztor szintű leírás és a cellák layout-jának kiadásával a saját szellemi tulajdonukat veszélyeztetnék. Az előbbieket miatt a cellakönyvtár fizikai megvalósítását tartalmazó LEF/DEF[41] formátumú fájlokat is úgy állítják elő, hogy azok csak a vezetékezési algoritmus szempontjából lényeges akadályokat (obstacle) és a portok helyeit definiálják, a konkrét layout nem látható. Amikor a megtervezett integrált áramkör gyártásra kerül, a gyártó automatikusan behelyettesíti a tervben szereplő cellák helyére a tényleges, tranzisztor szintű layout-ot, és ez alapján készíti el a maszkokat.

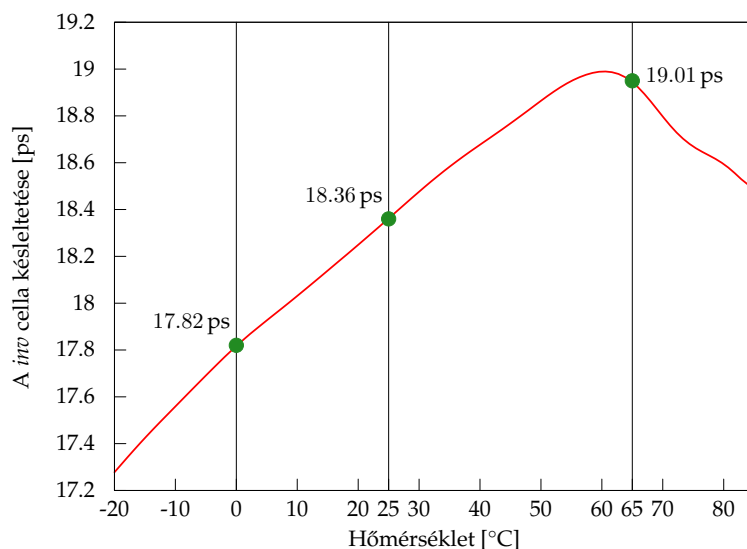
Munkám során több tervezői eszközkészletet (design kit) is használtam, melyek közül egyet a Mentor Graphics cég szabadon elérhetővé tett demonstrációs célokra (ADK, ASIC Design Kit, TSMC 0,35 μm -es technológia). Ebben a készletben a cellák tranzisztor szintű leírása és layout-ja is rendelkezésre állt, így ezen a demonstrációs könyvtáron tudtam analóg szimulációkat is végezni. Az analóg szimuláció segítségével meg tudtam határozni bármelyik cella tényleges késleltetés-változását a hőmérséklet függvényében.

A 6.2–6.4. ábrákon látható három sztenderd cella hőmérséklet-késleltetés függvénye. Az analóg szimulációban a cellák kimenetét egy-egy inverter terheli, mely szintén a cellakönyvtár része, így a késleltetés értékek a valóságnak megfelelőek lesznek. Ha nem teszünk terhelő cellát a kimenetekre, akkor terhelés hiányában a kimenetek lebegni fognak és hamis késleltetés értékeket mérhetünk. A késleltetések jelentősen függenek a kimeneti terheléstől, hiszen a mérendő cella meghajtó képessége állandó, egy maximális áramnál többel nem tudja meghajtani terhelését. Ez a maximális áram adott idő alatt tölti fel (vagy süti ki) a terhelés kapacitását. A töltés/kisütés ideje tehát a terhelő kapacitás nagyságától függ.

A kapukésleltetést mérő szimulációk során a hőmérsékletet $-20\text{ }^{\circ}\text{C}$ és $85\text{ }^{\circ}\text{C}$ tartományban, fokenkénti pásztázással hajtottam végre[J2].



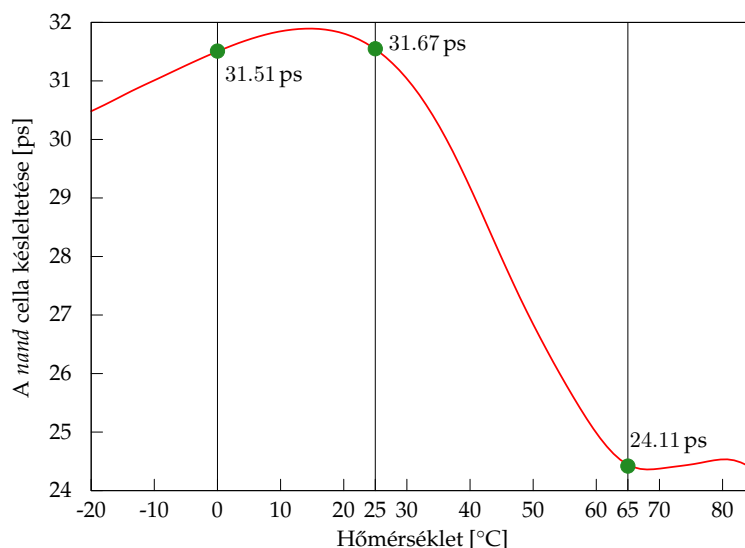
6.2. ábra. A BUF cella késleltetés-hőmérséklet függvénye



6.3. ábra. Az INV cella késleltetés-hőmérséklet függvénye

A könyvtári cellák többségénél a hőmérséklet-késleltetés görbe monoton növekszik a hőmérséklettel. A 6.3. és 6.4. ábrákon azonban megfigyelhető, hogy bizonyos hőmérsékleti tartományokban a késleltetés csökken a növekvő hőmérséklettel. Ezzel a jelenséggel több irodalmi forrás is foglalkozik részletesen[52, 53, 54, 55].

Ahogy a technológia fejlődik és a csíkszélesség nanométeres tartományban jár (< 90 nm), az eszközök teljesítménye már olyan tényezőktől is függ, melyeket eddig másodlagos hatásként vettünk figyelembe. Az egyik ilyen hatás a *fordított hőmérsékletfüggés*, (ITD, *Inverse Temperature Dependence*)[52, 53, 54, 55]. Amikor egy cella alacsony feszültségen működik, a terjedési késleltetés csökkenhet a hőmérséklet növekedésével. A jelenség hátterében a küszöbfeszültség (V_{th}) hőmérsékletfüggése áll. Ahogy a tápfeszültség csökken, a tranzisztor *gate-source* feszültsége és a küszöbfeszültség különbségének abszolút értéke ($|V_{gs} - V_{th}|$) is csökken. A kisebb



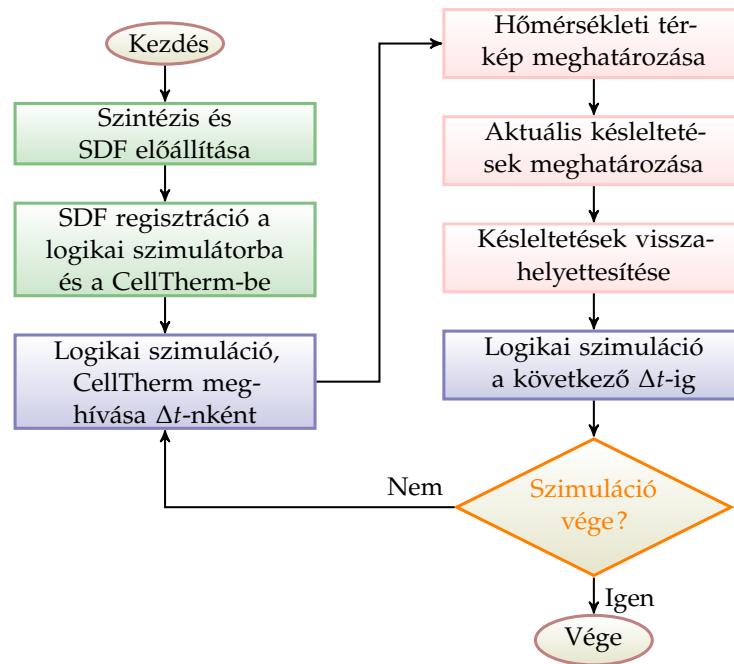
6.4. ábra. A NAND cella késleltetés-hőmérséklet függvénye

$|V_{gs} - V_{th}|$ miatt a telítési áram sokkal érzékenyebb lesz a V_{th} változásaira, ami a hőmérséklet növekedésével csökken. A kisebb V_{th} miatt kisebb feszültségen nagyobb áram fog folyni, így a billenés is gyorsabb lesz. Másrészt a késleltetés arányos a töltéshordozó-mobilitással, ami a hőmérséklet növekedésével csökken. Emiatt az eszköz gyorsasága az elektron-mobilitástól és a V_{th} értékétől együttesen függ. Hagyományosan a hőmérsékleti szélsőértékeket (cornereket) egy technológiához úgy állapították meg, hogy a magas hőmérséklet jelentette a működés szempontjából a *legrosszabb esetet* (*worst case*), a legalacsonyabb hőmérséklet pedig a *legjobb esetet* (*best case*). Az említett hatás (ITD) miatt azonban ez már nem garantálható. Ez nagy problémát jelent az időzítési szélsőértékek meghatározásánál és a cellakönyvtárak időzítési karakterizálásánál. Lehetséges, hogy a *worst case* forgatókönyv egy köztes hőmérséklet értéken jelentkezik, aminek a felderítése nehéz feladat lehet[56].

6.2. A hőmérsékletfüggő időzítési szimulációs eljárás lépései

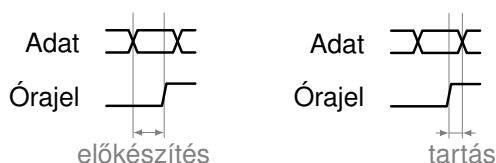
A hőmérsékletfüggő időzítéseket figyelembe vevő logi-termikus szimulációs eljáráshoz szükséges lépéseket és a szimuláció menetét a következő felsorolásban mutatom be részletesen. Az eljárás lépései a 6.5. ábrán követhetők nyomon.

1. A hőmérsékletfüggő késleltetéseket alapvetően a szintézer által elkészített, időzítési adatokat leíró SDF fájl segítségével lehet figyelembe venni. A szimulálandó áramkör viselkedési leírását tehát sztenderd cellás strukturális leírássá kell átalakítani. Ezt a feladatot automatizálja a szintézer.
2. A logi-termikus szimuláció elején az időzítési SDF adatbázist a logikai szimulátorban regisztrálni kell. Erre a logikai szimulátorok és a hardver leíró nyelvek külön utasításokat tartalmaznak. Az SDF adatokra azért van szüksége a logikai szimulátornak, mert így képes pontos időzítési szimulációt végezni.



6.5. ábra. A hőmérsékletfüggő időzítési szimulációs eljárás lépései

3. A logi-termikus szimulátorban (CellTherm) is regisztrálni kell az SDF fájlt. A logi-termikus szimulátor feldolgozza és egy belső adatbázisában tárolja az egyes kapuk időzítéseit, így képes interpolálni a különböző hőmérséklet-késleltetés értékek között.
4. A logi-termikus szimuláció indítása után a meghatározott termikus időlépésenként (Δt) a logikai szimulátor átadja a CellTherm alkalmazásnak az abban a termikus időlépésben történt kapcsolási aktivitás adatokat.
5. A CellTherm a kapcsolási adatok és a hozzájuk tartozó teljesítmények alapján a termikus szimulátor segítségével meghatározza a kialakuló hőmérsékleteket a chipen.
6. Ezekkel a hőmérsékletekkel a hőmérséklet-késleltetés adatbázisból minden cellára meghatározza az éppen aktuális késleltetési értékeket.
7. A meghatározott késleltetési értékeket még a jelenlegi időlépésben visszahelyettesíti minden cellába. A késleltetések menet közben történő megváltoztatására a Verilog PLI programozói interfész beépített függvényeket biztosít.
8. A logikai szimulátor a megváltozott késleltetésekkel lép a következő logikai szimulációs lépésbe. A megváltoztatott késleltetések miatt a cellák aktivitása automatikusan a megfelelő szimulációs időpillanatra lesz előjegyezve.
9. A logikai szimulátor ezután folytatja a szimulációt a következő termikus időlépésig, ahol a 4. lépéstől folytatódik a folyamat, amíg a szimuláció véget nem ér.



6.6. ábra. Adat előkészítési és tartási idők

6.3. Időzítési problémák detektálása

A logikai szimulátor a szimuláció teljes ideje alatt figyelemmel kíséri az időzítési követelményeket. Ilyen időzítési követelmények lehetnek a cellák maximális késleltetései, az órajel kényszerek (slack), a sorrendi cellák adat előkészítési- és adat tartási idejei (setup- és hold time), a különböző logikai jelszintekhez tartozó minimális pulzus szélesség. Ezek a követelmények az időzítéseket leíró SDF fájlban vannak eltárolva és a szintézer alkalmazás határozza meg őket a cellakönyvtár adatai, az elhelyezés és a huzalozás alapján. Az általam kidolgozott logi-termikus szimulátorban a logikai szimulátor szerepét jellemzően egy iparban használt, szabványos szimulátor tölti be, melyben az időzítési kényszerek figyelemmel kísérése magas szinten van implementálva. Ebből következik, hogy ezek a logikai szimulátorok értesíteni tudják a felhasználót a szimuláció közben, ha valamilyen időzítési követelmény nem teljesül.

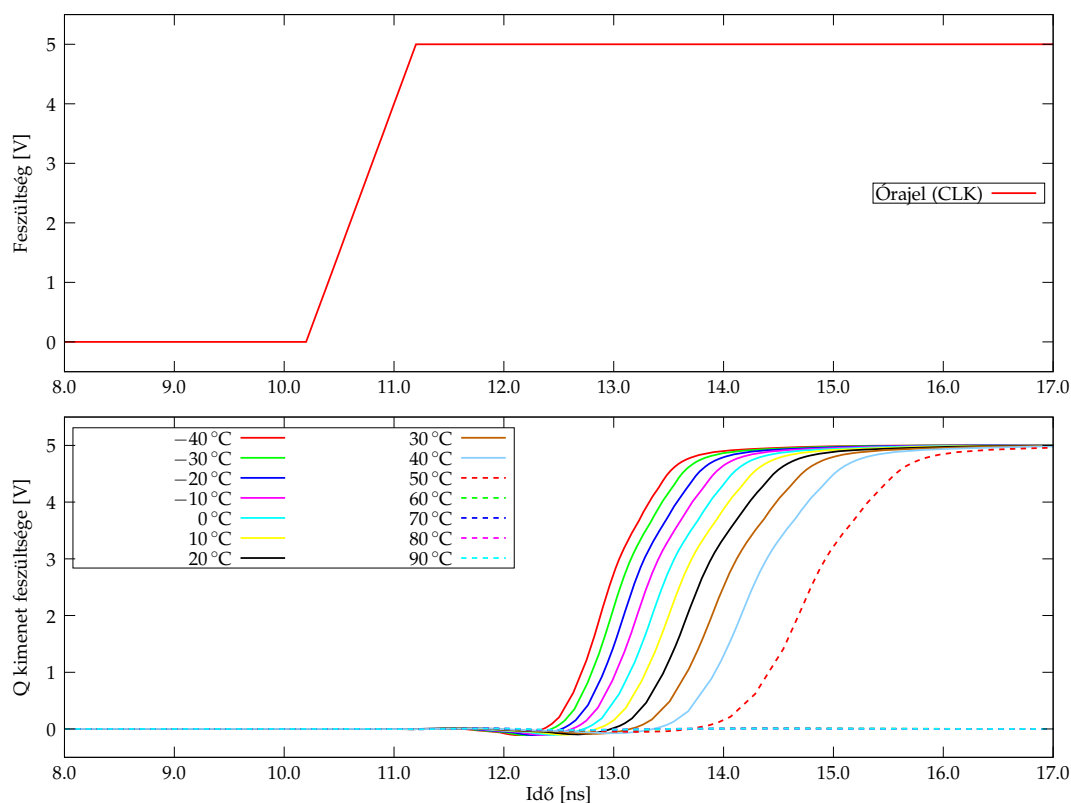
Az időzítési hibák detektálását egy mintapéldán mutatom be[J4],[C6]. A példában egy D flip-flop cella adat-előkészítési és adat-tartási időzítésének nem-teljesülését demonstrálom. A logi-termikus szimulációk során a változó hőmérsékletek miatt eltolódó késleltetések okozhatnak ilyen jellegű időzítési hibákat, amiket a logikai szimulátor kiszűr és jelez. A bemutatott példában a D flip-flop adat bemenetét úgy vezérlem, hogy szándékosan nem tartom be az adat előkészítési időt, így megfigyelhető, hogy a szimulátor az időzítési hibát jelzi és megállítja a szimulációt.

A D flip-flop cella tranzisztor szintű modelljén elvégeztem az adat-előkészítési időt (setup time) meghatározó szimulációt. Az adat-előkészítési- és adat-tartási idő mérése illetve szimulációja iteratív úton történhet, hiszen definíció szerint az adat előkészítési idő az az idő, amennyivel előbb az adat bemenetnek már stabilnak kell lennie az órajel felfutó éle előtt. Az adat tartási idő definíció szerint az az idő, amennyivel az adatnak még stabilnak kell lennie a bemeneten az órajel felfutó éle után. Az adat-előkészítési és tartási idő definícióját mutatja a 6.6. ábra.

Ha a cella hőmérsékletét széles tartományban változtatjuk, az órajel bemenet és a kimenet közötti késleltetés is változni fog. Lesz egy olyan kritikus hőmérséklet érték, ahol a késleltetés pontosan olyan mértékű lesz, hogy az adat bemenetet nem lehet időben mintavételezni az órajel felfutó élekor, így az adat bemeneti jelváltozás nem jut ki a kimenetre. Ez a kritikus késleltetés a minimális adat előkészítési idő.

Analóg SPICE szimulációkkal meghatároztam a D flip-flop órajel-kimenet késleltetését úgy, hogy a környezeti hőmérsékletet -40 – $+90$ °C tartományban pásztáztam. Az adat bemenet változását olyan közel helyeztem az órajel felfutó (mintavevő) éléhez, hogy megfigyelhető legyen bizonyos hőmérséklet felett az adat-előkészítési idő (setup-time) nem-teljesülése. A 6.7. ábrán látható, hogy 50 °C fölött az adat-előkészítési időzítés nem teljesül, a kimenet nem változik meg.

Ugyanebből a szimulációból meghatároztam a D flip-flop késleltetését a hőmérséklet függvé-



6.7. ábra. D flip-flop késleltetés tranziensek

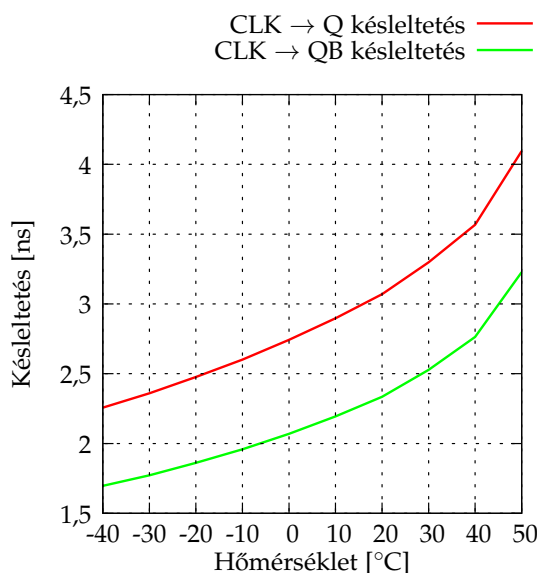
nyében, ez látható a 6.8. ábrán. Itt is megfigyelhető, hogy a görbe csak a -40 – $+50$ tartományban értelmezhető, hiszen 50 °C fölött nincs a kimeneten logikai jelváltás (mivel nem jutott ki a bemeneti változás a kimenetre), így nem is lehet késleltetést mérni a bemeneti és kimeneti jelek között.

A 6.8. ábrán a D flip-flop ponált és negált kimeneteinek késleltetés-hőmérséklet függvénye látható. A két kimenet (Q és $\overline{Q}$) késleltetése közötti különbséget az okozza, hogy a D flip-flop cellában a negált kimenet áll elő először, a ponált kimenetet pedig az invertált kimenet ismételt invertálásával állítják elő. A ponált kimenet előtti inverter fokozat késleltetése miatt nagyobb késleltetést szenved a ponált kimenet.

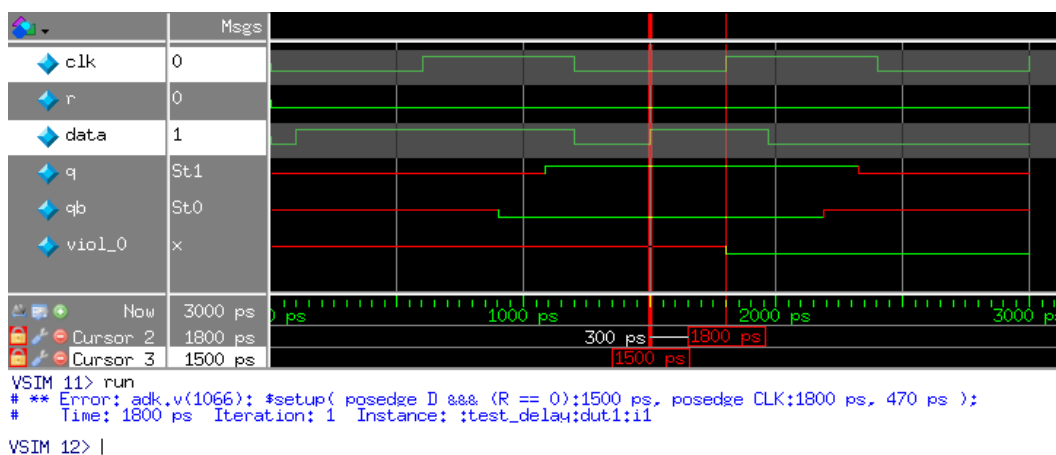
Az adat előkészítési és tartási idők szimulációját elvégeztem a logikai szimulátorral is. A szimuláció Verilog leírását az A. függelék A.1. kódrészlete tartalmazza.

A 6.9. ábrán látható, hogy a logikai szimulátor az adat-előkészítési idő nem teljesülését jelzi. Az adat bemenet (data) az órajel bemenet (clk) felfutó éle előtt változik meg 300 ps-mal. Az időzítéseket tartalmazó SDF fájl adatait tekintve a DFF cella D bemenetén legalább 470 ps-mal a CLK felfutó éle előtt már stabil jelnek kell lennie. Ez ebben az esetben láthatóan nem teljesül, az adat bemenetnek még 170 ps-mal korábban kellene stabilnak lennie. Ez azt jelenti, hogy nem teljesült a cellára vonatkozó adat-előkészítési idő (setup time).

A logikai szimulátor időzítéseket figyelő tulajdonsága jól felhasználható a hőmérséklet-függő késleltetések vizsgálatakor, ahol a kialakuló hőmérsékletek miatt megváltozó késleltetések a bemutatotthoz hasonló időzítési problémákat okozhatnak[C6].



6.8. ábra. D flip-flop késleltetés-hőmérséklet függvény



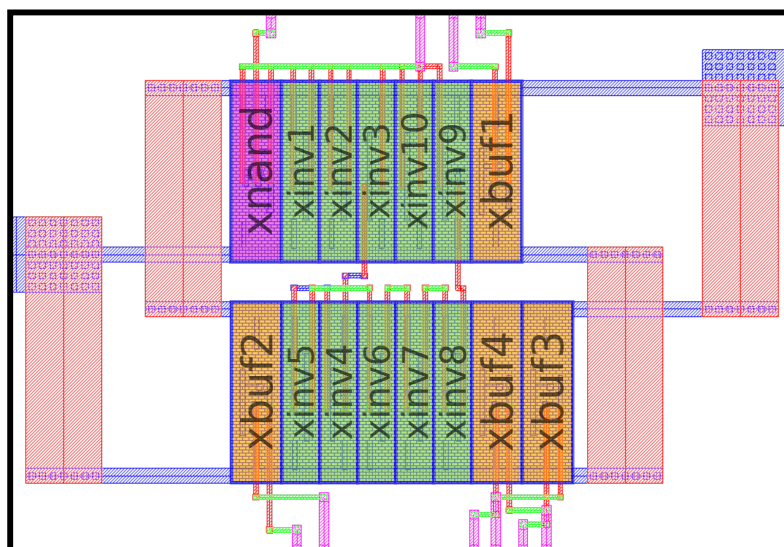
6.9. ábra. Időzítési hibajelzés a logikai szimulátorban

6.4. Demonstrációs áramkör

A hőmérsékletfüggő időzítések hatásait egy demonstrációs áramkörön mutatom be[J2],[C7]. Az áramkör egy egyszerű, 10 invertert tartalmazó gyűrűs oszcillátor melynek frekvenciáját elhangolja az inverterek hőmérsékletfüggő késleltetése. Az áramkör része még egy 4 bites puffertömb, mely számlálóként van meghajtva. A puffer cellák melededése is befolyásolja a gyűrűs oszcillátor eredő frekvenciáját. Az inverter-sor elején egy NAND kapu adja az indító jelet az oszcilláció megkezdésére.

Az áramkör a Mentor Graphics ASIC tervezői eszközkészlet (design kit) segítségével, TSMC 0,35 μm -es technológián lett szintetizálva és vezetékeztve. Az ASIC eszközkészlet oktatási és kutatási célra lett kifejlesztve és tartalmaz minden szükséges adatbázist a szintézishez, elhelyezéshez, vezetékeztetéshez és SPICE eszköz szintű szimulációhoz. A könyvtár celláinak magassága 120 μm . Egy puffer cella mérete 33,5 $\mu\text{m} \times 120 \mu\text{m}$. A felületi topológia a 6.10. ábrán

látható. Az áramkör felülete $A = 0,518 \text{ mm} \times 0,355 \text{ mm}$. Az elhelyező alkalmazás a cellákat két sorban helyezte el. A négy puffer cellát a tesztelő Verilog forrásprogramban hajtom meg, mintha az egy 4-bites számláló kimenete lenne. A számláló órajel periódusa $1 \mu\text{s}$. A puffer cellák segítségével az áramkörben nemcsak az inverterek saját-melegedése által okozott késleltetés változást lehet megfigyelni, hanem a külső meghajtású puffer cellák melegedése is szerepet játszik.



6.10. ábra. Demonstrációs áramkör felületi topológiája

Az áramkört strukturális Verilog leírás alapján a Mentor Graphics® Pyxis alkalmazás szintetizálta. Az automatikus elhelyezést és huzalozást (P&R) szintén a Pyxis programmal végeztem. Az áramkör Verilog strukturális leírása a 6.1. kódrészletben látható.

A demonstrációs áramkörtön bemutatható a felületi hőmérséklet eloszlása, a cellák teljesítménysűrűsége, a kialakuló forró pontok (hot-spots) helye és az időzítések változása a hőmérséklet függvényében. A cellák kapcsolási energiáját és hőmérsékletfüggő késleltetéseit SPICE szimulációkkal határoztam meg. A kapukésleltetéseket tartalmazó SDF fájl hőmérsékleti szélsőértékeit 0°C , 25°C és 65°C értékeknél állapítottam meg.

A bemutatott áramkör segítségével demonstrálni és ellenőrizni tudtam a téziseimben megfogalmazott állításokat. Az egyszerű áramköri terv segítségével a bemutatott módszereket SPICE szimulációkkal tudtam ellenőrizni és pontosságukat meghatározni. Egy több ezer cellát tartalmazó áramkör esetében a SPICE verifikációkat nem lehetett volna elfogadható időn belül végrehajtani. A módszerek és algoritmusok verifikálása után, a fejezet további részében ipari bonyolultságú áramkörök logi-termikus szimulációját is bemutatom.

6.4.1. Teljesítmény karakterizáció

A 6.4. szakaszban bemutatott demonstrációs áramkör esetében SPICE szimulációkkal határoztam meg a cellák dinamikus és statikus fogyasztását. Ezeket a fogyasztási adatokat használtam fel később a logi-termikus szimuláció bemenő paramétereként. A SPICE szimulációkkal történő karakterizálásra azért volt szükség a demonstrációs áramkör esetében, mert a felhasznált tervezői eszközkészletben (PDK) nem állt rendelkezésre a teljesítmény és energia

```

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

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

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

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

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

endmodule

```

6.1. kódrészlet. Gyűrűs oszcillátor Verilog leírása

adatokat tartalmazó Liberty adatbázis. A SPICE karakterizációkat a CCS összetett áram-alapú modell iránymutatása alapján végeztem[30, 32].

6.5. Teljesítménysűrűség és hőmérsékleti térkép

A demonstrációs áramkör meghajtását Verilog tesztkörnyezetben (testbench) végeztem el a Mentor Graphics® QuestaSim logikai szimulátorával. A logikai szimulátornak induláskor meg kell adni a cellák késleltetéseit tartalmazó SDF fájlt, így időzítéshelyesen tudja a szimulációt lefuttatni. Az időzítéseket tartalmazó SDF fájlt a szintézer illetve a P&R alkalmazás állítja elő, de a SPICE időzítési szimulációk alapján kézzel is összeállíthatóak. Az összeköttetések definiálva vannak az áramkör netlistájában, így a gyűrűs oszcillátor esetében csak egy indító jelről kell gondoskodni a testbenchben, és az oszcilláció a megadott késleltetések miatt máris elindul. Fontos, hogy a késleltetéseket tartalmazó SDF fájlt határozzuk meg a szimuláció indítása előtt, mert ellenkező esetben a logikai szimulátor zérus késleltetésekkel fogja a szimulációt lefuttatni, ami egy visszacsatolt gyűrűs oszcillátor esetében végtelen ciklust eredményezne. A testbenchben az indító jelen kívül gondoskodni kell még a négy puffer (BUF) cella meghajtásáról is. A puffer cellák fogyasztása által generált hőmérséklet változás befolyásolja az oszcillátor celláinak hőmérsékletét is[J2],[C7]. A puffer cellák egy számláló kimeneteit jelképezik. A puffer cellák kapcsolási aktivitását ($A()$) mutatja (6.1):

$$A(\text{xbuf1}) = 2A(\text{xbuf2}) = 4A(\text{xbuf3}) = 8A(\text{xbuf4}). \quad (6.1)$$

6.1-ből látható, hogy az xbuf1 cella az LSB és az xbuf4 cella az MSB a négy bites buszon.

A logi-termikus szimulációban a logikai szimulátorból kinyerhető minden cella kapcsolási aktivitása, így azt kell meghatározni, hogy az egyes cellák mennyi energiát fogyasztanak egy kapcsolási tranziens alatt. A teljesítményt, ami az áramkört fűti (Joule-hő) a kapcsolásokhoz tartozó energia-fogyasztásból és a kapcsolások számából lehet meghatározni. A fűtőteljesítményt a kapcsolások száma és a kapcsolási energia szorzata adja átlagolva egy adott időablakra (6.2).

$$P_{\text{fűtő}} = \frac{E_{\text{kapcs}} \cdot n}{\Delta t}, \quad (6.2)$$

ahol E_{kapcs} az egy kapcsoláshoz tartozó energia, n a kapcsolások száma, Δt a termikus átlagoló időablak.

A kapcsolási energiákat a cellakönyvtár Liberty formátumú adatbázisa tartalmazza minden cellára, de SPICE szimulációkkal is meg lehet határozni. A SPICE szimulációkkal történő meghatározást mutattam be a 4.6. szakaszban.

A logi termikus szimuláció végrehajtásához a fogyasztási adatokon és a kapcsolási aktivitáson kívül még szükség van az áramkör felületi topológiájára, a layoutra. Ezt az elhelyező és huzalozó alkalmazás szabványos formátumban szolgáltatja. Egy ilyen formátum a LEF/DEF formátum, mely szöveges fájlokban tárolja a könyvtári cellák layout adatait (LEF) és az adott áramkör topológiáját (DEF). A nyílt forrású szabványos LEF/DEF formátum feldolgozásához a szabadon elérhető programkönyvtárat (API) használtam mely ingyenes regisztráció után letölthető a www.si2.org weboldalról [57].

A testbench leírásban az oszcillátor indítását egy megfelelően hosszú kényszerített bil-lentéssel végzem el. A kényszerített jelváltás legalább olyan hosszú kell legyen, mint a teljes inverterlánc késleltetése, mivel ellenkező esetben korábban érne vissza a bemenetre a visszacsatolt kimeneti jel, és a szimulátor nem tudná pontosan meghatározni a kimeneti frekvenciát.

A logi-termikus szimuláció során a tesztáramkörnél termikus időlépésnek $\Delta t = 10$ ms-ot adtam meg. Az inverterek késleltetése néhányszor tíz ps, ahogy ez megfigyelhető a 6.2–6.4. ábrákon is. A digitális áramkör működési frekvenciája így a gigahertzes tartományban van. A 10 ms-os termikus időablak több nagyságrenddel az áramkör periódusideje fölött van, ugyanakkor elég nagy felbontást nyújt a termikus időállandók tartományában, amik a ms-os nagyságrendtől egészen a több tíz másodperces tartományig terjedhetnek a chipen belül.

A logi-termikus szimulátor termikus megoldó algoritmus (Laytherm) olyan struktúrák szimulációjára lett kifejlesztve, ahol a chip laterális méretei jóval nagyobbak a vastagságnál. Ha a chip egyes rétegeinek vastagsága összemérhető a laterális méretekkel, vagy akár vastagabb azoknál, a megoldó algoritmus egyre nagyobb hibát követ el. Ez a megszorítás azonban integrált áramköröknél nem okoz gondot, hiszen a nagy bonyolultságú integrált áramkörök, processzorok, jelfeldolgozó alkalmazások esetében a laterális méretek a $10 \text{ mm} \times 10 \text{ mm}$ -es nagyságrendben vannak, a szelet vastagsága pedig 1 mm körüli, a termikus interfész anyagokkal (Thermal Interface Material, TIM) és a kivezető kerettel (leadframe) együtt. Az ilyen típusú integrált áramkörök tehát legalább tízszer vékonyabbak laterális méreteiknél. A demonstrációs áramkör, melyen a logi-termikus működést és a verifikációt végeztem mindössze 15 cellából áll, annak érdekében, hogy még SPICE szimulációkat is elfogadható idő alatt lehessen végezni rajta. A kevés cella miatt az áramkör felülete $A = 0,518 \text{ mm} \times 0,355 \text{ mm} = 1,841 \cdot 10^{-7} \text{ m}^2$. Ez a laterális méret egy nagyságrenddel kisebb, mint egy ipari bonyolultságú $10 \text{ mm} \times 10 \text{ mm}$ -es integrált áramkör esetében.

A szimuláció során a rétegstruktúra három rétegből állt. A felső szilícium rétegen valószínűleg meg a sztenderd cellás demonstrációs áramkört. A szilícium réteg vastagsága egy ipari méretű szelet esetében $500\ \mu\text{m}$ körüli. A második réteg egy $50\ \mu\text{m}$ vastag TIM anyag. A legalsó harmadik réteg a kivezető keret, mely a kivezetések szétosztására és a chip rögzítésére, tartására szolgál. A kivezető keret anyaga Kovar, vastagsága $500\ \mu\text{m}$.

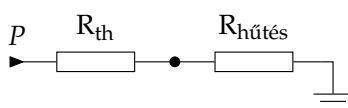
A logi-termikus szimulációhoz a struktúra vastagságát tízszeres faktorral csökkentettem, hogy a kívánt vastagság/felület arányt (aspect ratio, AR) megtartsam egy ipari méretű chiphez képest.

A kapcsolási aktivitás ismeretében a logi-termikus szimulátor meghatározza az áramkör átlagos fogyasztását, mely $4,022\ \text{mW}$ -ra adódott. A struktúra méreteit, anyagi paramétereit és a fűtőtéljesítményt foglalja össze a 6.1. táblázat.

6.1. táblázat. A struktúra fizikai paramétereit

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

A struktúra várható DC hőmérsékletét a méretek és fizikai paraméterek ismeretében analitikusan is kiszámítottam. A DC hőmérséklet közelítő meghatározásához ki kell számolni a struktúra hőellenállását és be kell számítani a peremfeltételek hatását is. A szimulációban a struktúra alját kivéve minden irányban adiabatikus határfeltételt határoztam meg. Lefelé egy $500\ \frac{\text{W}}{\text{m}^2 \text{K}}$ hőátadási együtthatójú hűtést helyeztem el. Az adiabatikus határfeltételek miatt a hőterjedés egydimenziós és a teljes struktúrán kialakuló hőmérséklet a struktúra hőellenállása és a hűtés hőellenállásának soros eredőjén a fűtőtéljesítmény hatására eső hőmérséklet. A struktúra hőellenállása (6.3) segítségével számítható.



6.11. ábra. Termikus eredő hőellenállás

$$R_{th} = \frac{1}{\lambda} \cdot \frac{d}{A} = \frac{1}{156,3} \cdot \frac{5 \cdot 10^{-5}}{1,841 \cdot 10^{-7}} \left[\frac{\text{m K}}{\text{W}} \cdot \frac{\text{m}}{\text{m}^2} \right] = 1,737\ \frac{\text{K}}{\text{W}}, \quad (6.3)$$

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

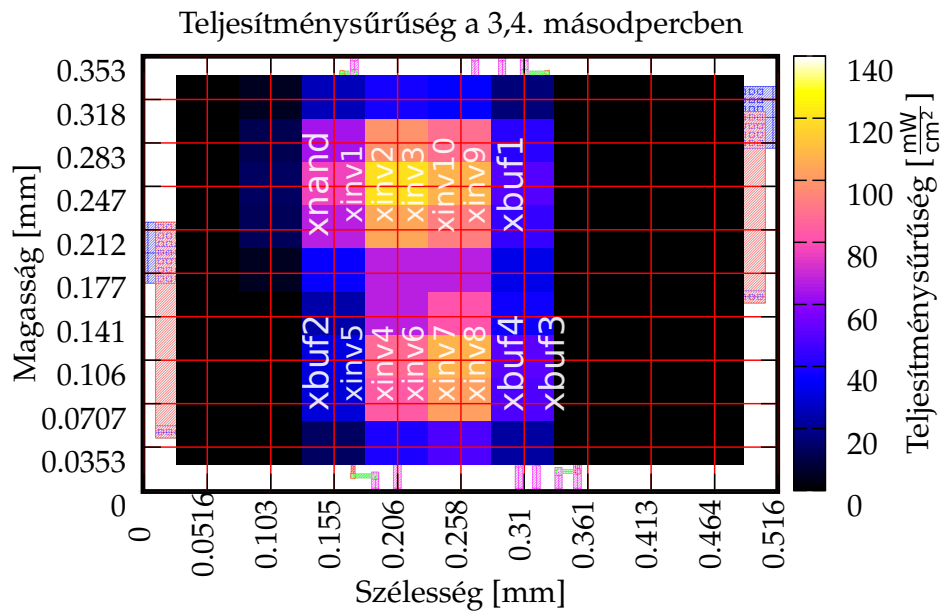
$$R_{hűtés} = \frac{1}{500} \cdot \frac{1}{1,841 \cdot 10^{-7}} \left[\frac{\text{m}^2 \text{K}}{\text{W}} \cdot \frac{1}{\text{m}^2} \right] = 10,863 \cdot 10^3\ \frac{\text{K}}{\text{W}}. \quad (6.4)$$

A bemenő teljesítmény $P = 4,022 \cdot 10^{-3}$ W. Az eredő DC hőmérséklet (6.5) alapján számítható:

$$T = P \cdot (R_{th} + R_{hűtés}) = 4,022 \cdot 10^{-3} \cdot (10,863 \cdot 10^3 + 1,737) = 43,7^\circ\text{C}. \quad (6.5)$$

Erre a hőmérsékletre melegedne fel az áramkör állandósult állapotban. Természetesen az elvi állandósult állapotot a végtelen időben érné el a struktúra. A logi-termikus szimulációk azt mutatták, hogy az áramkör DC hőmérséklete (6.5)-tel megegyezően $43,7^\circ\text{C}$ -hoz tart.

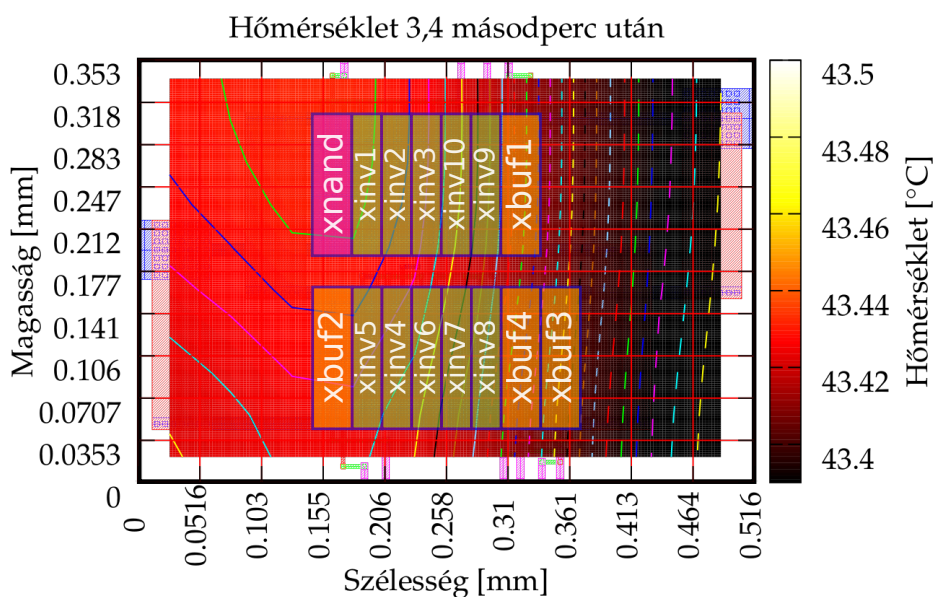
Az általam készített logi-termikus szimulátor a felületi teljesítménysűrűséget és hőmérsékleti eloszlást táblázatos formában és grafikusan is meg tudja jeleníteni. A táblázatos megjelenítés esetében egy szöveges fájlba kerül a fűtőteljesítmény, a cellahőmérséklet a kapcsolások száma az adott időegységben és a cellák koordinátái. A szimulációs eredmények minden időlépésben és minden cellára kimentésre kerülnek. Ebből az adatfájlból tetszőleges megjelenítőprogrammal lehet diagramokat, grafikonokat készíteni. A grafikus megjelenítést a nyílt forráskódú *Gnuplot* alkalmazással végeztem [58]. A 6.12. ábra mutatja a demonstrációs szimulációban kialakult teljesítménysűrűséget az áramkör felszínén. A grafikus megjelenítéshez a szimulációt 10×10 -es particionálással végeztem el.



6.12. ábra. Felületi teljesítmény eloszlás a 3,4. másodpercnél

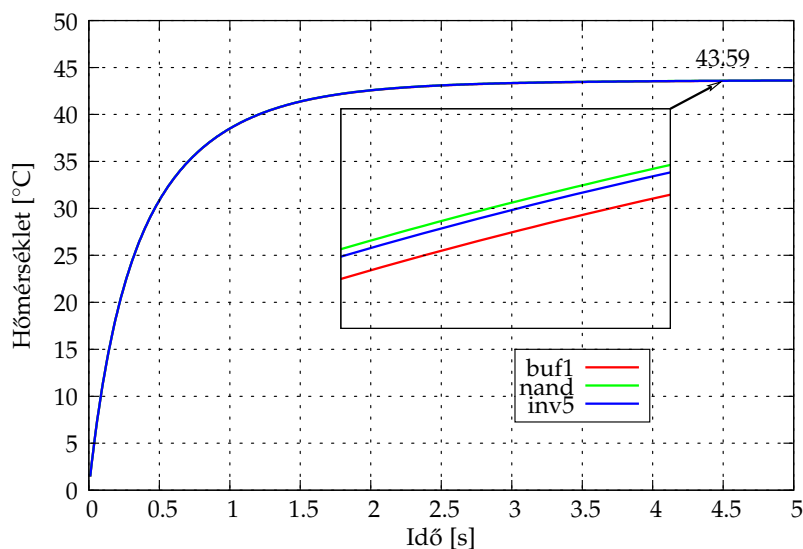
A 6.12. ábra a teljesítménysűrűséget mutatja a szimuláció 3,4. másodpercében. Az áramkör celláinak két sora megfigyelhető az ábrán. A 3,4. másodpercig kialakult hőmérséklet eloszlást mutatja a 6.13. ábra. Az ábrán az azonos hőmérsékletű pontokat izoterma vonalak kötik össze.

Az áramkör három jellemző cellájának hőmérsékleti görbéje az idő függvényében látható a 6.14. ábrán. Mind a 6.13. ábrán, mind a 6.14. ábrán megfigyelhető, hogy az áramkörben a legalacsonyabb és legmagasabb hőmérséklet értékek különbsége –tehát a hőmérsékleti gradiens a felületen– néhány tized fok. Ez a kevés különbség azonban így is képes elhangolni az oszcillációs frekvenciát, amint azt később be is mutatom. A hőmérsékleti gradiens értéke mellett nagyon fontos a teljes áramkör átlagos hőmérséklete is, hiszen a késleltetések hőmérsékletfüggőek és



6.13. ábra. Felületi hőmérséklet eloszlás a 3,4. másodpercnél

változnak az átlagos hőmérséklet változásával. Így például a gyűrűs oszcillátor frekvenciája 100 MHz-et is változhat egy teljes szimuláció során, ahol az áramkör önmelegedése miatt a felületi hőmérséklet több, mint 40 °C fokot növekszik.

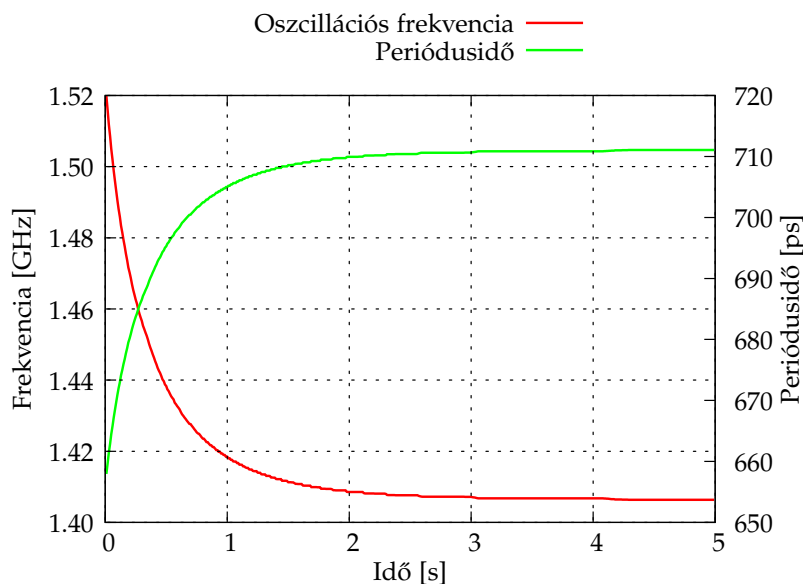


6.14. ábra. Cellák tranziens hőmérséklete

A 6.14. ábrán bejelöltem a 4,5. másodpercben kialakult felületi hőmérséklet értéket, ami 43,59 °C. Ez a hőmérséklet csak 0,11 °C-kal marad el a DC állapotbeli 43,7 °C-tól. A 6.14. ábrán tehát megfigyelhető, hogy az áramkör hőmérséklete 5 másodperc után a DC állapot közelébe áll be.

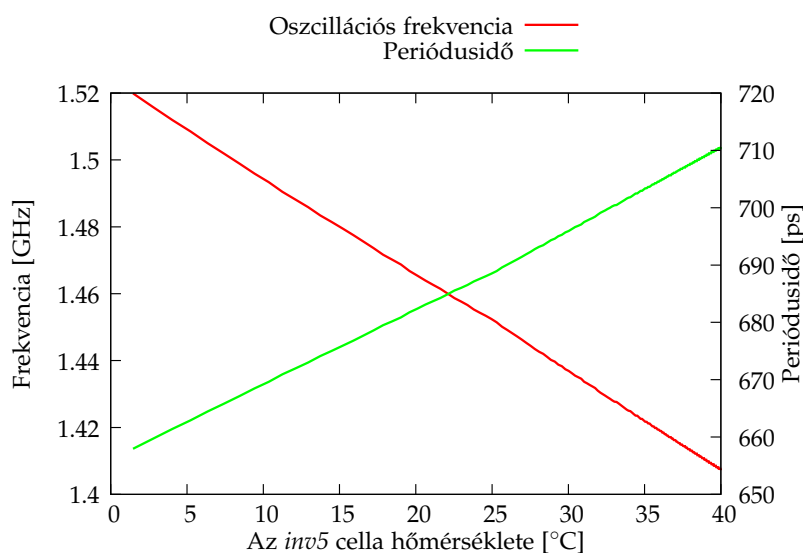
6.6. Hőmérsékletfüggő frekvencia

Az áramkör melegedése miatt a cellák késleltetései megváltoznak, így a gyűrűs oszcillátor kimeneti frekvenciája is elhangolódik[J2],[C7]. A 6.5. szakaszban látható, hogy az áramkör a DC állapotbeli hőmérsékletet 5 s után megközelíti. Az oszcillációs frekvencia változását a 0–5 s-os tartományban a 6.15. ábra mutatja a szimulációs idő függvényében.



6.15. ábra. A ring oszcillátor periódusideje és frekvenciája a szimulációs idő függvényében

A 6.16. ábrán a frekvencia és a periódusidő változása látható az `inv5` cella hőmérsékletének függvényében.



6.16. ábra. A ring oszcillátor periódusideje és frekvenciája az `inv5` cella hőmérsékletének függvényében

Az áramkör logi-termikus szimulációját 0 °C-on kezdtem és a DC állapotot megközelítve 43,7 °C végértékhez tart a struktúra átlagos hőmérséklete. Az állandósult állapotot megközelítve

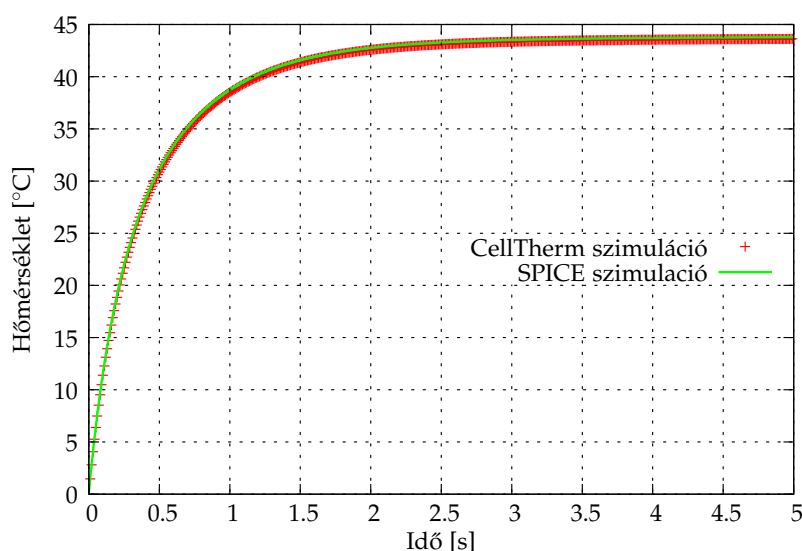
a gyűrűs oszcillátor kimeneti frekvenciája 1,406 GHz-hez tart. A kezdeti állapotban (0 °C-nál) a kimeneti frekvencia 1,519 GHz volt. A 43 °C-os változás hatására a kimeneti frekvencia 113 MHz-et csökkent. Ez –8,03 %-os változást jelent a végértékre vonatkoztatva. A 6.16. ábrán megfigyelhető, hogy a kimeneti frekvencia szinte lineáris az `inv5` cella hőmérsékletének függvényében. Ezt a linearitást magyarázza a 6.1. szakasz 6.3. ábrája is, ahol megfigyelhető, hogy az inverter késleltetés-hőmérséklet karakterisztikája közel lineáris a –20–+40 °C tartományban. Az egyes inverterek és a NAND cella késleltetésének hőmérsékletfüggése eltér ugyan egymástól, de végeredményben megfigyelhető, hogy a különböző késleltetés-hőmérséklet görbékből egy alapvetően lineáris frekvencia-hőmérséklet függvény alakul ki.

6.7. Validálás és verifikáció

Az általam kidolgozott logi-termikus szimulációs eljárást többféleképpen ellenőriztem[J2]. A szimulációk helyességét kézi számításokkal, SPICE tranzisztorszintű analóg szimulációkkal és más logi-termikus szimulátorral történő verifikációval bizonyítottam be. A 6.5. szakasz (6.5) egyenletét kézi számításokkal vezettem le, melyek megegyeztek a szimulációk során adódó hőmérséklet értékekkel.

6.7.1. SPICE–logi-termikus verifikáció

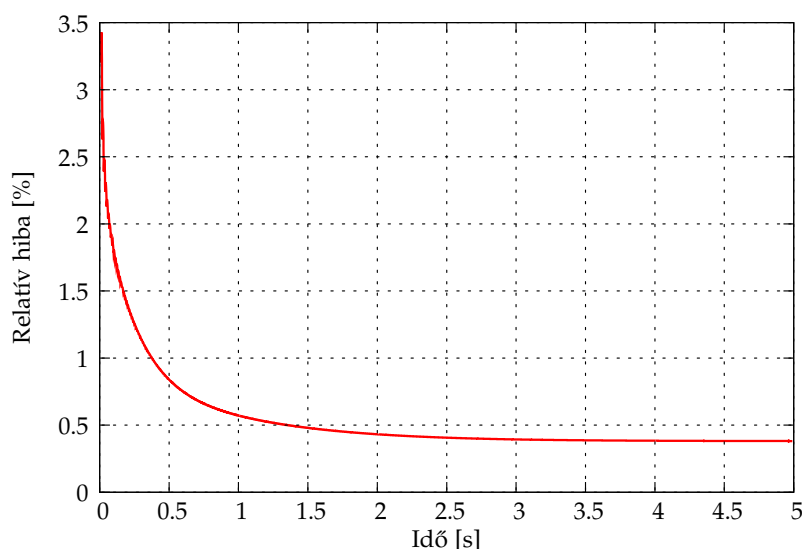
A CellTherm logi-termikus szimulátort a 6.4. szakaszban bemutatott demonstrációs áramkör segítségével verifikáltam egy SPICE kompatibilis szimulátor (Mentor Graphics® Eldo) felhasználásával. A SPICE szimulátorral a demonstrációs áramkör tranzisztor szintű megvalósítását szimuláltam. A 6.17. ábrán látható, hogy a SPICE szimulációval előállított eredmény és a CellTherm logi-termikus szimulátor számított hőmérsékletei jó egyezést mutatnak.



6.17. ábra. A CellTherm és SPICE szimulációk összehasonlítása

A 6.18. ábrán látható a SPICE és CellTherm szimulációk relatív százalékos hibája. Megfigyelhető, hogy a hiba a szimuláció teljes időtartama alatt 3,5 % alatt marad, és a rendszer

termikus időállandójánál kevesebb idő alatt 1 % alá csökken. A kezdeti nagyobb eltérést a két szimulátor között az okozza, hogy a CellTherm logi-termikus szimulátor csak az első termikus időlépés után tud hőmérsékleti adatot szolgáltatni (ami esetünkben 10 ms), hiszen az első termikus átlagoló időablakba eső fűtőtéljesítmény hatása ezekben a diszkrét időpontokban kerül kiszámításra. A termikus időablak szélességének csökkentésével a hiba 1 % alá csökkenése is korábban következik be. A termikus időablak szélességét azonban nem érdemes túlságosan kicsire választani, mert így szükségtelenül nagy időbeli felbontással vizsgálnánk a kialakuló hőmérséklet-eloszlást.



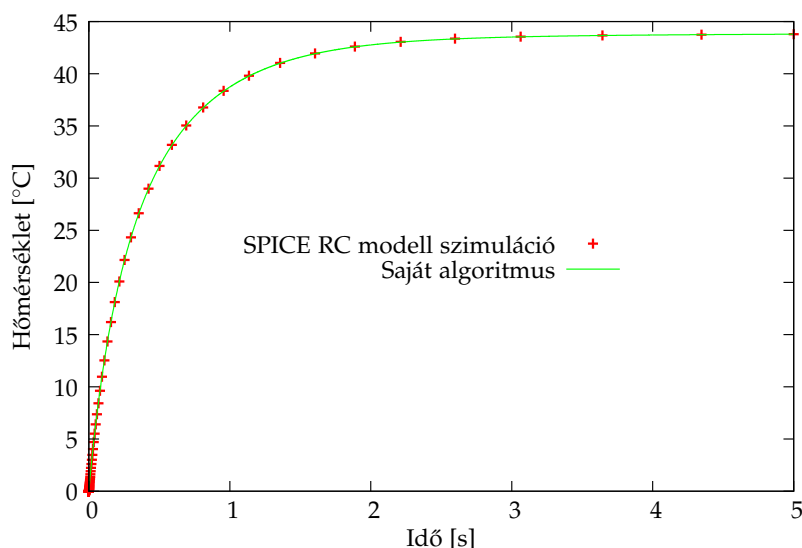
6.18. ábra. A CellTherm és SPICE szimulációk összehasonlítása

6.7.2. Saját megoldó algoritmus összehasonlítása SPICE szimulációval

A 3.2. szakaszban bemutatott termikus ekvivalens RC hálózat (Foster modell) számítására egyszerű algoritmust dolgoztam ki, melynek előnye, hogy nem kell bonyolult, esetleg nemlineáris hálózatomegoldót alkalmazni a hőmérsékleti eloszlás számításakor. Az általam a CellTherm logi-termikus szimulátorban használt algoritmus kifejezetten a Foster topológiájú lineáris hálózatok megoldására lett kidolgozva. Az algoritmus számítási pontosságát SPICE szimulációkkal verifikáltam.

A CellTherm logi-termikus szimulátor a rétegstruktúra, az anyagparaméterek és a felszínen lévő disszipáló alakzatok elhelyezkedése alapján meghatározza a struktúrát jellemző termikus RC hálózatot. Ennek a hálózatnak a megoldása szolgáltatja a hőmérsékleti eloszlást a bemenő disszipáció függvényében a chip felületén. A termikus ekvivalens RC hálózat modelljét egy egyedi szöveges fájlban tárolja az alkalmazás. Annak érdekében, hogy az így megadott RC hálózatot SPICE szimulátorban is szimulálni lehessen, elkészítettem egy konverziós alkalmazást (*rccalc*), mely szabványos SPICE bemeneti formátumba alakítja át a hálózat modelljét.

A 6.19. ábrán látható a saját algoritmus és a SPICE szimuláció eredménye ugyanarra a termikus ekvivalens RC hálózatra. Megfigyelhető, hogy a két szimuláció eredményei fedik egymást.



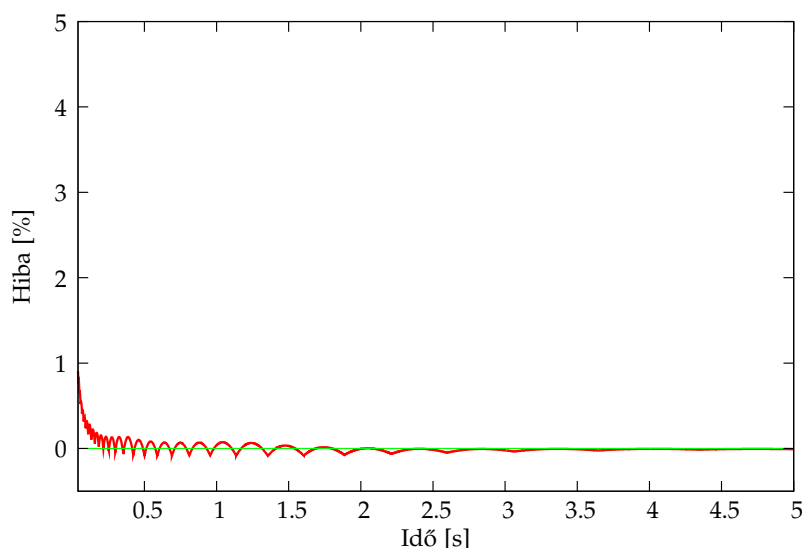
6.19. ábra. Saját megoldó algoritmus és SPICE szimuláció

A 6.20. ábra mutatja a két szimuláció relatív százalékos hibáját. Látható, hogy a hiba 1 % alatt marad a teljes szimuláció alatt. A SPICE szimulátor, mivel sokkal általánosabb és bonyolultabb megoldó algoritmust használ, valamint az időlépéseket adaptívan tudja állítani a konvergencia érdekében, pontosabb szimulációkat tud végezni. A pontosabb szimulációt azonban lassabban is hajtja végre. A verifikációhoz a SPICE-kompatibilis ELDO szimulátort használtam fel. Az ELDO szimulátor többféle számítási algoritmussal képes elősegíteni a szimulációk konvergenciáját. A nemlineáris hálózatok megoldásához negyedrendű Runge-Kutta (RK4) eljárást használ, ami számításigényesebb, mint az Euler iteráció, cserébe pontosabb eredményt ad és hibátűrőbb. Ez az oka annak, hogy az általam kidolgozott algoritmus valamennyi hibával számol az RK4 algoritmushoz képest, azonban a hiba így is 1 % alatti értékkel megközelíti a SPICE szimulátor eredményeit és akár több tízezer cella esetében is milliszekundumos nagyságrendű idő alatt szolgáltatja a hőmérsékleteket[J2].

6.7.3. A frekvenciamenet SPICE verifikációja

A SPICE tranzisztorszintű analóg szimulációk egy gigahertzen működő áramkör esetében rendkívül sok időt vesznek igénybe. A 6.4. szakaszban bemutatott demonstrációs áramkörön végrehajtott 64 μ s-ig futó SPICE szimuláció valós időben 34 percet vett igénybe egy HP Pro 3130 Microtower, Intel Core i3 3.2Ghz, 2 magos, hyperthread-es munkaállomáson. Ilyen finom szimulációs felbontással a SPICE szimulációk futtatása több hónapot venne igénybe. A CellTherm logi-termikus szimulátor által szolgáltatott hőmérsékletfüggő frekvencia adatok verifikálására eljárást dolgoztam ki, melynek segítségével a SPICE validációt is több nagyságrenddel rövidebb idő alatt lehet végrehajtani.

Az általam használt Mentor Graphics® Eldo SPICE kompatibilis szimulátort úgy lehet vezérelni, hogy egy hosszú futásidejű és több gigabyte méretű eredmény adatbázist előállító szimuláció indításakor meg lehet határozni, hogy milyen időpontokban és időintervallumokban készítsen kimenetet. Ezzel a módszerrel több nagyságrenddel fel lehet gyorsítani egy SPICE



6.20. ábra. A saját algoritmus és a SPICE szimuláció hibája

szimulációt a pontosság feláldozása nélkül. Az említett eljárással a szimulálandó áramkört a szimuláció elején egy meghatározott állapotba lehet hozni.

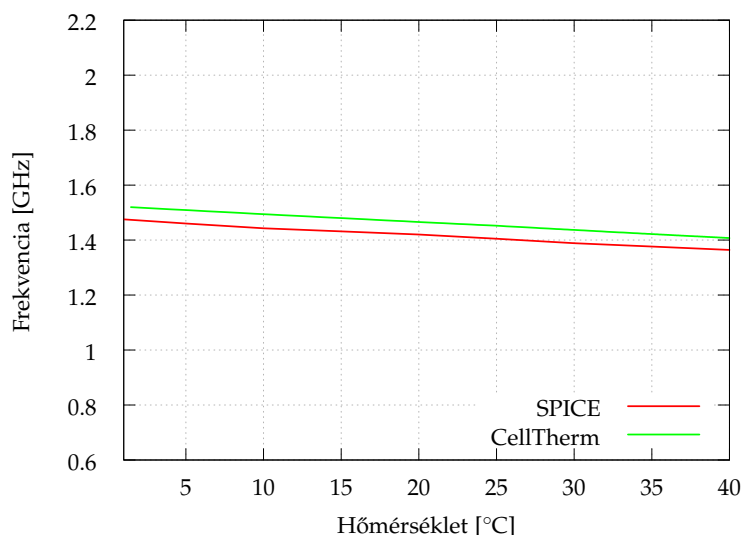
A SPICE szimuláció előkészítésekor figyelembe vettem az áramköri szimulációim korábbi eredményeit, ahol a gyűrűs oszcillátor kimeneti frekvenciája 1,5 GHz körüli értékre adódott. A SPICE szimulációban tehát elegendő egy 1 GHz-es mintavételi frekvenciájú időablakban vizsgálni a gyűrűs oszcillátor kimenetét. Ez azt jelenti, hogy, ha a kimeneti frekvenciát például egy 20 ns-os időablakban vizsgáljuk, akkor 1 GHz-es oszcillációs frekvencia esetén is legalább 20 jelváltást tudunk megfigyelni a kimeneten. Ebből a 20 jelváltásból átlagolással már meg lehet határozni a kimeneti frekvenciát egy adott hőmérsékletre, és mégsem kell több másodpercig futtatni a szimulációt. A 20 ns-os időablak szimulációját az analóg szimulátor pár másodperc alatt tudja szolgáltatni 1 GHz-es nagyságrendű oszcilláció mellett. A verifikáció során a SPICE szimulátort úgy állítottam be, hogy mindig a 801–821 ns közötti időablakban átlagolja az oszcillátor kimeneti frekvenciáját. Ebben az időablakban már minden digitális tranzienst lejátszódott a 800 ns előtt, mivel az oszcilláció indító jelét 1 ns-nál adtam ki, így a 800. nanoszekundumra 1,5 GHz frekvencián már $800/0,66 = 1212$ jelváltás történt a kimeneten.

Egy 20 ns-os időablak szimulációja pár másodpercet vett igénybe, így ezt a szimulációt megismételtem több különböző környezeti hőmérsékletre, melyet a SPICE szimulátorban lehetett beállítani. Az analóg szimulációkat sorozatban 0 °C és 45 °C között végeztem el 5 °C-os lépésekben. Az egyes hőmérsékletekhez tartozó átlagos kimeneti frekvenciák adatpontjait összekötve meg tudtam határozni a gyűrűs oszcillátor frekvenciamevetét a hőmérséklet függvényében.

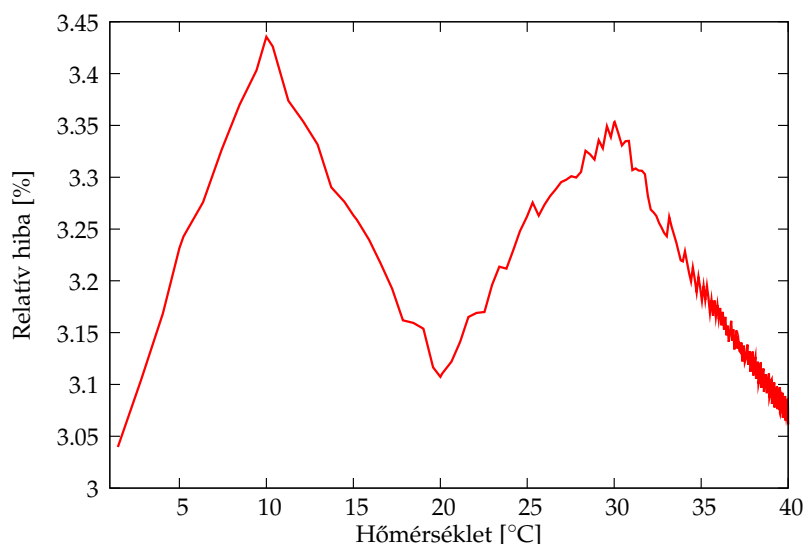
Az így kapott szimulációs eredményeket összevetettem a korábban logi-termikus szimulációval nyert frekvencia-hőmérséklet függvényvel (lásd a 6.16. ábrát). Az összevetés eredményét mutatja a 6.21. ábra.

A két szimuláció közötti relatív százalékos eltérést mutatja a 6.22. ábra. Látható, hogy az eltérés átlagosan 3,2 % körül ingadozik, és végig 5 % alatt marad.

Ez az eltérés tovább javítható, ha a késleltetéseket tartalmazó SDF fájlban minden cella példányhoz egyesével határozzuk meg a konkrét késleltetéseket, mind a felfutó, mind a lefutó irányba. Jelenleg cellatípusonként (inverter, puffer, NAND kapu) egy-egy késleltetés



6.21. ábra. Frekvenciamenet a hőmérséklet függvényében SPICE szimulációval



6.22. ábra. SPICE és CellTherm frekvencia-hőmérséklet függvény relatív eltérése

adat szerepel az SDF adatbázisban. Pontos SPICE szimulációkkal lehetőség van minden cella példány egyedi időzítési karakterizációjára, illetve ezeket az adatokat a Liberty adatbázis is tartalmazza, minden fel- és lefutó élre. A Liberty adatbázisban a cellák karakterizált adatai függenek a cella állapotától is, minden lehetséges állapotra külön-külön 2 vagy 3 dimenziós táblázatok szolgáltatják az időzítési és energia adatokat.

6.7.4. CellTherm és LogiTherm összehasonlítás

A BME Elektronikus Eszközök Tanszékén a Terminator FP7-es keretrendszerbeli Európai Uniós projekt keretében a 2010–2013. években a logi-termikus és elektro-termikus szimulátorok kutatása folyt. A projekt részeként készült el a CellTherm logi-termikus szimulátor, melyet

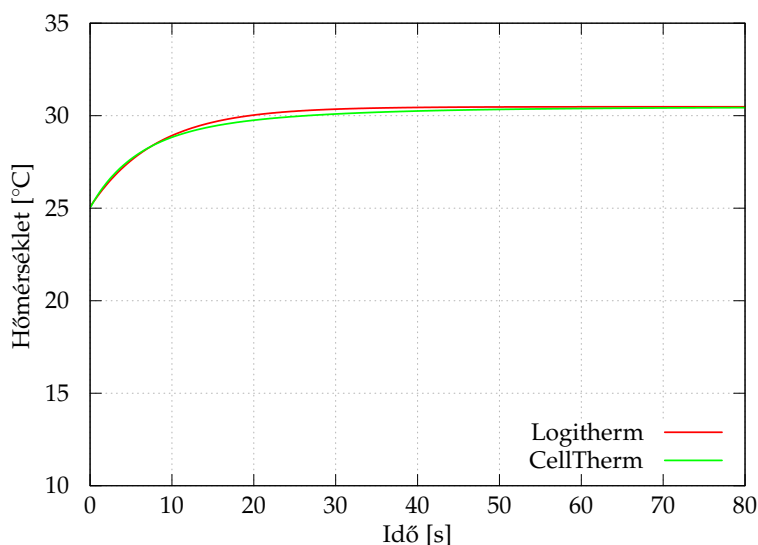
jelen értekezés mutat be. A projektben különböző elveken működő logi-termikus szimulátorok kutatása valósult meg. Egy ilyen, a szintén az Elektronikus Eszközök Tanszékén kifejlesztett *LogiTherm* szimulátor is [C9], mely a SuNRed algoritmussal véges térfogatok módszerén alapulva oldja meg a hővezetési tér számítását [59],[60],[61]. A CellTherm és a LogiTherm szimulátorok működésükben alapvetően eltérnek, mégis az általuk szolgáltatott szimulációs eredményeknek egyezniük kell.

A két logi-termikus szimulátor eredményeit ellenőriztem és összevetettem. A szimulált struktúra adatait a 6.2. táblázat összegzi.

6.2. táblázat. A struktúra fizikai paraméterei

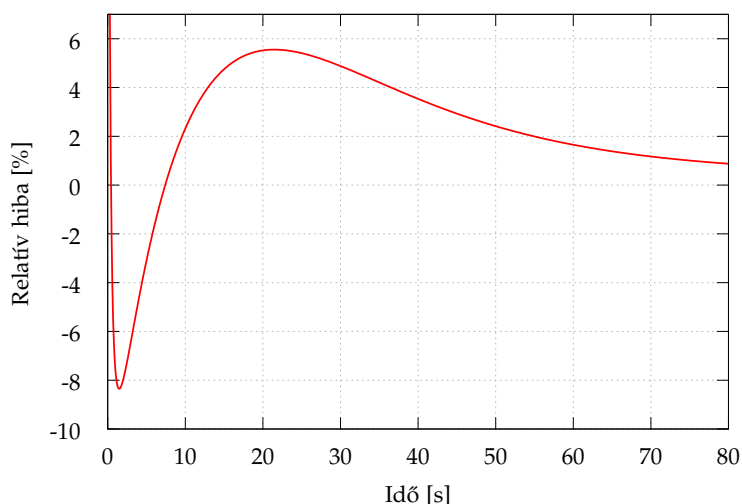
	Jelölés	Érték
Felület	A	0,516 mm $\times$ 0,353 mm
Si réteg vastagsága	d	0,05 mm
Si termikus vezetőképessége	λ	156,3 $\frac{\text{W}}{\text{mK}}$
Fűtőteljesítmény	P	10 mW
Alsó hűtés	HTC	10 $\frac{\text{W}}{\text{m}^2\text{K}}$
Környezeti hőmérséklet	T_{amb}	25 °C

A 6.23. ábrán látható, hogy a két szimulátor által generált hőmérsékleti görbe jól közelíti egymást.



6.23. ábra. CellTherm és LogiTherm összehasonlítás

A 6.24. ábrán megfigyelhető a két szimulátor eredményeinek relatív százalékos eltérése. A szimulátorok eredményeinek relatív eltérése abszolút értékben a szimuláció elején 10 % alatti, a szimulációs idő növekedtével csökken és 0 %-hoz tart.



6.24. ábra. CellTherm és LogiTherm relatív százalékos hibája

6.8. Ipari méretű és bonyolultságú áramkörök szimulációja

6.8.1. 1000 inverteres gyűrűs oszcillátor

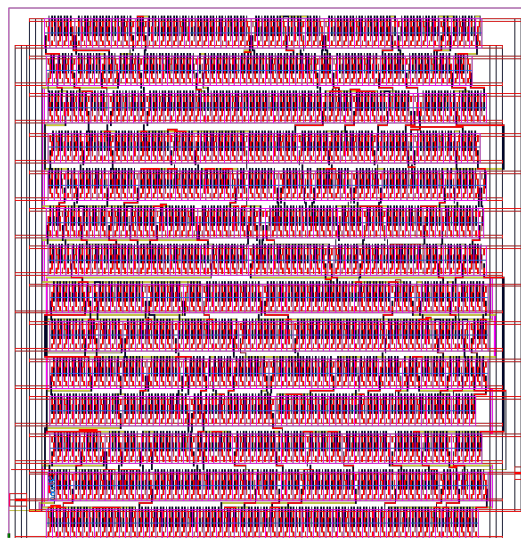
A Terminator projekt keretében az Elektronikus Eszközök Tanszéke együttműködést folytatott több olaszországi egyetemmel és az STMicroelectronics mikroelektronikai tervező és gyártó céggel. A projektben résztvevő olasz egyetemek közül szoros együttműködés valósult meg a University of Bologna (UNIBO) és a Politecnico di Torino (POLITO) egyetemekkel. Az általam kidolgozott logi-termikus szimulátort a projekt partnerek által tervezett ipari digitális integrált áramkörökön is lefuttattam. Annak érdekében, hogy a logi-termikus szimulátor ipari alkalmazhatóságát bemutassam, olyan áramkörökön kellett tesztelnem a szimulátort, melynek bonyolultsága és méretei egy ipari célú áramkör komplexitásával összemérhetőek.

Egy általam megvalósított integrált gyűrűs oszcillátor áramkört hoztam létre, melyben 1000 inverter cellából épül fel az oszcillátor. Az ezres nagyságrendű cellaszám már megfelel egy ipari bonyolultságú rendszernek. Az oszcillátor áramkörben az invertereken kívül még egy indító NAND cellát is elhelyeztem az oszcilláció indítására. Az oszcillátort TSMC 350 nm-es CMOS technológián valósítottam meg. A tesztáramkör strukturális Verilog leírásának elkészítésére cél-alkalmazást dolgoztam ki. A strukturális leírás szintézisét, elhelyezését és vezetékezését a Mentor Graphics® Pyxis környezetben végeztem. A szintetizált áramkör felületi topológiája a 6.25. ábrán látható.

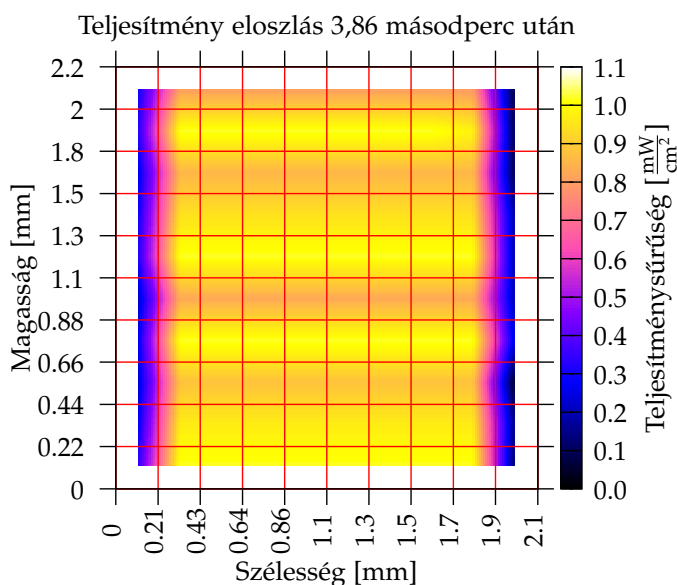
Az áramkör felülete a kivezető gyűrű nélkül $2,13 \text{ mm} \times 2,19 \text{ mm}$. A logi-termikus szimuláció teljesítménysűrűségét mutatja a 6.26. ábra.

Az eredő hőmérséklet eloszlást mutatja a 6.27. ábra az állandósult állapot közelében. Az állandósult állapot $0,193^\circ\text{C}$ körül van a kezdeti 0°C -hoz képest. A szimulált három rétegű struktúra felső szilícium rétege $500 \mu\text{m}$, a középső ragasztó TIM réteg $50 \mu\text{m}$, az alsó Kovar réteg szintén $500 \mu\text{m}$ volt. A szimulált struktúrát minden irányból adiabatikus határfeltétel vette körbe, az alsó határfeltételt kivéve, ahol egy $5000 \frac{\text{W}}{\text{m}^2 \text{K}}$ hőátadási tényezőjű hűtést helyeztem el.

Az áramkör az állandósult állapotbeli hőmérsékletet $3,86 \text{ s}$ alatt közelítette meg.



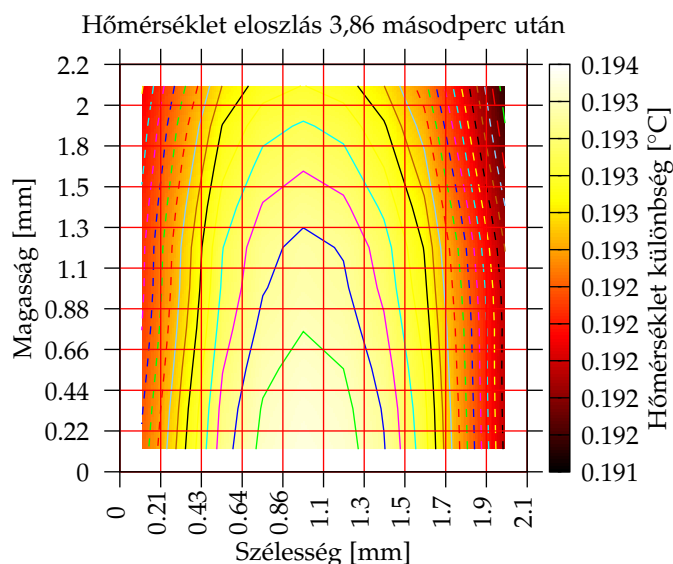
6.25. ábra. 1000 inverteres gyűrűs oszcillátor



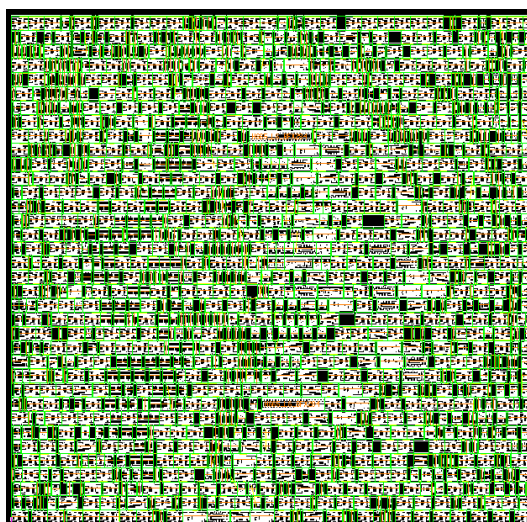
6.26. ábra. 1000 inverteres gyűrűs oszcillátor teljesítménysűrűsége

6.8.2. STMicroelectronics–POLITO tesztáramkör

A CellTherm logi-termikus szimulátorral megvizsgáltam a Terminator[4, 5] projekt keretében, az STMicroelectronics 65 nm-es technológiájára tervezett teszt áramkört is. Az áramkört a Torinói Műszaki Egyetemen (Politecnico di Torino, POLITO) tervezték, mely jelfeldogozó feladatokat lát el. A tesztáramkör több DSP (Digital Signal Processing), LFSR (Linear Feedback Shift Register) blokkot tartalmaz. A chip 1490 sztenderd cellát tartalmaz és felülete a kivezető gyűrű nélkül $0,12 \text{ mm} \times 0,12 \text{ mm}$. A szintézist a Cadence® Resource Compiler (RC) alkalmazás készítette, az elhelyezést és huzalozást a Velocity (korábban Silicon Encounter) alkalmazás végezte. A DSP chip felületi topológiáját a 6.28. ábra mutatja. A struktúra rétegei és vastagságai megegyeznek a 6.8.1. alszakaszban ismertetettel.



6.27. ábra. 1000 inverteres gyűrűs oszcillátor hőmérséklet eloszlása

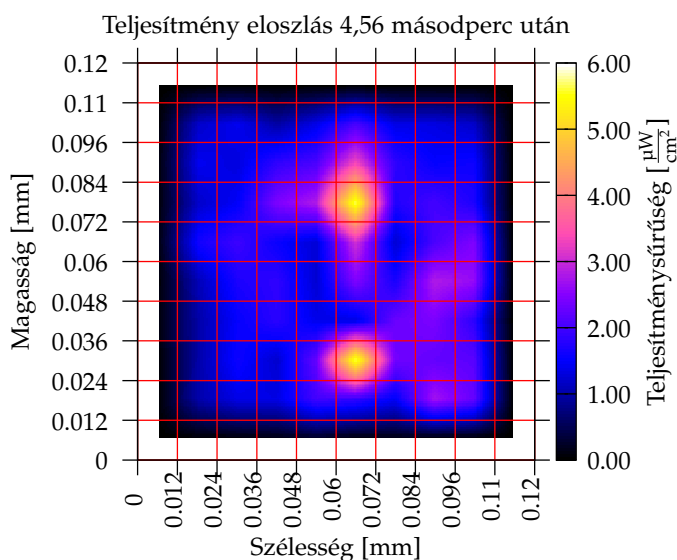


6.28. ábra. POLITO DSP chip

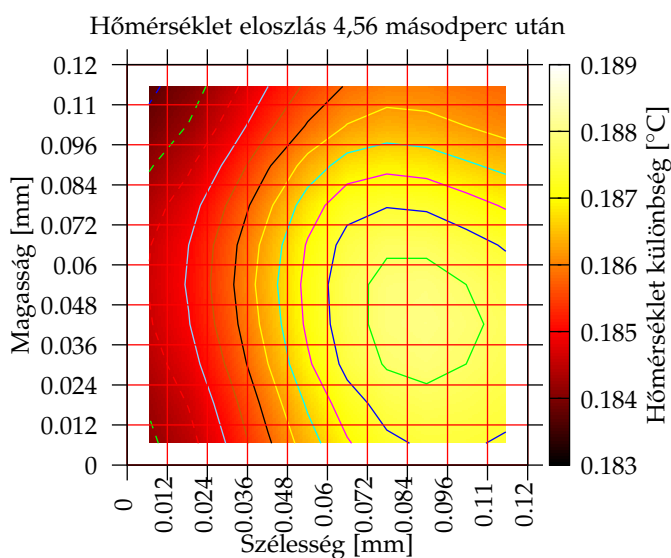
A 6.29. ábrán látható a jelfeldolgozó áramkör felületén kialakult teljesítmény eloszlás térkép. Megfigyelhető, hogy az áramkörben két helyen van a környezet fogyasztásánál magasabb érték. A teljesítménysűrűség skálát tekintve látható, hogy mértékegysége $\frac{\mu W}{cm^2}$, ami elég kicsiny fogyasztást jelent. A kis fogyasztást az okozza, hogy az áramkör –az előzőekben bemutatott áramkörökkel szemben– 65 nm-es csíkszélességű technológián lett tervezve és itt a cellák méretei és fogyasztásai is jóval kisebbek, mint a TSMC 350 nm-es CMOS technológiáján.

Az eredő hőmérsékleti térkép látható a 6.30. ábrán. A skála értékeiből látható, hogy a felületen szinte homogén a hőmérséklet eloszlása, köszönhetően a jó minőségű alsó hőelvezetésnek ($5000 \frac{W}{m^2 K}$) és a szilícium jó hővezető képességének. Így az áramkör hőmérséklete a $0^\circ C$ -os környezethez képest 4,56 s után $0,189^\circ C$ lett.

A bemutatott ipari bonyolultságú digitális sztenderd cellás áramkörök szimulációjával igazoltam, hogy az általam kidolgozott logi-termikus szimulátor alkalmas ipari méretű és



6.29. ábra. POLITO DSP chip teljesítmény eloszlása



6.30. ábra. POLITO DSP chip hőmérséklet eloszlása

komplexitású áramkörök szimulációjára. A digitális áramkör cellaszámának növekedése a particionálással végzett szimuláció során nem okozza a szimuláció idejének arányos növekedését, mivel ebben az esetben a partíciók száma határozza meg a szimuláció gyorsaságát, nem pedig a cellák száma. A particionálás nélkül a szimuláció sebessége fordítottan arányos az áramkör cellaszámának négyzetével. A particionálást felhasználva a cellák mennyisége szintén hatással van a szimulációs időre, de mégsem négyzetes függvénnel növekszik.

Az áramkör logi-termikus szimulációját 5 másodperces szimulációs időig végeztem. Ekkorra az áramkör már az állandósult hőmérsékletet 1% alatti eltéréssel megközelítette. A szimuláció elején a topológia (layout) beolvasása, az áramkör hierarchiájának feltérképezése, a cellák fogyasztási adatainak beolvasása, a termikus időállandó spektrumok meghatározása és ezekből a struktúra termikus modelljének elkészítése történik meg. A CellTherm alkalmazás

6.3. táblázat. A logi-termikus számítások időtartama

Részfeladat	Időtartam
Layout beolvasása	0,2 s
Hierarchia feltérképezése és fogyasztási adatok beolvasása	147,2 s
Termikus időállandó spektrumok meghatározása	76,5 s
Termikus modellek elkészítése	2,3 s
10 ms-os időlépés szimulációs ideje átlagosan	38–40 s
Teljes szimuláció időtartama (5 s-ig)	19 387,5 s (5,385 óra)

ezeket a számítási időket minden lépéshez meghatározza. Az egyes részfeladatok időtartamát a 6.3. táblázat tartalmazza. A szimulációkat egy HP Pro 3130 Microtower, Intel Core i3 3.2Ghz, 2 magos, hyperthread-es munkaállomáson végeztem. Az áramkör 1490 sztenderd cellát tartalmaz. A szimulációt az 1.2. altézisben bemutatott particionálási eljárással hajtottam végre 10×10 partícióra. A termikus szimulációs időablak szélessége 10 ms volt.

7. fejezet

Nagy felbontású hőmérsékletfüggő időzítési modell

3. tézis. Új időzítési modellt dolgoztam ki, melynek segítségével az integrált áramkörök logi-termikus szimulációja során a cellák hőmérséklettől függő időzítéseit hőmérséklet-késleltetés függvényekkel lehet figyelembe venni. Az általam kidolgozott modell segítségével olyan hatások is szimulálhatóak, mint a késleltetések inverz hőmérsékletfüggése. A modell megalkotásához a hőmérséklet-késleltetés függvényeket előzetesen minden cellára karakterizálni kell. Az új modell segítségével kimutattam, hogy a jelenleg használatos időzítési modellek bizonyos hőmérsékleti tartományokban nem megfelelően írják le a tényleges hőmérsékletfüggő működést. Javaslatot tettem a Liberty fájlformátum új modellel történő bővítésére [J5],[C8].

7.1. A jelenlegi időzítési modellek problémája

Napjainkban az ipar legfejlettebb integrált áramkör tervező rendszereiben a sztenderd cellák időzítési adatait különböző adatbázisokban tárolják. A legjellemzőbb, hogy a cellakönyvtár celláinak időzítési és késleltetési adatait Liberty formátumú fájlokban helyezik el és a különböző áramköri tervező alkalmazások ezeket használják fel szintézis, elhelyezés és vezetékezés közben. A logikai szimulátorok a szintézer alkalmazások által az elhelyezés és huzalozás előtt (pre-layout) vagy után (post-layout) létrehozott időzítési adatbázisokat használják fel. Az ilyen adatbázisok legelterjedtebb és széles körben támogatott formátuma a Standard Delay Format (SDF). Az SDF állományokat a szintézer alkalmazás hozza létre a szintézis illetve az elhelyezés-huzalozás lépésben a Liberty adatbázis alapján. A Liberty adatbázisban a cellakönyvtár elemeinek késleltetési adatai parametrikusan vannak meghatározva. Ez azt jelenti, hogy a Liberty adatbázis előállításakor a könyvtári cellák időzítési adatait több különböző bemeneti jelmeredekség és kimeneti terhelő kapacitás értékre határozzák meg. A szintézer így egy konkrét topológia és huzalozási elrendezés függvényében határozza meg az SDF adatbázis késleltetési adatait. Különböző elhelyezési és huzalozási topológiák tehát különböző SDF adatokat eredményeznek. A logikai és időzítési szimulációk során így lehet pontosabban figyelembe venni a fizikai elhelyezés és huzalozás hatásait.

Az említett időzítési modellek hátránya, hogy a pontos karakterizációkat előre meghatározott, állandó környezeti és gyártási paraméterek mellett hajtják végre. Ilyen fontos, a gyártást és az áramkör működését befolyásoló tényezők az áramkör tápfeszültsége, a hőmérséklet (saját és környezeti hőmérséklet) valamint a gyártásból származó bizonytalanságok. Az így kapott karakterizációs adatok az adott környezeti és gyártási paraméterekhez tartoznak, így más paraméterek esetén a cellakönyvtárat újra kell karakterizálni a megadott szélsőértékekre (design corner). A tápfeszültségek és gyártási bizonytalanságok skálázhatóságára megfelelő megoldást ad a 4. fejezetben bemutatott CCS modellezési technika. Az áram alapú modell azonban a változó saját és környezeti hőmérsékletek hatását ugyanúgy nem veszi figyelembe, mint a hagyományos NLDM vagy SPDM modellek. Különböző hőmérsékleti értékekre a cellák időzítési és teljesítmény karakterizációját újra el kell végezni.

A jelenlegi időzítési modellekkel a digitális integrált áramkörök melegedése által okozott késleltetés változásokat csak az előre karakterizált hőmérsékleti szélsőértékeken tudjuk pontosan meghatározni, az ezektől eltérő hőmérsékleteken a szélsőértékekhez tartozó késleltetés értékek lineáris interpolációjával kell számolni. Az interpolációval meghatározott késleltetések nem biztosítanak olyan pontos értékeket, mintha a cellák hőmérsékletfüggő késleltetés karakterizációját széles hőmérsékleti tartományban Celsius fokonként meghatároztuk volna.

A cellakönyvtár celláinak késleltetéseit egy széles hőmérsékleti tartományra Celsius fokonként karakterizálva részletesebb időzítési és logi-termikus szimulációs eredmények nyerhetők. Az időzítések, a kapcsolási aktivitás, a fogyasztás és a hőmérsékletek kölcsönösen függenek egymástól, hiszen a késleltetések megváltozásával egységnyi idő alatt megváltozik a kapcsolási aktivitás, ami a dinamikus fogyasztást változtatja, aminek következtében megváltozik az eszköz hőmérséklete is. A változó hőmérséklet viszont az időzítések változását is jelenti.

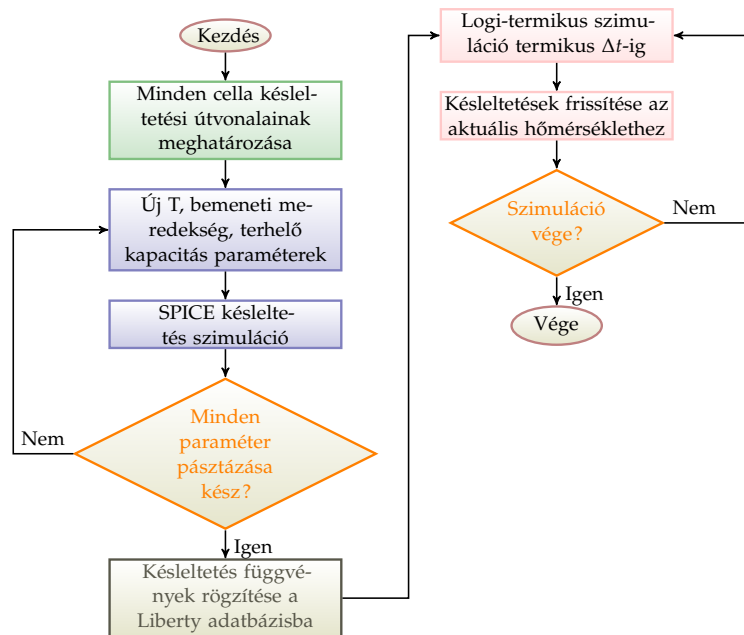
Értekezésemben olyan hőmérsékletfüggő időzítési modellt dolgoztam ki, ahol a sztenderd cellás könyvtár celláinak késleltetéseit széles hőmérsékleti tartományban karakterizáltam. A cellák hőmérséklettől függő időzítéseit hőmérséklet-késleltetés függvényekkel lehet figyelembe venni a logi-termikus szimulációk során. A modell segítségével pontosabban lehet leírni az áramkör hőmérsékletfüggő viselkedését.

Az általam kidolgozott modell egyszerűen integrálható a Liberty formátumú adatbázisba, melyet úgy alkottak meg, hogy hatékonyan lehessen benne többdimenziós függvényeket eltárolni és felhasználni.

7.2. Az új modell leírása és előállítása

Az új modell előállítását lépésenként részletezve mutatom be. A szöveges magyarázat egyszerűbb érthetőségét a 7.1. ábra segíti.

1. Az új modell előállításához ismerni kell a cellakönyvtár celláinak késleltetési útvonalaikat. Ezekre az útvonalakra határozhatók meg egy cella terjedési késleltetései. Egy cella esetében annyi késleltetést lehet definiálni, ahány bemenet-kimenet útvonala van a cellában. Egy két bemenetű, egy kimenetű NAND kapu esetében például két útvonala létezik a bemenetek és a kimenet között. A hőmérséklettől függő késleltetéseket tehát meg kell határozni minden lehetséges útvonala. Az említett időzítési útvonalakokat nem szükséges kézzel meghatározni, mivel a cellakönyvtár rendszerint tartalmazza a cellák viselkedési leírását Verilog és/vagy VHDL nyelven, ami definiálja ezeket az útvonalakokat. A cellák



7.1. ábra. Az új modell létrehozásának és használatának lépései

HDL szintű leírásának hiányában a tisztán logikai és időzítési szimuláció sem lenne lehetséges.

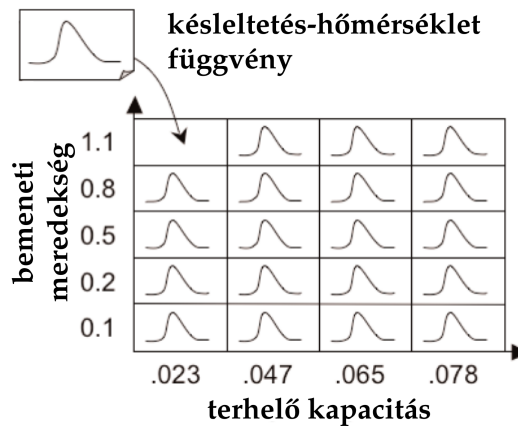
2. A cellák nagy felbontású hőmérsékletfüggő késleltetéseit analóg SPICE szimulációkkal tudjuk meghatározni a cellák tranzisztor szintű kapcsolásának ismeretében. Az általam használt ELDO analóg szimulátor tartalmaz a jelterjedési késleltetés mérésére használható nyelvi parancsokat, melyek segítségével a késleltetések egyszerűen kinyerhetők az analóg szimulációból.

Egy inverter késleltetését az `.EXTRACT TRAN VECT TPD(V(A), V(Y))` paranccsal tudjuk meghatározni. Az analóg szimulációnál a cella hőmérsékletének változását a környezeti hőmérséklet változtatásával vesszük figyelembe. Egy hőmérsékleti értékre egy késleltetés értéket fogunk kapni (minden útvonalra).

Késleltetés-hőmérséklet függvényekhez tehát szükség van a környezeti hőmérséklet (ambient temperature) széles tartományban történő pásztázására. Így minden hőmérsékleti pontra kiszámítunk egy késleltetés értéket. A pontosságot a hőmérsékleti mintavételi pontok számával tudjuk befolyásolni. Az ELDO analóg szimulátor képes tetszőleges paraméter értékét pásztázni, ami lehetővé teszi, hogy egyszerűen határozzunk meg késleltetés-hőmérséklet függvényeket. A `.STEP TEMP -20 85 1` utasítással -20°C és 85°C között 1 fokként tudjuk pásztázni a környezeti hőmérsékletet.

A terjedési késleltetés azonban nem csak a hőmérséklet függvénye, hanem függ a bemenetre adott jel felfutási/lefutási meredekségétől és a cella kimenetére kapcsolt terhelő kapacitástól is. Ezekkel tehát szintén paraméterezni kell a kapott késleltetés-hőmérséklet függvényeket. Az ELDO szimulátor segítségével ezek az adatok szintén egyszerűen kinyerhetők a bemeneti meredekség és a terhelő kapacitás értékének léptetésével (`.STEP` parancs). A nagy felbontású parametrikus késleltetés-hőmérséklet modell tehát minden

hőmérséklet, bemeneti meredekség, terhelő kapacitás ponthoz rögzít egy késleltetés értéket. A modell adatait így négy dimenziós táblázatokban lehet eltárolni a logi-termikus szimulációhoz. Erre mutat példát a 7.2. ábra.



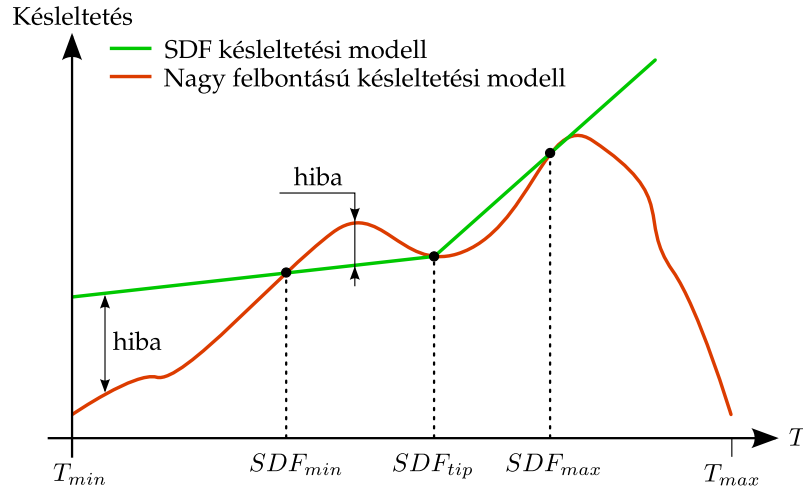
7.2. ábra. A késleltetés-hőmérséklet függvények a Liberty adatbázisban

Ilyen négy dimenziós táblázatok tárol hatékonyan a 4. fejezetben bemutatott Liberty formátum és a CCS áram alapú modell is. A Liberty formátum tehát ideális tárolási lehetőséget biztosít a hőmérsékletfüggő késleltetés függvényeknek.

3. Az analóg szimulációk végrehajtása után a késleltetés-hőmérséklet függvényeket el kell tárolni a Liberty adatbázisban. A logi-termikus szimulátor ebből az adatbázisból fogja előhívni az adott cellákhoz és hőmérsékletekhez tartozó késleltetéseket minden termikus időlépésben (Δt).
4. A logi-termikus szimuláció minden Δt termikus lépésben meghatározza a kialakult hőmérséklet-eloszlást az áramkörön. A hőmérséklet értékekből az eltárolt késleltetés-hőmérséklet függvények segítségével frissíti minden cella minden késleltetését. Ezeket a lépéseket addig ismétli a logi-termikus szimulátor, amíg a megadott szimulációs időt el nem éri.

A klasszikus SDF késleltetési modell esetében csak három hőmérsékleti sarokpontra tudunk késleltetéseket meghatározni. A három pontot a továbbiakban *minimális*, *tipikus* és *maximális* nevekkkel jelölöm. Az SDF modellel tehát a három ismert hőmérsékleti pont között lineáris interpolációval lehet meghatározni a késleltetéseket. A szélső pontokon túl extrapolációt lehet alkalmazni. Az interpoláció miatt a felhasznált késleltetési értékek jelentősen eltérhetnek a tényleges késleltetésektől az adott hőmérsékleten. A két modell közötti különbséget szemlélteti a 7.3. ábra.

Az SDF_{min} , SDF_{tip} , SDF_{max} pontok a *minimális*, *tipikus* és *maximális* sarokpontokhoz tartozó SDF késleltetéseket jelölik. Ezekben a pontokban az SDF modell értékei megegyeznek a nagy felbontású hőmérsékletfüggő késleltetés függvény értékével. A nagy felbontású karakterizációt a $T_{min}-T_{max}$ hőmérsékletek között végezzük. Látható, hogy az általam kidolgozott időzítési modell jóval részletesebben írja le a cellák hőmérséklettől függő késleltetéseit. A három sarokpontos közelítéssel a klasszikus SDF késleltetési modell jelentős hibát tud okozni a

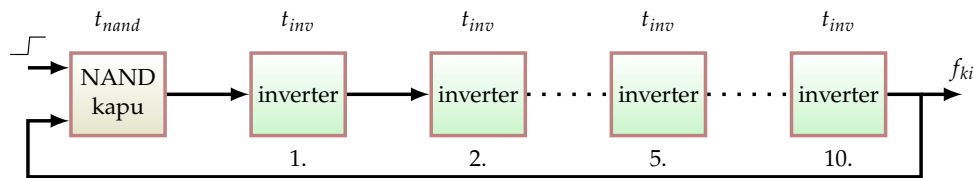


7.3. ábra. Az új és a klasszikus SDF modell összevetése

logi-termikus szimulációban a késleltetések frissítése során. Az általam bemutatott új modell segítségével három sarokpont helyett tetszőleges számú hőmérséklet értékre lehet meghatározni a késleltetéseket.

7.3. Az új modell bemutatása egy példán

Az általam javasolt hőmérsékletfüggő időzítési modellt egy egyszerű gyűrűs oszcillátor áramkörön mutatom be. Az áramkör kialakítása hasonló a 6.4. szakaszban bemutatott oszcillátor áramkörhöz a számláló funkciót ellátó puffer cellák nélkül. Az oszcillátor elvi sémáját a 7.4. ábra mutatja be. Az oszcillátor 10 inverter cellát és egy indító NAND cellát tartalmaz.



7.4. ábra. A gyűrűs oszcillátor sematikus vázlata

A gyűrűs oszcillátor kimeneti frekvenciája (7.1) alapján számítható, ha az inverterek késleltetését egyformának vesszük.

$$f_{ki} = \frac{1}{2 \cdot (t_{nand} + 10 \cdot t_{inv})}, \quad (7.1)$$

ahol t_{nand} a NAND kapu késleltetése, t_{inv} az inverterek késleltetése és f_{ki} az oszcillátor kimeneti frekvenciája.

Az általam ismertetett modellt a hagyományos SDF adatbázisból lineáris interpolációval meghatározott késleltetési adatokkal hasonlítom össze. Mindkét modellel ugyanazt a logi-termikus szimuláció elvégzve bizonyítom be az általam javasolt modell előnyét a hagyományossal

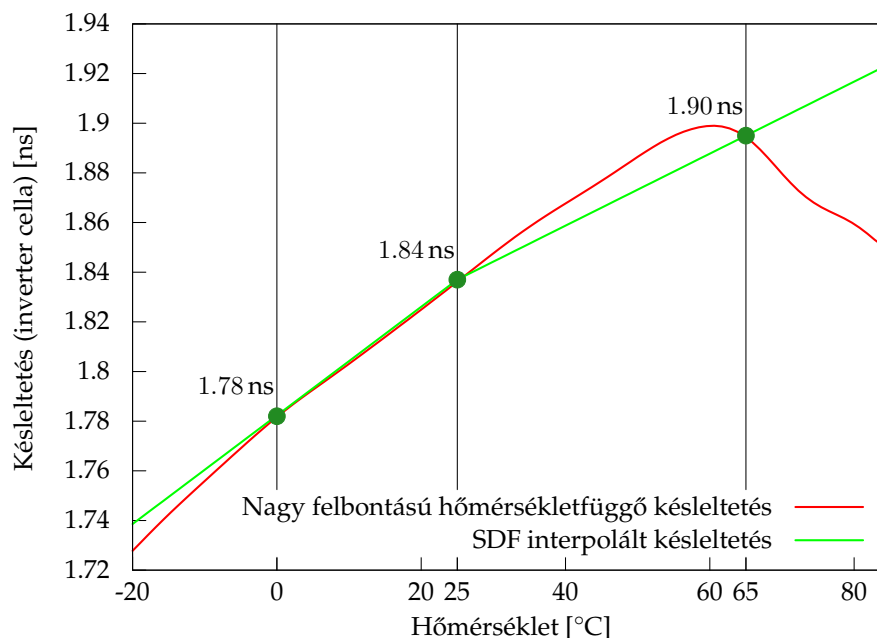
szemben. Az egyszerűség és az érthetőség kedvéért a bemeneti meredekségek és a terhelő kapacitások értékét a szimulációk során rögzítettem egy átlagos értéken, így a továbbiakban ezeket a paramétereket állandónak tekintem.

A cellák késleltetéseinek hőmérsékletfüggését leíró késleltetés-hőmérséklet függvényeket a szimuláció előtt kell előállítani. A függvények előállítása egy időben történhet a cellakönyvtár időzítési és teljesítmény karakterizációjával, amit rendszerint az integrált áramkör gyártó végez el. A fejlesztők ezeket az adatbázisokat készen kapják a tervezői eszközkészlet (Process Design Kit, PDK) részeként. Javaslatom szerint az áramkörgyártó cégek a hőmérsékletfüggő időzítési függvényeket karakterizálva és a Liberty formátumba integrálva biztosíthatják az áramkör tervezőknek.

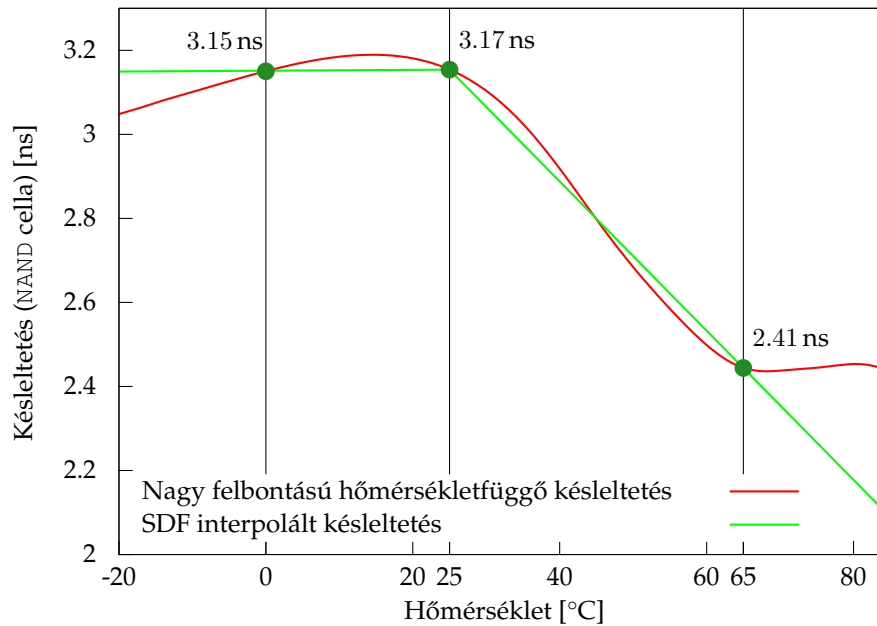
A karakterizációt SPICE szimulációk segítségével lehet elvégezni. A cellák késleltetés-hőmérséklet függvényeinek meghatározása egy SPICE kompatibilis analóg szimulátorban a környezeti hőmérséklet pásztázásával állítható elő. A SPICE szimulátor minden környezeti hőmérsékleti értékre meghatározza a vizsgálat alatt lévő cella bemeneti és kimeneti lábai közötti késleltetést. A környezeti hőmérsékletet tetszőleges tartományban és felbontással lehet állítani. A SPICE szimulációk során egyszerre egy cellát karakterizálunk, így a környezeti hőmérséklet változtatásával lehet imitálni a cella hőmérsékletének változását.

7.4. Késleltetés-hőmérséklet függvények

Az általam bemutatott példában a környezeti hőmérséklet változtatását $-20-85^{\circ}\text{C}$ tartományban végeztem el. A karakterizációt elegendő volt az áramkörben szereplő cellákra végrehajtani. Az inverter és NAND cellák hőmérsékletfüggő késleltetési görbéit mutatja a 7.5. és a 7.6. ábra. Ezekhez hasonló késleltetés-hőmérséklet görbéket mutat be [J2] is.



7.5. ábra. Az inverter cella késleltetés-hőmérséklet függvényei



7.6. ábra. A NAND cella késleltetés-hőmérséklet függvényei

A SPICE szimulációkkal kinyert késleltetéseket és kapcsolási energiák értékeit az 5.4. szakaszban ismertetett eljárás használatával 100-szoros skálafaktoral megnöveltem. Az általam javasolt modell és a hagyományos SDF késleltetési modell összehasonlítását a skálázás nem befolyásolja, viszont segítségével a gigahertzes frekvencia tartományban működő áramkör szimulációját töredék idő alatt lehet végrehajtani.

A 7.5. és a 7.6. ábrákon bejelöltem a hagyományos SDF modell által reprezentált hőmérsékletfüggő késleltetéseket is. Az SDF adatbázisban a késleltetési értékek 0 °C, 25 °C és 65 °C hőmérsékletekre vannak karakterizálva. A köztes hőmérsékleti értékekhez tartozó késleltetés értékeket lineáris interpolációval lehet meghatározni. Az SDF adatbázis késleltetési értékeit szintén 100-szoros skálafaktoral növeltem meg a gyorsítás és az összehasonlíthatóság érdekében.

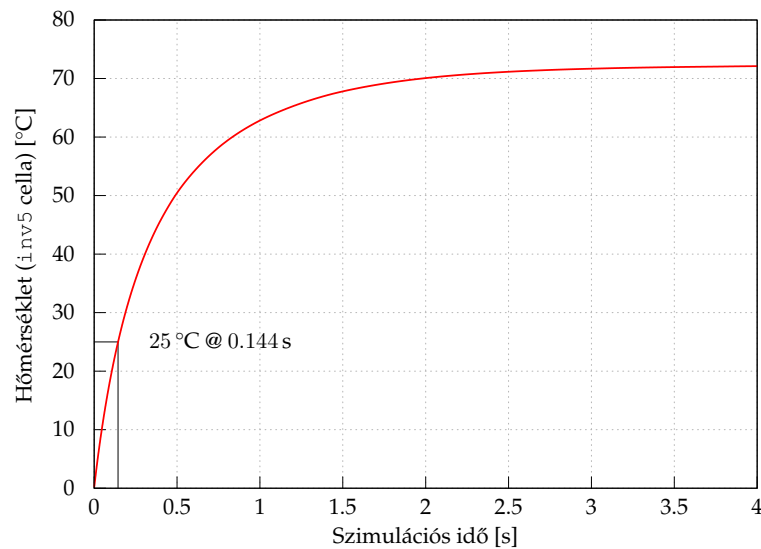
Észrevehető a 7.5. és a 7.6. ábrákon, hogy a lineáris interpolációval készült szakaszonként lineáris függvények a tényleges, SPICE szimulációkkal számított késleltetésektől a karakterizációs pontoktól távolodva a széleknél egyre nagyobb eltérést mutatnak. Ezek az eltérések hibát okoznak az áramkör hőmérsékletfüggő késleltetéseinek logi-termikus szimulációjában.

7.5. Logi-termikus szimuláció SDF késleltetések alapján

A demonstrációs áramkörön mind a hagyományos SDF modellel, mind az általam kidolgozott új modellel lefuttattam a logi-termikus szimulációt. A szimulációt mindkét esetben 4 másodperces szimulációs időig végeztem. Ekkorra az áramkör már mindkét esetben megközelítette az állandósult hőmérsékletet.

Első lépésként a hagyományos SDF időzítésekkel futtattam le a logi-termikus szimulációt. A tranziens hőmérsékleti görbét az `inv5` cellára a 7.7. ábra mutatja. A bemutatott demonstrációs áramkörben az egyes cellák között kialakuló hőmérsékleti különbség elhanyagolható a teljes

áramkör hőmérsékletéhez képest, így a továbbiakban az *inv5* cella hőmérsékletét veszem alapul minden hőmérsékletfüggést bemutató ábrán.



7.7. ábra. Az *inv5* cella hőmérsékleti tranziense

A 7.7. ábrán megfigyelhető, hogy az áramkör több mint 70 °C-ra melegszik fel 0 °C környezeti hőmérséklet esetében. A 7.5. és a 7.6. görbékét megfigyelve látható, hogy a 65 °C fölötti tartományon a hagyományos SDF modell késleltetési értékei már jelentősen eltérnek a tényleges késleltetési értékektől.

A 7.7. ábrán bejelöltem azt az időt, amikor az áramkör *inv5* cellája eléri a 25 °C hőmérsékletet. Ez a pont a görbén azért érdekes, mert az SDF interpolált szakaszonként lineáris függvénynek itt van az egyetlen töréspontja¹, tehát itt számíthatunk a hagyományos SDF modell esetén a szimulációs eredmények ugrásszerű változására. Az *inv5* cella 0,144 s eltelte után éri el a 25 °C hőmérsékletet.

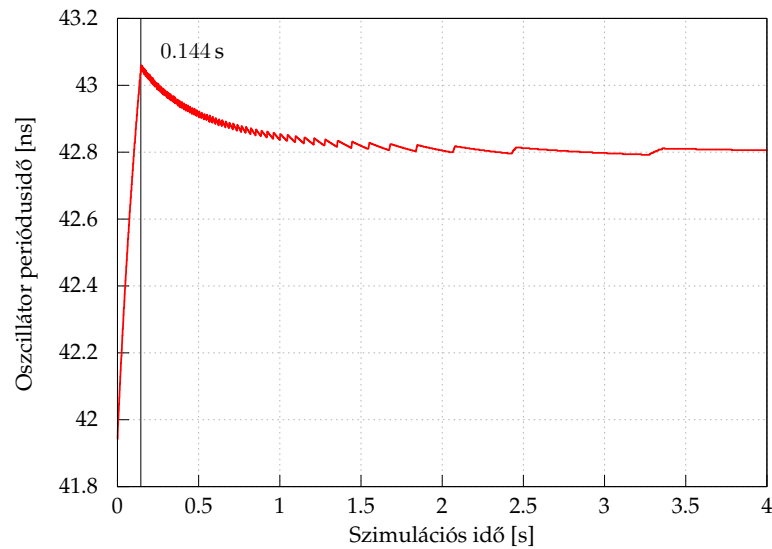
A hagyományos SDF modellel történt szimuláció eredményét mutatja a 7.8. ábra. A 7.8. ábrán a gyűrűs oszcillátor kimenetén mért oszcilláció periódusideje látható a szimulációs idő függvényében. Az egyszerűség és összehasonlíthatóság kedvéért csak a periódusidőt tüntettem fel a diagramokon, a frekvencia értékek a periódusidő reciprokával egyenlőek.

A 7.8. ábrán megfigyelhető, hogy az oszcillátor periódusideje 0,144 s-nál ugrásszerűen megváltozik. Ennek oka az SDF modellben található töréspont 25 °C hőmérsékleten. A 7.8. ábrából is látható, hogy az SDF modellel történő szimuláció esetében a késleltetés-hőmérséklet görbék lineáris közelítése jelentős hibákat okozhat.

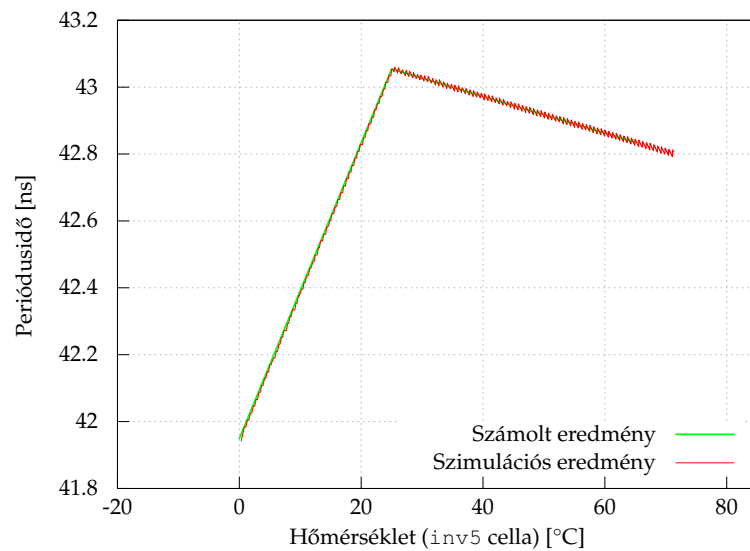
Az oszcillátor periódusidejét ábrázolja a 7.9. ábra az *inv5* cella hőmérsékletének függvényében. A 7.9. ábrán szembetűnő a görbék lineáris jellege, ami nyilvánvaló az SDF késleltetés-hőmérséklet függvények lineáris volta miatt. A 7.9. ábrán két függvény figyelhető meg. A *szimulációs eredmény* görbéje mutatja a szimuláció során meghatározott periódusidőt.

A szimulációs eredmény helyességének ellenőrzésére analitikusan is meghatároztam a

¹ A lineáris interpolációt három pont között hajtjuk végre a hagyományos SDF modell esetében, ezért a szakaszonként lineáris függvény töréspontja a középső értéknél lesz. Az első és harmadik interpolációs pont által meghatározott tartományon kívül extrapolációt alkalmazunk.



7.8. ábra. SDF késleltetésekkel mért periódusidő

7.9. ábra. Periódusidő az *inv5* cella hőmérsékletének függvényében

hőmérsékletfüggő periódusidő függvényt. Az SDF modell szakaszonként lineáris függvényeiből meghatározható a ring oszcillátor eredő kimeneti periódusideje. Az SDF adatbázisban az egyszerűség kedvéért az inverter cellák késleltetés görbéi megegyeznek. Adott még ezen kívül a NAND cella késleltetés szakaszonként lineáris függvénye² az SDF adatbázisban. Ezekkel az adatokkal könnyen meghatározható analitikusan is az eredő periódusidő (7.2) segítségével.

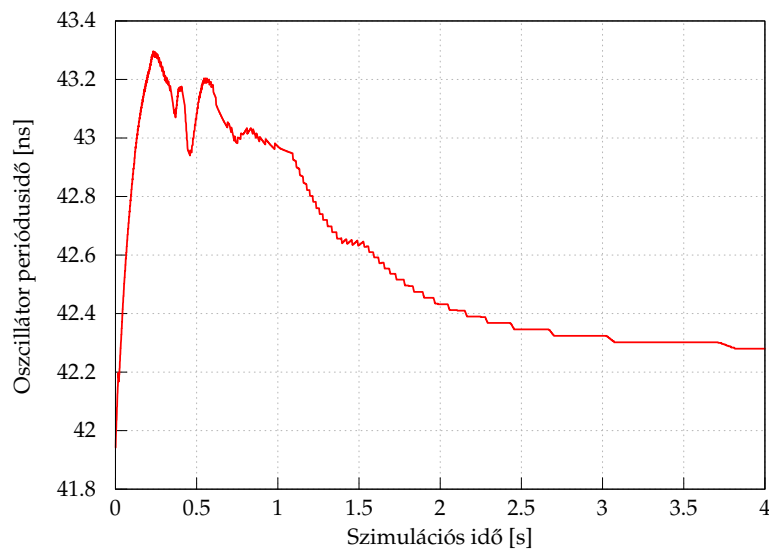
$$t_{\text{periódus}} = 2 \cdot (t_{\text{nand}} + 10 \cdot t_{\text{inv}}), \quad (7.2)$$

² A NAND cella két bemenete és egy kimenete közötti késleltetések a CMOS kialakítás miatt jó közelítéssel egyformának tekinthetők.

ahol t_{nand} a NAND kapu késleltetése és t_{inv} az inverterek késleltetése. A 7.9. ábrán látható, hogy a szimuláció eredménye megegyezik az analitikus számítás eredményével.

7.6. Logi-termikus szimuláció nagy felbontású hőmérsékletfüggő késleltetésekkel

A 7.5. szakaszban bemutatott logi-termikus szimulációt elvégeztem az általam kidolgozott új hőmérsékletfüggő késleltetési modellel is. Az oszcillátor kimeneti periódusidejét a szimulációs idő függvényében a 7.10. ábra mutatja be.



7.10. ábra. Nagy felbontású késleltetés-hőmérséklet görbékkel szimulált periódusidő-idő függvény

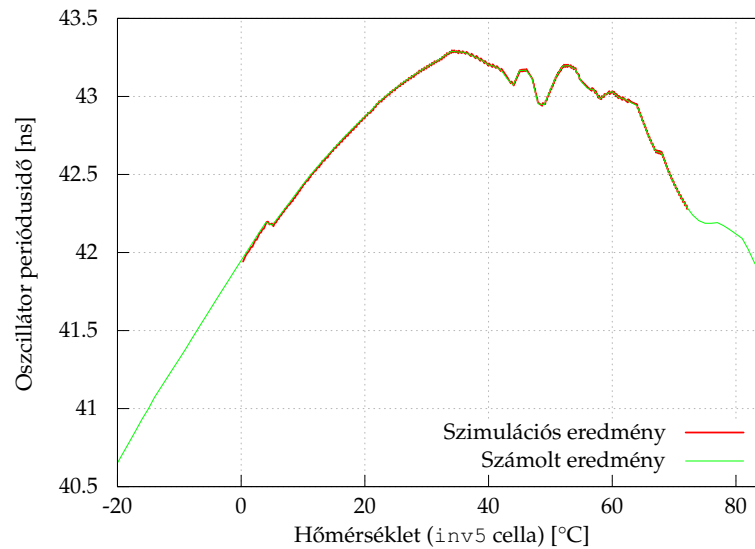
Megfigyelhető a 7.10. ábrán, hogy a periódusidő változása a 7.5. szakaszban bemutatott SDF késleltetésekkel szimulált eredményekhez képest sokkal részletesebb[C8].

Az eredő kimeneti periódusidő függvényt meghatároztam az `inv5` cella hőmérsékletének függvényében is, melyet a 7.11. ábra mutat.

Az általam javasolt hőmérsékletfüggő időzítési modellel is végrehajtottam a periódusidők analitikus számítását a (7.2) képlet segítségével. A 7.11. ábra két görbéje a szimulált és az analitikusan számított görbéket ábrázolja. Látható, hogy a számítás és a szimuláció eredményei itt is megegyeznek. Megfigyelhető a 7.11. ábrán az is, hogy a szimuláció görbéje a 0–+72 °C tartományban értelmezett, mivel ekkora hőmérsékletre melegedett fel az áramkör a logi-termikus szimuláció során. Az analitikusan számított görbe a karakterizáció –20–85 °C tartománya miatt szintén a –20–85 °C tartományba esik.

7.7. A hagyományos és az új modell összehasonlítása

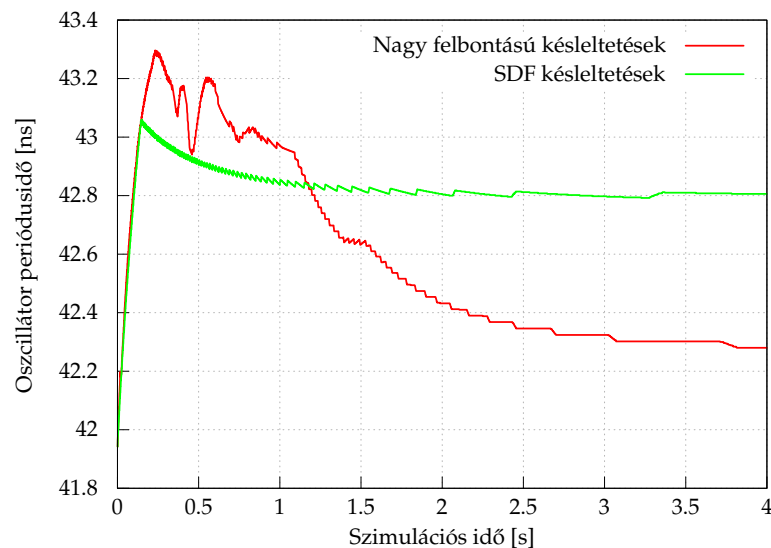
A két modellel elvégzett logi-termikus szimulációk eredményét összevetve kijelenthető, hogy az általam kidolgozott nagy felbontású hőmérsékletfüggő késleltetések alkalmazásával



7.11. ábra. Nagy felbontású késleltetésekkel szimulált periódusidő-hőmérséklet függvény

az eddiginél pontosabb logi-termikus szimulációk hajthatók végre. Az időzíti értékek hőmérsékletfüggésének pontos karakterizációja a logi-termikus szimulációk előtt tehát fontos lépés. A klasszikus SDF interpolált késleltetésekkel számolt és az általam kidolgozott új modellel történő logi-termikus szimulációk között jelentős felbontásbeli különbség van, ami a szimulációk pontosságát befolyásolja.

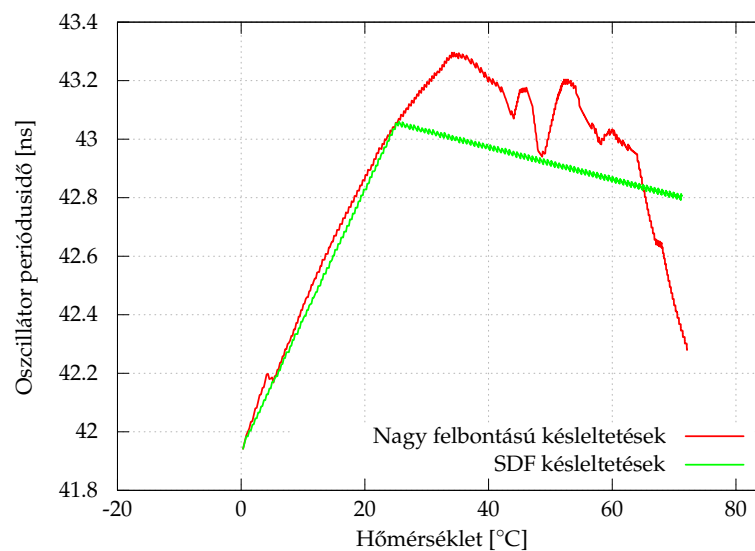
A két modell közötti különbségeket a szimulációs eredmények közös diagramban történő ábrázolásával lehet bemutatni[C8]. Ez látható a 7.12. ábrán.



7.12. ábra. A klasszikus és az új modellek összehasonlítása az idő függvényében

Megfigyelhető a 7.12. ábrán, hogy a két modell felbontása és pontossága között jelentős különbség van. A pontos hőmérsékletfüggő késleltetésekkel végzett szimuláció esetében a periódusidő görbe nem változik meg ugrásszerűen a 25 °C hőmérséklet elérésével. A pontos

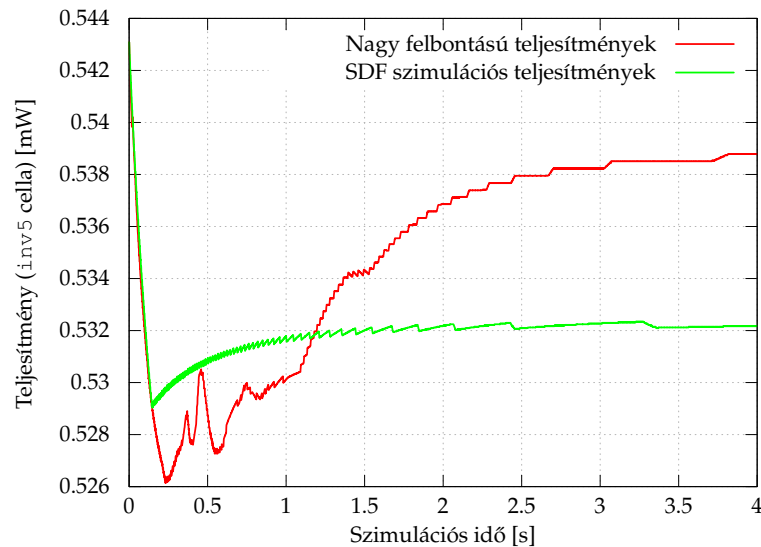
késleltetésekkel mért görbe jellegét a 10 inverter és a NAND cellák együttesen határozzák meg. A NAND cella késleltetési értékei nagyobbak az inverterek késleltetésénél (7.5. és 7.6. ábrák), viszont a 10 inverterből álló inverterlánc együttes késleltetése domináns. Az inverter és NAND celláknál kimutatható az inverz hőmérsékletfüggés jelensége is [52, 53, 54, 55], melyet a 6.1. szakaszban fejtek ki bővebben. A jelenség miatt előfordulhat, hogy a hőmérsékletek növekedésével a cella késleltetése lecsökken, tehát a működés felgyorsul. Ez a jelenség figyelhető meg a pontos hőmérsékletfüggő késleltetésekkel végzett szimulációs eredményeken is. Az oszcillátor periódusideje a hőmérséklet növelésével egy ideig növekszik, tehát lassul a működés, majd 35 °C felett csökken a periódusidő, tehát gyorsabban működik az eszköz. Az állandósult állapot hőmérsékletén végeredményben az áramkör működése lassabb lesz a kezdeti állapot sebességénél, de a két szélsőérték közötti periódusidő (vagy frekvencia) változását a klasszikus egyenesekkel közelített késleltetési modellekkel (SDF) nem tudjuk pontosan kimutatni. A 7.13. ábrán látható a periódusidő az `inv5` cella hőmérsékletének függvényében.



7.13. ábra. A klasszikus és az új modellek összehasonlítása a hőmérséklet függvényében

A cellák késleltetéseinek hőmérsékletfüggő változása az egységnyi idő alatt bekövetkezett fogyasztást is módosítja, hiszen a késleltetések mértékétől függ, hogy adott idő alatt a kapcsolási aktivitás milyen mértékű, tehát mennyi az áramkör dinamikus fogyasztása. Az általam bemutatott hőmérsékletfüggő késleltetési modell segítségével a késleltetések változása által okozott fogyasztásbeli különbségek is kimutathatóak. A logi-termikus szimuláció miatt ezek a fogyasztásban történt változások implicit módon szimulálhatóak és a teljesítmény-változás következtében kialakuló hőmérsékleti különbségek is felderíthetőek. A két modell fogyasztásbeli eltéréseit mutatja be a 7.14. ábra.

Megfigyelhető a 7.14. ábrán, hogy a két modellel végrehajtott logi-termikus szimulációk eltérése az áramkör által fogyasztott teljesítmények különbségében is megmutatkozik.



7.14. ábra. Fogyasztási különbségek a két modell között

7.8. A Liberty formátum bővítése

A pontos hőmérsékletfüggő késleltetés függvényeket minden cellához tárolni kell valamilyen formában. A Liberty nyílt forráskódú formátum lehetővé teszi felhasználó által definiált adattípusok létrehozását valamint alkalmas többdimenziós függvények tárolására[45]. A Liberty adatbázisból kinyerhető időzítés, teljesítmény és zaj adatok mind táblázatos formában szerepelnek. Az adatbázist feldolgozó értelmező ezeket az adatokat elérhetővé teszi egy programozói interfész (API) segítségével, melyet saját alkalmazásunkban felhasználhatunk. A CellTherm alkalmazás ezt a feldolgozó modult beépítve tartalmazza, így tudja kinyerni a fogyasztási adatokat a Liberty adatbázisból. Javaslatom szerint a hőmérsékletfüggő késleltetés függvények kompakt formában szintén eltárolhatóak ugyanebben a Liberty adatbázisban, a többi adat mellett. A függvények Liberty adatbázisban történő eltárolásával a hőmérsékletfüggő késleltetés függvények minden cellához a meglévő, beépített eszközkészlettel kinyerhetők, mint a teljesítmények esetében. A 7.1. kódrészlet egy ilyen bővített Liberty fájlt mutat be.

A Liberty formátumban az új függvénytípusok tárolására új tárolócsoporthoz és attribútumokat kell definiálni. Erre azért van szükség, mert a nyelv szabályrendszerébe nincs beépítve expliciten a hőmérsékletfüggő görbék leírásának lehetősége. Az új típusok és attribútumok definíciójával a nyelvi szabályoknak is eleget teszünk, és az értelmező ezeket az új típusokat is értelmezni tudja a feldolgozás során.

A `temper_template` csoport egy olyan sablont definiál, melyet egységesen fel lehet használni különböző hőmérsékletfüggő függvények leírására. A sablon definiálja a hőmérsékletfüggő görbék értelmezési tartományát, tehát a hőmérsékleti mintavételi tartományt. Ahány értéket definiálunk az `index_1` mezőben, annyi hőmérséklet értékhez kell megadni a függvényértékeket. A példa esetében az értelmezési tartomány $-20-85^{\circ}\text{C}$ fokenkénti felbontással, tehát 106 érték szerepel az `index_1` mezőben.

A `temperature_dep_delay` csoportban ezekhez a hőmérsékletekhez határozzuk meg a késleltetési értékeket. A `t_temp1` azonosítóval határozzuk meg a felhasználni kívánt sablont. A `values` csoport tartalmazza a 106 függvényértéket.

```

library(example) {
    define_group("temper_template", "library");
    define_group("temperature_dep_delay", "timing");
    define("variable_1", "temper_template", "string");
5    define_group("index_1", "temper_template");
    define_group("values", "temperature_dep_delay");

    technology(cmos);

10    temper_template(t_templ) {
        variable_1 : temperature;
        index_1 (" -20, -19, -18, -17, -16, \
/* ... */
        75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85") {}
15    }
    cell(inv01) {
        area : 2;
        pin(A) {
            direction : input;
            capacitance : 1;
20        }
        pin(Y) {
            direction : output;
            function : "!A";
25            timing () {
                intrinsic_rise : 0.49;
                intrinsic_fall : 0.77;
                rise_resistance : 0.1443;
                fall_resistance : 0.0523;
30                related_pin : "A";
/* ... */

            temperature_dep_delay(t_templ) {
                values (\
35                "1.727783e-9, 1.730767e-9, 1.733713e-9, 1.736751e-9, 1.739880e-9, \
                1.742603e-9, 1.745300e-9, 1.747969e-9, 1.750661e-9, 1.753310e-9, 1.755928e-9, \
                1.858606e-9, 1.855467e-9, 1.851824e-9, 1.849510e-9, 1.849963e-9") {}
            }
40        }
    }
}

```

7.1. kódrészlet. Bővített Liberty formátum

A Liberty formátum bővítésére bemutatott példa leegyszerűsítve mutatja be a formátum testre szabhatóságát. A példában az érthetőség és egyszerűség kedvéért a hőmérsékletfüggő késleltetési értékeket csak a hőmérséklettel paraméterezve mutattam be. A SPICE szimulációkkal meghatározott késleltetési értékek azonban különböző bemeneti jelmeredekségek és terhelő kapacitások esetében szintén különbözőek lesznek. A Liberty formátum lehetővé teszi többdimenziós függvények hasonló módon való tárolását. Így a hőmérsékletfüggő késleltetési görbéket négy dimenziós függvényekként lehet eltárolni, ahol a négy dimenzió a bemeneti jelmeredekség, a kimeneti terhelő kapacitás, a hőmérséklet és a késleltetés.

Az általam bemutatott bővített Liberty formátum hasonlóképpen felhasználható a teljesítmény-görbék hőmérséklettel történő paraméterezésére is. A többdimenziós függvények tárolására mutat példát a B. függelékben a B.1. kódrészlet.

8. fejezet

Összefoglalás

Értekezésemben modern, nanoméretű mikroelektronikai sztenderd cellás digitális integrált áramkörök logikai és termikus együttes szimulációjára dolgoztam ki modellt és eljárást [J3],[C1, C2, C3].

A termikus szimulátorok egyik alapvető bemeneti paramétere az áramkörben disszipált *teljesítmény*, amely a melegedést okozza. A bemenő teljesítmények ismeretében egy termikus szimulátor segítségével meghatározható egy áramkör felületén a hőmérséklet eloszlás és a kiugróan magas hőmérsékletű területek (hot-spotok) felderíthetők.

A logikai szimulátorok segítségével a digitális sztenderd cellás áramkörök működése szimulálható. A funkcionális működés ellenőrzése mellett az áramkör időzítési viszonyai is meghatározhatóak. Az időzítési vizsgálatok segítségével már a tervezés fázisában kimutathatóak a különböző időzítések miatt bekövetkező hibák, hazárdok.

Kutatásom során olyan szimulációs eszközt dolgoztam ki, amely egyesíti a logikai és termikus szimulátorok tulajdonságait és melynek segítségével lehetővé válik a logikai szimuláció a hőmérsékleti hatások figyelembe vételével. Az általam kidolgozott szimulációs eszközt úgy alkottam meg, hogy az szabványos illesztőfelületeken keresztül épüljön be az elektronikai tervező rendszerekbe (EDA) [J1, J3, J4],[C4, C5, C6].

A logikai szimulátorok segítségével időzítési analízis, hazárdok, időzítési hibák felderítése lehetséges a digitális áramkörökben. Az időzítési, késleltetési paraméterek hőmérsékletfüggése miatt az általam kidolgozott eljárás segítségével a logikai szimulációkba hőmérsékletfüggő késleltetések formájában vihető be a változó hőmérsékletek hatása. A logikai szimulátor az időzítések elcsúszásából, változásából adódó hibákat, hazárdokat képes detektálni és megjeleníteni, így a megváltozott hőmérsékletek miatt eltolódott késleltetések hatásaként keletkezett hazárdokat is képes kimutatni. A hőmérsékletfüggő késleltetések használatával a logikai szimuláció logi-termikus szimulációvá bővíthető [J2, J4],[C6, C7].

Az általam bemutatott nagy hőmérsékleti felbontású késleltetés görbék segítségével olyan hatások is modellezhetők, mint például az inverz hőmérsékletfüggés. Az inverz hőmérsékletfüggés jelensége nanométeres csíkszélességű áramköröknél jelentkezik és háttérben a töltéshordozók mobilitásának és a küszöb feszültség hőmérsékletfüggése áll. Inverz hőmérsékletfüggés esetén a növekvő hőmérséklettel az eszköz gyorsabban működik. Ezt a jelenséget a hagyományos időzítési modellek nem, vagy csak korlátozottan tudják leírni [C8].

A logi-termikus szimuláció gyorsítására két módszert is kidolgoztam, melyeket egymástól függetlenül és együttesen is fel lehet használni. A particionálás segítségével a termikus kompakt

modell mérete csökkenthető, és így a kialakuló hőmérsékletek számítása gyorsítható [J2],[C7]. A rendszer késleltetéseinek és a cellák disszipációjának arányos skálázásával a logikai szimuláció gyorsítható [C8].

Az általam kidolgozott új időzítési modellt és logi-termikus szimulációs eljárást ipari méretű és bonyolultságú nanométeres integrált áramkörök szimulációjára használták fel az európai FP7-es keretrendszerbeli *THERMINATOR* integrált projektben.

8.1. Az eredmények gyakorlati alkalmazásai

Kutatói munkám eredményeit a CellTherm alkalmazásban implementáltam, melyet a THERMINATOR projekt keretében az STMicroelectronics áramkörtervező és gyártó cég igényeinek megfelelően dolgoztam ki. Az STMicroelectronics a megvalósult implementációt a projekt lezártaival megkapta és a továbbiakban saját ipari áramköreinek logi-termikus vizsgálatára használja fel. A THERMINATOR projekt 2012 decemberében fejeződött be és a nemzetközi bíráló bizottság a projektet sikeresnek értékelte. Kutatásom új eredményeit tetszőleges digitális sztenderd cellás integrált áramkör szimulációjában fel lehet használni. Az áramkör változó termikus viselkedése miatt esetlegesen fellépő hazardjelenségeket kutatásom eredményeivel ki lehet mutatni és javaslatot lehet tenni ezek kivédésére. Munkám további lehetséges alkalmazása az elhelyező és huzalozó algoritmusok kiegészítése termikus kényszerekkel, melynek segítségével már az elhelyezés és huzalozás lépésben a felületi topológiát a kritikus termikus kölcsönhatások minimalizálásával lehetne kialakítani.

Köszönetnyilvánítás

Szeretnék köszönetet mondani Dr. Rencz Márta professzor asszonynak, témavezetőmnek, aki motivált a dolgozat elkészítésében. Köszönöm, hogy az értekezés számtalan elolvasásával, korrekciójával és megjegyzéseivel segítette a dolgozat színvonalának növelését.

Köszönöm Dr. Székely Vladimírnek a sok segítséget és a hasznos megbeszéléseket, melyek lényegesen hozzájárultak munkám eredményességéhez. Közös beszélgetéseinken rendkívül sokat tanultam a termikus szimulációkról és méréstechnikáról, amiért hálás vagyok. Köszönöm a termikus szimulációkhoz írt forráskódokat, melyeket felhasználtam a logi-termikus szimulátor implementációjában.

Szeretnék köszönetet mondani minden kollégámnak a Budapesti Műszaki és Gazdaságtudományi Egyetem Elektronikus Eszközök Tanszékén, akik munkámban támogattak és tanácsokkal láttak el. Külön szeretném megköszönni Nagy Gergelynek, Szalai Albinnak, Horváth Péternek, Ender Ferencnek, Dr. Szabó Péternek és Dr. Czirkos Zoltánnak a hasznos szakmai tapasztalat-cseréket, beszélgetéseket. Rendkívül sokat tanultam ezeken az eszmecseréken.

Kiemelten szeretném megköszönni Dr. Bognár György barátomnak a biztató szavakat, a tapintatos és emberséges motivációt, amiből mindig sok erőt meríthettem. Köszönöm, hogy remek pedagógusként mindig tudta mikor kell alkotásra buzdítani, dicsérettel motiválni és együttérzéssel bátorítani. Külön köszönöm, hogy idejét –akár éjszakáját is– feláldozva segített az értekezés átnézésében és javításában.

Szeretném megköszönni szüleimnek a feltétel nélküli szeretetet, bátorítást és támogatást, ami nélkül nem jutottam volna el idáig. Köszönöm, hogy mindig szeretettel gondoltak rám és tisztességes emberi értékrend szerint neveltek fel.

Külön köszönöm Sashegyi Szilviának a lelki támaszt, a rengeteg türelmet és biztatást, amellyel támogatott. Köszönöm a „mágikus” dobozt, amiből mindig bátorító és sokszor megható gondolatokat olvashattam és erőt meríthettem.

Köszönöm Makásnak, hogy beszélgetéseink hatására a nehézségeken és kihívásokon könnyebben átlendülhettem és pozitív irányba terelte gondolataimat az értekezéssel kapcsolatban.

A függelék

Adat-előkészítési idő szimuláció

```
`timescale 1ps / 100fs
`include "adk.v"

module delay(d, clk, q, qb);
5   input d, clk;
   output q, qb;

   wire d, clk, q, qb;
   wire buf_out;

10  buf02 b1(d,buf_out);
   dff i1(.D(buf_out), .CLK(clk), .Q(q), .QB(qb)) ;

endmodule
15

`timescale 1ps / 100fs
module test_delay;
   reg clk = 0;
20  reg data;
   wire q, qb;

   delay dut1(.d(data), .clk(clk), .q(q), .qb(qb) );

25  initial
   begin
       clk = 0;
       data = 0;

30  #100 data = 1;
       #1100 data = 0;
       #300 data = 1;
       #470 data = 0;
       #2480 data = 1;

35  end

   always #600 clk = ~clk;

endmodule
```

A.1. kódrészlet. DFF logikai időzítési szimuláció

B függelék

Liberty fájlformátum

```
library (c35_CORELIB) {
technology ( cmos ) ;

delay_model      : table_lookup;
5 date           : "13-MAR-2007" ;
revision         : 2.22 ;

time_unit        : "1ns" ;
leakage_power_unit : "1pW" ;
10 voltage_unit   : "1V" ;
pulling_resistance_unit : "1kohm" ;
current_unit     : "1uA" ;
capacitive_load_unit(1000.000,ff) ;

15 /* *** power tables are in energy units of 'pJ' *** */

power_lut_template(power_outputs_0) {
variable_1 : input_transition_time;
index_1("0.050,0.500,1.000,2.000");
20 variable_2 : total_output_net_capacitance;
index_2("0.001,0.005,0.020,0.080,0.160");
}
/*...*/
/*
25 -- Description      :    1 bit HALF_ADDER
*/
cell(ADD21) {
area : 145.6;
cell_footprint : "CORE_ADD2";
30 cell_leakage_power : 94.075 ;
/* ... */
pin(CO) {
direction : output;
max_capacitance : 0.32;
35 internal_power() {
rise_power(power_outputs_0) {
values( " 0.078, 0.054, -0.029, -0.358, -0.798",\
" 0.085, 0.061, -0.023, -0.352, -0.796",\
" 0.096, 0.072, -0.013, -0.344, -0.798",\
40 " 0.123, 0.097, 0.009, -0.323, -0.837");
}

fall_power(power_outputs_0) {
values( " 0.149, 0.168, 0.246, 0.570, 1.00",\
45 " 0.151, 0.170, 0.248, 0.572, 0.995",\
" 0.163, 0.182, 0.259, 0.582, 0.989",\
" 0.191, 0.208, 0.283, 0.603, 0.873");
}
}
}
/* ... */
```

```
50     timing() {  
        related_pin      : "A";  
        timing_sense      : positive_unate;  
        cell_rise(del_0_4_5) {  
55         values( " 0.177, 0.206, 0.293, 0.614, 1.036", \  
                  " 0.221, 0.247, 0.338, 0.657, 1.079", \  
                  " 0.240, 0.268, 0.357, 0.678, 1.099", \  
                  " 0.241, 0.269, 0.358, 0.679, 1.1");  
        }  
        rise_transition(del_0_4_5) {  
60         values( " 0.124, 0.174, 0.360, 1.142, 2.202", \  
                  " 0.125, 0.182, 0.367, 1.142, 2.202", \  
                  " 0.140, 0.189, 0.372, 1.148, 2.203", \  
                  " 0.161, 0.209, 0.390, 1.156, 2.209");  
        }  
65     }  
} /* cell(ADD21) */  
}
```

B.1. kódrészlet. Liberty fájlformátum

C függelék

Rövidítésjegyzék

ADK. <u>A</u> nalog- <u>M</u> ixed <u>S</u> ignal <u>D</u> esign <u>K</u> it	FPGA. <u>F</u> ield <u>P</u> rogrammable <u>G</u> ate <u>A</u> rray
ALU. <u>A</u> rithmetic <u>L</u> ogic <u>U</u> nit	GDSII. Felületi topológia (layout) leíró formátum
AMS. <u>A</u> nalog- <u>M</u> ixed <u>S</u> ignal/ <u>A</u> ustria <u>M</u> icro <u>S</u> ystems	HDL. <u>H</u> ardware <u>D</u> escription <u>L</u> anguage
ANSYS. Multidomén szimulátor	HTC. <u>H</u> eat <u>T</u> ransfer <u>C</u> oefficient
API. <u>A</u> pplication <u>P</u> rogramming <u>I</u> nterface	ITD. <u>I</u> nverse <u>T</u> emperature <u>D</u> ependence
ASIC. <u>A</u> pplication <u>S</u> pecific <u>I</u> ntegrated <u>C</u> ircuit	IEEE. <u>I</u> nstitute of <u>E</u> lectrical and <u>E</u> lectronics <u>E</u> ngineers
BME. <u>B</u> udapesti <u>M</u> űszaki <u>E</u> gyetem	LEF. Felületi topológia (layout) leíró formátum
CCS. <u>C</u> omposite <u>C</u> urrent <u>S</u> ource	LFSR. <u>L</u> inear <u>F</u> eedback <u>S</u> hift <u>R</u> egister
CMOS. <u>C</u> omplementary <u>M</u> etal <u>O</u> xide <u>S</u> emiconductor	LSB. <u>L</u> east <u>S</u> ignificant <u>B</u> it
COMSOL. Multidomén szimulátor	LVS. <u>L</u> ayout <u>V</u> ersus <u>S</u> chematic
CSV. <u>C</u> omma <u>S</u> eparated <u>V</u> alues	MOS. <u>M</u> etal <u>O</u> xide <u>S</u> emiconductor
CUDA. Párhuzamos programozói API (NVidia)	MSB. <u>M</u> ost <u>S</u> ignificant <u>B</u> it
DEF. Felületi topológia (layout) leíró formátum	MTA. <u>M</u> agyar <u>T</u> udományos <u>A</u> kadémia
DFF. <u>D</u> <u>F</u> lip-flop	NID. <u>N</u> etwork <u>I</u> dentification with <u>D</u> ecomposition
DRC. <u>D</u> esign <u>R</u> ules <u>C</u> heck	NLDM. <u>N</u> on- <u>L</u> inear <u>D</u> elay <u>M</u> odel
DSP. <u>D</u> igital <u>S</u> ignal <u>P</u> rocessing	NLPM. <u>N</u> on- <u>L</u> inear <u>P</u> ower <u>M</u> odel
DTAS. <u>D</u> ynamic <u>T</u> hermal <u>A</u> ware <u>S</u> cheduling	NXP. Integrált áramkörgyártó társaság
DTM. <u>D</u> ynamic <u>T</u> hermal <u>M</u> anagement	NYHL. <u>N</u> yomtatott <u>H</u> uzalozású <u>L</u> emez
DVFS. <u>D</u> ynamic <u>V</u> oltage and <u>F</u> requency <u>S</u> caling	OASIS. <u>O</u> pen <u>A</u> rtwork <u>S</u> ystem <u>I</u> nterchange <u>S</u> tandard
ECSM. <u>E</u> ffective <u>C</u> urrent <u>S</u> ource <u>M</u> odel	PDK. <u>P</u> rocess <u>D</u> esign <u>K</u> it
EDA. <u>E</u> lectronic <u>D</u> esign <u>A</u> utomation	PLI. <u>P</u> rogramming <u>L</u> anguage <u>I</u> nterface
ELDO. Analóg áramkör szimulátor (Mentor Graphics)	POLITO. Politecnico di Torino
FDM. <u>F</u> inite <u>D</u> ifference <u>M</u> ethod	PVT. <u>P</u> rocess- <u>V</u> oltage- <u>T</u> emperature
FEM. <u>F</u> inite <u>E</u> lement <u>M</u> ethod	RTL. <u>R</u> egister <u>T</u> ransder <u>L</u> evel
FET. <u>F</u> ield <u>E</u> ffect <u>T</u> ransistor	SAIF. <u>S</u> witching <u>A</u> ctivity <u>I</u> nterchange <u>F</u> ormat
	SDF. <u>S</u> tandard <u>D</u> elay <u>F</u> ormat

SISSI. <u>S</u> imulator for <u>I</u> ntegrated <u>S</u> tructures by <u>S</u> imultaneous <u>I</u> teration	TLF. <u>T</u> iming <u>L</u> ibrary <u>F</u> ormat
SPDM. <u>S</u> calable <u>P</u> olynomial <u>D</u> elay <u>M</u> odel	TSMC. Integrált áramkörgyártó társaság (Taiwan)
SPICE. <u>S</u> imulation <u>P</u> rogram with <u>I</u> ntegrated <u>C</u> ircuit <u>E</u> mphasis	UNIBO. University of Bologna
SPPM. <u>S</u> calable <u>P</u> olynomial <u>P</u> ower <u>M</u> odel	VCD. <u>V</u> alue <u>C</u> hange <u>D</u> ump
SRPG. <u>S</u> tate <u>R</u> etention <u>P</u> ower <u>G</u> ating	VHDL. <u>V</u> HSIC <u>H</u> ardware <u>D</u> escription <u>L</u> anguage
TIM. <u>T</u> hermal <u>I</u> nterface <u>M</u> aterial	

Megjelent folyóiratcikkek

- [J1] A. Timar and M. Rencz. Real-time heating and power characterization of cells in standard cell designs. *MICROELECTRONICS JOURNAL*, 44(11):977–985, November 2013. URL: <http://dx.doi.org/10.1016/j.mejo.2012.03.002>, doi:10.1016/j.mejo.2012.03.002.
- [J2] A. Timar and M. Rencz. Temperature dependent timing in standard cell designs. *MICROELECTRONICS JOURNAL*, 2013. [Nyomtatásban, 2013. szept. 27.]. URL: <http://dx.doi.org/10.1016/j.mejo.2013.08.016>, doi:10.1016/j.mejo.2013.08.016.
- [J3] András Timár, György Bognár, András Poppe, and Márta Rencz. Electro-thermal co-simulation of ICs with runtime back-annotation capability. *INTERNATIONAL JOURNAL OF MICROELECTRONICS AND COMPUTER SCIENCE*, 1(3):287–292, 2010. URL: mycite.omikk.bme.hu/doc/101184.pdf.
- [J4] A. Timár and M. Rencz. Studying the influence of chip temperatures on timing integrity using improved power modeling. *JOURNAL OF LOW POWER ELECTRONICS*, 7(10):531–540, 2011. ScopusID: 84857269360. doi:10.1166/jolpe.2011.1153.
- [J5] A. Timár and M. Rencz. New accurate temperature dependent timing model in digital standard cell designs. *JOURNAL OF LOW POWER ELECTRONICS*, 9(4), December 2013. [Közlésre elfogadva, 2013. szept. 30.].

Konferenciakiadványban megjelent előadás

- [C1] Andras Timar and Marta Rencz. Acquiring real-time heating of cells in standard cell designs. In *Test Workshop (LATW), 2012 13th Latin American*, pages 121–125, Quito, Ecuador, April 2012. URL: <http://mycite.omikk.bme.hu/doc/134104.pdf>, doi: 10.1109/LATW.2012.6261249.
- [C2] András Timár, András Poppe, and Márta Rencz. A novel approach of logi-thermal simulation methodology and implementation for ASIC designs. In *Proceedings of the 17th International Conference Mixed Design of Integrated Circuits and Systems (MIXDES'10)*, pages 351–356, Wroclaw, Poland, June 2010. ScopusID: 77957263525. URL: <http://mycite.omikk.bme.hu/doc/102333.pdf>.
- [C3] A. Timár, Gy. Bognár, A. Poppe, and M. Rencz. Electro-thermal co-simulation of ICs with runtime back-annotation capability. In *Proceedings of the 16th International Workshop on THERMal INvestigation of ICs and Systems (THERMINIC'10)*, pages 183–188, Barcelona, Spain, October 2010. ScopusID: 78650463570. URL: <http://mycite.omikk.bme.hu/doc/89560.pdf>.
- [C4] A. Timár, Gy. Bognár, and M. Rencz. Improved power modeling in logi-thermal simulation. In *Proceedings of the 17th International Workshop on THERMal INvestigation of ICs and Systems (THERMINIC'11)*, pages 143–148, Paris, France, September 2011. ScopusID: 83655182705. URL: <http://mycite.omikk.bme.hu/doc/104528.pdf>.
- [C5] A. Timár and M. Rencz. Studying the influence of chip temperatures on timing integrity. In *Proceedings of the 12th IEEE Latin-American Test Workshop (LATW'11)*, pages 1–5, Porto de Galinhas, Brazil, March 2011. ScopusID: 80052628646. URL: <http://mycite.omikk.bme.hu/doc/102645.pdf>, doi:10.1109/LATW.2011.5985920.
- [C6] Gergely Nagy, András Timár, Albin Szalai, Márta Rencz, and András Poppe. New simulation approaches supporting temperature-aware design of digital ics. In *Proceedings of the 28th IEEE Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM'12)*, pages 313–318, San Jose, California, USA, March 2012. URL: <http://mycite.omikk.bme.hu/doc/131234.pdf>.
- [C7] A. Timar and M. Rencz. Temperature dependent timing in standard cell designs. In *Thermal Investigations of ICs and Systems (THERMINIC), 2012 18th International Workshop on*, pages 179–183, Budapest, Hungary, September 2012.

- [C8] A. Timar and M. Rencz. Logi-thermal simulation using accurate temperature dependent delay models. In *Thermal Investigations of ICs and Systems (THERMINIC), 2013 19th International Workshop on*, pages 376–380, Berlin, Germany, September 2013.
- [C9] G. Nagy, L. Pohl, A. Timar, and A. Poppe. Yield enhancement by logi-thermal simulation based testing. In *Thermal Investigations of ICs and Systems (THERMINIC), 2012 18th International Workshop on*, pages 196–199, Budapest, Hungary, September 2012.

Tézisekhez szorosan nem kapcsolódó közlemények

- [N1] Péter G. Szabó, György Bognár, Sándor Ress, András Timár, András Poppe, Márta Rencz, and Gyula Horváth. Hands-on teaching of microelectronics for yearly 1000 students at BME. In *Proceedings of the 9th European Workshop on Microelectronics Education (EWME'12)*, pages 127–130, Grenoble, France, May 2012. URL: <http://mycite.omikk.bme.hu/doc/132823.pdf>.
- [N2] Gy. Bognár and A. Timár. Cad eszközök az IC tervezésben az Elektronikus Eszközök Tanszékén. *ELEKTRONET*, 9(5):14–16, 2011.
- [N3] A. Timár and Gy. Bognár. Contactless characterization of MEMS devices using optical microscopy. In *Proceedings of the 12th IEEE workshop on Design & Diagnostics of Electronic Circuits & Systems (DDECS '09)*, pages 210–213, Liberec, Czech Republic, April 2009. UT: 000267364000045. URL: <http://mycite.omikk.bme.hu/doc/62897.pdf>, doi: 10.1109/DDECS.2009.5012130.
- [N4] A. Timár and Gy. Bognár. E-learning system and video tutorials in the microelectronics education. In *Proceedings of EWME 2008 - the 7th European Workshop on Microelectronics Education*, pages 150–151, Budapest, Hungary, May 2008.
- [N5] András Timár and Márta Rencz. Design issues of a low frequency low-pass filter for medical applications using Cmos technology. In P. Girard, A. Krasniewski, E. Gramatová, A. Pawlak, and T. Garbolino, editors, *Proceedings of the 10th IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems (DDECS'07)*, pages 325–328, Krakow, Poland, April 2007. UT: 000250200100056. doi:10.1109/DDECS.2007.4295304.
- [N6] A. Timár, Á. Vámos, and Gy. Bognár. Comprehensive design of a high frequency PLL synthesizer for ZigBee application. In *Proceedings of the 9th IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems (DDECS'06)*, pages 37–41, Prague, Czech Republic, April 2006. ScopusID: 33847145383. doi:10.1109/DDECS.2006.1649567.

Irodalomjegyzék

- [1] Kunquan Ma and Jing Liu. Liquid metal cooling in thermal management of computer chips. *Frontiers of Energy and Power Engineering in China*, 1(4):384–402, 2007. URL: <http://dx.doi.org/10.1007/s11708-007-0057-3>, doi:10.1007/s11708-007-0057-3.
- [2] Székely-Tarnay-Valkó. *Elektronikus eszközök I.* Műegyetemi Kiadó, 2000.
- [3] A. B. Kahng, S. Kang, R. Kumar, and J. Sartori. Enhancing the efficiency of energy-constrained DVFS designs. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, PP(99):1–1, 2012. doi:10.1109/TVLSI.2012.2219084.
- [4] A. Calimera, A. Macii, E. Macii, S. Rinaudo, and M. Poncino. Therminator: Modeling, control and management of thermal effects in electronic circuits of the future. In *Thermal Investigations of ICs and Systems (THERMINIC), 2010 16th International Workshop on*, pages 1–6, 2010.
- [5] F. Moschella, G. Gangemi, E. Macii, M. Rencz, and S. Rinaudo. The therminator project: Midway achievements and perspectives. In *Thermal Investigations of ICs and Systems (THERMINIC), 2011 17th International Workshop on*, pages 1–6, Sept. 2011.
- [6] Gordon E. Moore. Cramming more components onto integrated circuits, reprinted from electronics, volume 38, number 8, april 19, 1965, pp.114 ff. *Solid-State Circuits Newsletter, IEEE*, 11(5):33–35, sept. 2006. doi:10.1109/N-SSC.2006.4785860.
- [7] Shekhar Borkar and Andrew A. Chien. The future of microprocessors. *Commun. ACM*, 54(5):67–77, May 2011. URL: <http://doi.acm.org/10.1145/1941487.1941507>, doi:10.1145/1941487.1941507.
- [8] J.L. Shin, Dawei Huang, B. Petrick, Changku Hwang, A.S. Leon, and A. Strong. A 40nm 16-core 128-thread SPARC SoC processor. In *Solid State Circuits Conference (A-SSCC), 2010 IEEE Asian*, pages 1–4, nov. 2010. doi:10.1109/ASSCC.2010.5716541.
- [9] IEEE standard for verilog hardware description language. *IEEE Std 1364-2005 (Revision of IEEE Std 1364-2001)*, pages 1–560, 2006. doi:10.1109/IEEESTD.2006.99495.
- [10] IEEE standard VHDL language reference manual. *IEEE Std 1076-2008 (Revision of IEEE Std 1076-2002)*, pages 1–626, 26 2009. doi:10.1109/IEEESTD.2009.4772740.
- [11] Icarus verilog, 8th August, 10:57 2012. URL: <http://iverilog.icarus.com>.

- [12] Arunvel Thangamani and Sanjeeva Reddy. Managing heat with multiphysics. URL: <http://www.ansys.com/staticassets/ANSYS/staticassets/resource-library/article/AA-V3-I2-Managing-Heat-with-Multiphysics.pdf>.
- [13] B. Benbakhti, M. Rousseau, and J-C. De Jaeger, editors. *Implementation of an 2D Electro-Thermal Model for Power Semiconductor Devices Simulation: Application on Gallium Nitride*. URL: <http://www.comsol.com/papers/1187/download/Benbakhti.pdf>.
- [14] Mentor graphics announces flowmaster thermal analysis tool with new capabilities for meta-models, 3d response surface, and fast batch simulation, 7th March 2013. URL: <http://www.mentor.com/products/mechanical/news/mentor-announce-s-flowmaster-thermal-analysis-tool>.
- [15] R. Gillon, P. Joris, H. Oprins, B. Vandeveld, A. Srinivasan, and R. Chandra. Practical chip-centric electro-thermal simulations. In *Thermal Investigation of ICs and Systems, 2008. THERMINIC 2008. 14th International Workshop on*, pages 220–223, 2008. doi:10.1109/THERMINIC.2008.4669912.
- [16] V. Székely, A. Poppe, M. Rencz, M. Rosental, and T. Teszéri. THERMAN: a thermal simulation tool for IC chips, microstructures and PW boards. *Microelectronics Reliability*, 40(3):517 – 524, 2000. URL: <http://www.sciencedirect.com/science/article/pii/S0026271499002498>, doi:10.1016/S0026-2714(99)00249-8.
- [17] Székely Vladimír. *Integrált áramkörök elektro-termikus jelenségeinek modellezése*. Magyar Tudományos Akadémia, 1977. Kandidátusi értekezés.
- [18] Székely Vladimír. *Félvezető eszközök termikus problémái*. Magyar Tudományos Akadémia, 1989. Akadémiai doktori értekezés.
- [19] V. Székely, A. Páhi, A. Poppe, and M. Rencz. Electro-thermal simulation with the SISSI package. *Analog Integrated Circuits and Signal Processing*, 21(1):21–31, 1999. URL: <http://dx.doi.org/10.1023/A%3A1008371625945>, doi:10.1023/A:1008371625945.
- [20] V. Székely. THERMODEL: a tool for compact dynamic thermal model generation. *Microelectronics Journal*, 29(4–5):257 – 267, 1998. Thermal Investigations of ICs and Microstructures II. URL: <http://www.sciencedirect.com/science/article/pii/S0026269297000657>, doi:10.1016/S0026-2692(97)00065-7.
- [21] 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(1-2):49–59, 2001. URL: <http://dx.doi.org/10.1023/A%3A1011226213197>, doi:10.1023/A:1011226213197.
- [22] Imre László. *Villamos gépek és eszközök melegedése és hűtése*. Műszaki Könyvkiadó, 1982.
- [23] OpenMP Architecture Review Board. *OpenMP Application Program Interface*. OpenMP Architecture Review Board, July 2011. URL: <http://www.openmp.org/mp-documents/OpenMP3.1.pdf>.

- [24] CUDA parallel computing platform, 27th March, 16:13 2013. URL: http://www.nvidia.com/object/cuda_home_new.html.
- [25] OpenCL - the open standard for parallel programming of heterogeneous systems, 27th March, 16:13 2013. URL: <http://www.khronos.org/opencl>.
- [26] K. Skadron, M.R. Stan, Wei Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture. In *Computer Architecture, 2003. Proceedings. 30th Annual International Symposium on*, pages 2–13, June. doi:10.1109/ISCA.2003.1206984.
- [27] Wei Huang, M.R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy. Compact thermal modeling for temperature-aware design. In *Design Automation Conference, 2004. Proceedings. 41st*, pages 878–883, July.
- [28] David Brooks, Vivek Tiwari, and Margaret Martonosi. Wattch: a framework for architectural-level power analysis and optimizations. *SIGARCH Comput. Archit. News*, 28(2):83–94, May 2000. URL: <http://doi.acm.org/10.1145/342001.339657>, doi:10.1145/342001.339657.
- [29] David Doman. *Engineering the CMOS Library: Enhancing Digital Design Kits for Competitive Silicon*. John Wiley & Sons, Inc., June 2012. ISBN: 978-1-1182-4304-6.
- [30] George Mekhtarian. *Composite Current Source (CCS) Modeling Technology Background*. Synopsys, 2005.11 edition, November 2005.
- [31] S.D. Pable and Mohd. Hasan. Ultra-low-power signaling challenges for subthreshold global interconnects. *Integration, the VLSI Journal*, 45(2):186 – 196, 2012. URL: <http://www.sciencedirect.com/science/article/pii/S0167926011000770>, doi:10.1016/j.vlsi.2011.09.001.
- [32] Synopsys. *CCS Power Technical White Paper*, 3.0 edition, August 2006.
- [33] Cadence Design Systems, Inc. *Timing Library Format Reference*, version 4.3 edition, October 2000.
- [34] A. Poppe, Gy. Horváth, G. Nagy, M. Rencz, and V. Székely. Electro-thermal and logi-thermal simulators aimed at the temperature-aware design of complex integrated circuits. *24th IEEE SEMI-THERM Symposium*, 2008.
- [35] V. Székely, A. Poppe, M. Rencz, A. Csendes, and A. Páhi. Electro-thermal simulation: a realization by simultaneous iteration. *Analog Integrated Circuits and Signal Processing*, 28, 1997.
- [36] V. Székely, A. Páhi, A. Poppe, and M. Rencz. Electro-thermal simulation with the SISSI package. *Microelectronics Journal*, 21:21–31, 1999.
- [37] V. Székely, A. Poppe, A. Páhi, A. Csendes, G. Hajas, and M. Rencz. Electro-thermal and logi-thermal simulation of VLSI designs. *IEEE Trans. VLSI Syst.*, 5(3), sep 1997.

- [38] M. Rencz, V. Székely, A. Poppe, K. Torki, and B. Courtois. Electro-thermal simulation for the prediction of chip operation within the package. *19th IEEE SEMI-THERM Symposium*, 2003.
- [39] Stefan Wünsche, Christoph Clauß, Peter Schwarz, and Frank Winkler. Electro-thermal circuit simulation using simulator coupling. *IEEE Trans. VLSI Syst.*, 5(3), sep 1997.
- [40] The GDSII stream format, 15th August, 11:30 2012. URL: http://www.buchanan1.net/stream_description.shtml.
- [41] Cadence Design Systems. *LEF/DEF Language Reference*, product version 5.7 edition, November 2009.
- [42] SEMI p39-0308 - OASIS - open artwork system interchange standard, 15th August, 11:12 2012. URL: <http://ams.semi.org/ebusiness/standards/SEMISTandardDetail.aspx?ProductID=1948&DownloadID=1108>.
- [43] What is openaccess?, 15th August, 11:27 2012. URL: <http://www.si2.org/?page=69>.
- [44] The intel ivy bridge (core i7 3770k) review, 23th April 2012. URL: <http://www.anandtech.com/show/5771/the-intel-ivy-bridge-core-i7-3770k-review/3>.
- [45] Synopsys. *Liberty User Guides and Reference Manual Suite*, 2011.09 edition, September 2011.
- [46] IEEE standard for standard delay format (SDF) for the electronic design process. *IEEE Std 1497-2001*, 2001. doi:10.1109/IEEESTD.2001.93359.
- [47] B. Rebaud, M. Belleville, E. Beigné, C. Bernard, M. Robert, P. Maurine, and N. Azemard. Timing slack monitoring under process and environmental variations: Application to a DSP performance optimization. *Microelectronics Journal*, 42(5):718 – 732, 2011. URL: <http://www.sciencedirect.com/science/article/pii/S0026269211000292>, doi:10.1016/j.mejo.2011.02.005.
- [48] Sheng Lin, Yong-Bin Kim, and Fabrizio Lombardi. Design and analysis of a 32nm PVT tolerant CMOS SRAM cell for low leakage and high stability. *Integration, the VLSI Journal*, 43(2):176 – 187, 2010. URL: <http://www.sciencedirect.com/science/article/pii/S0167926010000040>, doi:10.1016/j.vlsi.2010.01.003.
- [49] Jinhui Wang, Na Gong, Ligang Hou, Xiaohong Peng, Ramalingam Sridhar, and Wuchen Wu. Leakage current, active power, and delay analysis of dynamic dual vt cmos circuits under P–V–T fluctuations. *Microelectronics Reliability*, 51(9–11):1498 – 1502, 2011. Proceedings of the 22th European Symposium on the RELIABILITY OF ELECTRON DEVICES, FAILURE PHYSICS AND ANALYSIS. URL: <http://www.sciencedirect.com/science/article/pii/S0026271411002010>, doi:10.1016/j.microrel.2011.06.011.
- [50] A.T. Winther, Wei Liu, A. Nannarelli, and S. Vrudhula. Temperature dependent wire delay estimation in floorplanning. In *NORCHIP, 2011*, pages 1 –4, nov. 2011. doi:10.1109/NORCHIP.2011.6126741.

- [51] A. Golda and A. Kos. Temperature influence on power consumption and time delay. In *Digital System Design, 2003. Proceedings. Euromicro Symposium on*, pages 378–382, sept. 2003. doi:10.1109/DSD.2003.1231970.
- [52] A. Sassone, A. Calimera, A. Macii, E. Macii, M. Poncino, R. Goldman, V. Melikyan, E. Babayan, and S. Rinaudo. Investigating the effects of inverted temperature dependence (itd) on clock distribution networks. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2012*, pages 165–166, 2012. doi:10.1109/DATE.2012.6176453.
- [53] D. Wolpert and P. Ampadu. A sensor to detect normal or reverse temperature dependence in nanoscale CMOS circuits. In *Defect and Fault Tolerance in VLSI Systems, 2009. DFT '09. 24th IEEE International Symposium on*, pages 193–201, oct. 2009. doi:10.1109/DFT.2009.47.
- [54] R. Kumar and V. Kursun. Reversed temperature-dependent propagation delay characteristics in nanometer CMOS circuits. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 53(10):1078–1082, oct. 2006. doi:10.1109/TCSII.2006.882218.
- [55] Ali Dasdan and Ivan Hom. Handling inverted temperature dependence in static timing analysis. *ACM Trans. Des. Autom. Electron. Syst.*, 11(2):306–324, April 2006. URL: <http://doi.acm.org/10.1145/1142155.1142158>, doi:10.1145/1142155.1142158.
- [56] Sean H. Wu, Alexander Tetelbaum, and Li-C. Wang. How does inverse temperature dependence affect timing sign-off. In Amara Amara, Thomas Ea, and Marc Belleville, editors, *Emerging Technologies and Circuits*, volume 2021 of *Lecture Notes in Electrical Engineering*, pages 179–189. Springer Netherlands, 2010. URL: http://dx.doi.org/10.1007/978-90-481-9379-0_13.
- [57] Lef/def exchange format ver5.8 and above, 21st February, 11:28 2013. URL: http://www.si2.org/openeda.si2.org/project/showfiles.php?group_id=84#p146.
- [58] Gnuplot, 25th February, 13:43 2013. URL: <http://www.gnuplot.info/>.
- [59] L. Pohl and E. Kollar. Extension of the SUNRED algorithm for electrothermal simulation and its application in failure analysis of large area (organic) semiconductor devices. In *Thermal Investigations of ICs and Systems (THERMINIC), 2011 17th International Workshop on*, pages 1–6, sept. 2011.
- [60] L. Pohl, E. Kollar, and A. Poppe. Nonlinear electro-thermal OLED model in SUNRED field simulator. In *Thermal Investigations of ICs and Systems (THERMINIC), 2010 16th International Workshop on*, pages 1–5, oct. 2010.
- [61] L. Pohl. Multithreading and Strassen’s algorithms in SUNRED field solver. In *Thermal Investigation of ICs and Systems, 2008. THERMINIC 2008. 14th International Workshop on*, pages 137–141, sept. 2008. doi:10.1109/THERMINIC.2008.4669895.